

# Package: CIMEHR (via r-universe)

June 9, 2026

**Title** Gaussian Clinically Informative Visiting and Observation Processes in Electronic Health Record (EHR) Data

**Version** 0.1.0

**Description** Fits semiparametric joint models for longitudinal electronic health record (EHR) data that addresses two-stage hierarchical missingness mechanism. The first stage is the visiting process, and the second stage is the observation process. The core CIMEHR method (Clinical Informative Missingness for Electronic Health Records) uses a three-stage procedure: partial likelihood with log-normal frailty for visit intensity, probit regression with shared latent factor-linked random effects for observation, and weighted least squares with risk-set centering for the outcome. These three stages are connected through a shared latent factor that induces dependence across all three processes. A data simulator and implementations of common benchmark methods (linear mixed models, multiple imputation, and others) are included for comparative studies. Detailed methods are described in Yang, Shi, and Mukherjee (2026) <[doi:10.48550/arXiv.2602.15374](https://doi.org/10.48550/arXiv.2602.15374)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Suggests** tibble, dplyr, tidyr, knitr, rmarkdown

**Imports** MASS, Rcpp, nleqslv, pbivnorm, numDeriv, stats, utils, data.table, mice, nlme, slim

**LinkingTo** Rcpp

**URL** <https://github.com/ysph-dsde/CIMEHR>

**BugReports** <https://github.com/ysph-dsde/CIMEHR/issues>

**VignetteBuilder** knitr

**Depends** R (>= 3.5)

**LazyData** true  
**Config/pak/sysreqs** cmake make libicu-dev libx11-dev zlib1g-dev  
**Repository** https://ysph-dsde.r-universe.dev  
**Date/Publication** 2026-06-02 14:58:53 UTC  
**RemoteUrl** https://github.com/ysph-dsde/cimehr  
**RemoteRef** HEAD  
**RemoteSha** 62c978710930a7248d2849aed64dbc15a27cba3b

## Contents

available_comparison_methods . . . . .	3
bootstrap . . . . .	4
CIMEHR . . . . .	7
CIMEHR_timevarying_integral . . . . .	10
CIMEHR_timevarying_ou . . . . .	14
coef.CIMEHR . . . . .	18
confint.CIMEHR . . . . .	19
EHRJoint . . . . .	20
extract_coefficient . . . . .	23
Inverse_intensity_rate_ratio_balancing . . . . .	24
Inverse_intensity_rate_ratio_weighting . . . . .	27
Joint_modeling_visiting_and_longitudinal_Liang . . . . .	29
Linear_increment_IP . . . . .	32
Linear_mixed_model . . . . .	35
method_comparisons . . . . .	37
Multiple_imputation_IP . . . . .	39
Pairwise_likelihood . . . . .	42
print.CIMEHR . . . . .	45
print.cimehr_bootstrap . . . . .	46
print.CIMEHR_timevarying_integral . . . . .	47
print.CIMEHR_timevarying_ou . . . . .	48
print.EHRJoint . . . . .	49
print.Inverse_intensity_rate_ratio_balancing . . . . .	50
print.Inverse_intensity_rate_ratio_weighting . . . . .	51
print.Joint_modeling_visiting_and_longitudinal_Liang . . . . .	52
print.Linear_increment_IP_result . . . . .	53
print.Linear_mixed_model_result . . . . .	54
print.Multiple_imputation_IP_result . . . . .	55
print.Pairwise_likelihood . . . . .	56
print.Summary_stat_result . . . . .	57
sim_data_gen . . . . .	57
sim_ehr_data . . . . .	61
summary.CIMEHR . . . . .	62
summary.cimehr_bootstrap . . . . .	63
summary.CIMEHR_timevarying_integral . . . . .	64
summary.CIMEHR_timevarying_ou . . . . .	65

summary.EHRJoint . . . . .	66
summary.Inverse_intensity_rate_ratio_balancing . . . . .	67
summary.Inverse_intensity_rate_ratio_weighting . . . . .	68
summary.Joint_modeling_visiting_and_longitudinal_Liang . . . . .	69
summary.Linear_increment_IP_result . . . . .	70
summary.Linear_mixed_model_result . . . . .	71
summary.Multiple_imputation_IP_result . . . . .	72
summary.Pairwise_likelihood . . . . .	73
summary.Summary_stat_result . . . . .	74
summary_observation . . . . .	75
summary_outcome . . . . .	75
Summary_stat . . . . .	76
summary_visiting . . . . .	77
vcov.CIMEHR . . . . .	78

**Index** **80**

available\_comparison\_methods  
*Available Methods for method\_comparisons()*

**Description**

Available Methods for method\_comparisons()

**Usage**

available\_comparison\_methods()

**Value**

Character vector of method names that can be fitted by method\_comparisons().

**Examples**

available\_comparison\_methods()

bootstrap

*Bootstrap Confidence Intervals for Selected Model Fits***Description**

Computes bootstrap standard errors and percentile confidence intervals by resampling subjects with replacement and re-fitting the original model. Suitable for methods that do not provide closed-form standard errors ([CIMEHR\\_timevarying\\_integral](#), [Inverse\\_intensity\\_rate\\_ratio\\_weighting](#), and [Inverse\\_intensity\\_rate\\_ratio\\_balancing](#)), and as a robustness check or supplement to the analytic standard errors returned by [CIMEHR](#), [CIMEHR\\_timevarying\\_ou](#), [EHRJoint](#), and [Joint\\_modeling\\_visiting\\_and](#)

**Usage**

```
bootstrap(
  fit,
  data,
  id = "id",
  B = 1000,
  level = 0.95,
  covars = list(),
  which_estimate = c("bal", "mle"),
  seed = NULL,
  verbose = FALSE
)
```

**Arguments**

<code>fit</code>	A fitted model object from one of the selected method functions. Supported classes: "CIMEHR", "CIMEHR_timevarying_integral", "CIMEHR_timevarying_ou", "EHRJoint", "Joint_modeling_visiting_and_longitudinal_Liang", "Inverse_intensity_rate_ratio_weighting", "Inverse_intensity_rate_ratio_balancing".
<code>data</code>	The data.frame used to fit <code>fit</code> , in long (panel) format with one row per subject-visit.
<code>id</code>	Character string giving the name of the subject identifier column. Default is "id".
<code>B</code>	Integer; number of bootstrap replicates. Default is 1000.
<code>level</code>	Confidence level for percentile intervals. Default is 0.95.
<code>covars</code>	Named list of model-specific arguments forwarded to the re-fitting function on each replicate. Defaults to an empty list, in which case the bootstrap re-uses the covariate arguments stored on <code>fit\$call</code> from the original fit. Any entries supplied here override the corresponding entries from the original call. Names must match the argument names of the underlying fitting function exactly; unrecognised names trigger a warning.

<code>which_estimate</code>	For "Inverse_intensity_rate_ratio_balancing" fits only: which estimator to display in print and summary. One of "bal" (default, balancing weights) or "mle" (maximum likelihood estimation (MLE) weights). Both estimators are always stored in <code>boot_matrix</code> with column prefixes "s3.bal." and "s3.mle.".
<code>seed</code>	Integer; random seed for reproducibility. Default is NULL (no seed set).
<code>verbose</code>	Logical; if TRUE, a progress message is printed every 100 replicates. Default is FALSE.

## Details

For two-stage methods ([Linear\\_increment\\_IP](#), [Multiple\\_imputation\\_IP](#)), the bootstrap must cover the full two-stage procedure. We recommend users implement a manual subject-level bootstrap loop.

Subjects are resampled with replacement. For each replicate, all rows belonging to a sampled subject are included; subjects drawn multiple times receive distinct integer identifiers (IDs) to avoid duplicate identifiers in the resampled dataset.

Replicates that throw an error or a convergence-related warning are silently discarded. A warning is issued if more than 10 percent of replicates fail.

## Value

An object of class "cimehr\_bootstrap", a named list with:

<code>estimates</code>	Named numeric vector of all-stage point estimates from the original fit, with the same "s1." / "s2." / "s3." prefixes as <code>boot_matrix</code> .
<code>boot_matrix</code>	Numeric matrix of dimension <code>B_success</code> by <code>p</code> ; each row is the full parameter vector from one bootstrap replicate, covering <i>all</i> estimated stages. Columns are prefixed by stage: "s1." (visiting process), "s2." (observation process, where applicable), "s3." (longitudinal outcome). For "Inverse_intensity_rate_ratio_balancing", outcome columns are prefixed "s3.bal." and "s3.mle.".
<code>se</code>	Named numeric vector of bootstrap standard errors for the Stage-3 outcome coefficients of the displayed estimator (prefixes stripped). Standard errors for all stages are recoverable from <code>boot_matrix</code> directly.
<code>ci</code>	Numeric matrix with columns <code>CI.low</code> and <code>CI.high</code> ; percentile intervals for the Stage-3 outcome coefficients of the displayed estimator (prefixes stripped).
<code>level</code>	The confidence level used.
<code>B</code>	Number of replicates requested.
<code>B_success</code>	Number of replicates that converged.
<code>method</code>	Character string; class of the original fit.
<code>which_estimate</code>	For "Inverse_intensity_rate_ratio_balancing": which estimator is shown in print/summary ("bal" or "mle"). NULL for all other methods.
<code>call</code>	The matched call.

### Covariate arguments by method

By default, `bootstrap()` re-uses the covariate arguments that were passed to the original fitting call (`fit$call`). Supply entries in `covars` only when you want to override one of those arguments for the bootstrap re-fits. Names must match the argument names of the underlying fitting function exactly. Recognised entries:

**CIMEHR, CIMEHR\_timevarying\_integral, CIMEHR\_timevarying\_ou:** `covars_visit_XV`, `covars_outcome_fixed_XY`, `covars_outcome_random_link_ZY`, `covars_obs_fixed_X0`, `covars_obs_random_link_Z0`.

**EHRJoint:** `visit_covs`, `long_covs`, `random_covs`, `obs_link`, `sigma_sq_floor`.

**Joint\_modeling\_visiting\_and\_longitudinal\_Liang:** `visit_covs`, `long_covs`, `random_covs`.

**Inverse\_intensity\_rate\_ratio\_weighting:** `visit_covs`, `formula`.

**Inverse\_intensity\_rate\_ratio\_balancing:** `visit_covs`, `balance_covs`, `formula`, `phi`.

### Plug-in vs. bootstrap

`CIMEHR` and `CIMEHR_timevarying_ou` already compute sandwich (plug-in) standard errors from per-subject score contributions evaluated at the fitted parameters — no re-fitting is needed. `EHRJoint` and `Joint_modeling_visiting_and_longitudinal_Liang` return model-based standard errors for the visiting (and observation, for `EHRJoint`) processes and a clustered sandwich estimator for the outcome stage; these are conditional on the Stage 1 nuisance estimates and do not propagate their estimation uncertainty. Bootstrap is provided for all four of these methods as an optional check or to propagate Stage 1 uncertainty. For `CIMEHR_timevarying_integral`, `Inverse_intensity_rate_ratio_weighting`, and `Inverse_intensity_rate_ratio_balancing`, per-subject score contributions are not exposed by the fitting routines, so bootstrap is the recommended approach for uncertainty quantification.

### Accessing standard errors for all stages

`print` and `summary` display bootstrap standard errors (SEs) and confidence intervals for the Stage-3 outcome coefficients only ("`s3.*`" columns of `boot_matrix`). Bootstrap SEs for all other stages are stored in `boot_matrix` and can be extracted manually, for example:

```
bs <- bootstrap(fit, data = dat, B = 500)

# Stage-1 SE (visiting process)
apply(bs$boot_matrix[, grepl("^s1\\.", colnames(bs$boot_matrix))], drop = FALSE],
      2, sd)

# Stage-2 SE (observation process, CIMEHR family)
apply(bs$boot_matrix[, grepl("^s2\\.", colnames(bs$boot_matrix))], drop = FALSE],
      2, sd)
```

### References

Bradley Efron and Robert J. Tibshirani (1994). *An Introduction to the Bootstrap*. Chapman & Hall/CRC.

### See Also

[print.cimehr\\_bootstrap](#), [summary.cimehr\\_bootstrap](#)

**Examples**

```

dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_weighting(dat,
                                             outcome = "log_HbA1c",
                                             visit_covs = "Age")
bs <- bootstrap(fit, data = dat, B = 3, seed = 1)
print(bs)

```

CIMEHR

*Clinical Informative Missingness for Electronic Health Records  
(CIMEHR) – Joint Three-Stage Model*

**Description**

Fits a three-stage semiparametric joint model for longitudinal Electronic Health Record (EHR) data that simultaneously accounts for an informative visiting process and an informative observation process. Stage 1 estimates the visiting-process intensity via partial likelihood with a log-normal frailty. Stage 2 models the observation indicator with a probit regression that may include latent-factor-linked random effects. Stage 3 estimates the longitudinal outcome coefficients by Weighted Least Squares (WLS) with risk-set centering. Sandwich variance estimators are provided for all three stages.

**Usage**

```

CIMEHR(
  data,
  id_col = "id",
  time_col = "time",
  y_col = "Y",
  r_col = "R",
  censor_col = "C",
  covars_visit_XV = NULL,
  covars_outcome_fixed_XY = NULL,
  covars_outcome_random_link_ZY = NULL,
  covars_obs_fixed_X0 = NULL,
  covars_obs_random_link_Z0 = NULL,
  time_precision = 0.01
)

```

**Arguments**

**data** A data.frame in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, censoring time, and all specified covariates. Rows with NA in the time column are treated as unvisited time points.

<code>id_col</code>	Character string giving the name of the subject identifier column in data. Default is "id".
<code>time_col</code>	Character string giving the name of the visit-time column in data. Default is "time".
<code>y_col</code>	Character string giving the name of the outcome column in data. Default is "Y".
<code>r_col</code>	Character string giving the name of the binary observation indicator column (1 = outcome observed at visit, 0 = outcome missing at visit). Default is "R".
<code>censor_col</code>	Character string giving the name of the column containing each subject's censoring (end-of-follow-up) time. One value per subject is expected; the value on the first row for each subject is used. Default is "C".
<code>covars_visit_XV</code>	Character vector of covariate column names for the visiting-process intensity model. These enter the exponent of the multiplicative intensity. Default is NULL (no covariates; the visiting model becomes a pure baseline-hazard plus frailty fit).
<code>covars_outcome_fixed_XY</code>	Character vector of covariate column names for the fixed-effect part of the longitudinal outcome model. No intercept is added automatically. At least one covariate is required. Default is NULL.
<code>covars_outcome_random_link_ZY</code>	Character vector of covariate column names that interact with the conditional expectation of the latent factor $U_i$ in the outcome model. An intercept column is prepended automatically. Including <code>time_col</code> in this vector adds a time-varying latent link. Set to NULL for an intercept-only latent link. Default is NULL.
<code>covars_obs_fixed_XO</code>	Character vector of covariate column names for the fixed-effect part of the observation-process probit model. An intercept is prepended automatically. Set to NULL for an intercept-only probit. Default is NULL.
<code>covars_obs_random_link_ZO</code>	Character vector of covariate column names that receive independent random effects in the observation process. An intercept is prepended automatically. Set to NULL or an empty vector to fit a fixed-effects-only probit. Default is NULL.
<code>time_precision</code>	Positive numeric scalar for time discretisation. Visit times are rounded to the nearest multiple of <code>time_precision</code> before constructing risk sets. Set to NULL or 0 to skip rounding. Default is 0.01.

## Details

The CIMEHR estimator proceeds in three stages:

**Stage 1 - Visiting Process.** The recurrent-event intensity for subject  $i$  is modelled as

$$\lambda_i(t) = \eta_i \exp(\gamma^\top X_i^V) \lambda_0(t),$$

where  $\eta_i = \exp(\mu_0 + \sigma_\zeta U_i)$  is a log-normal frailty with  $U_i \sim N(0, 1)$  and  $\mu_0 = -\sigma_\zeta^2/2$  ensures  $E[\eta_i] = 1$ . The coefficient vector  $\gamma$  is estimated by Newton-Raphson on the partial likelihood

score, the baseline hazard  $\lambda_0$  by the Breslow estimator, and the frailty variance  $\sigma_\zeta^2$  by a method-of-moments estimator. Subject-level posterior summaries ( $\hat{\mu}_{U_i}, \hat{\sigma}_{U_i}^2$ ) are obtained via Laplace approximation. A sandwich estimator provides inference for  $\gamma$ .

**Stage 2 - Observation Process.** At each visit the probability that the outcome is recorded is modelled as

$$P(R_{ij} = 1 \mid U_i, q_i) = \Phi(\alpha^\top X_{ij}^O + \delta^\top Z_{ij}^O \cdot U_i + q_i^\top Z_{ij}^O),$$

where  $q_i \sim N(0, \Sigma_q)$  are independent random effects and  $\Phi$  is the standard-normal cumulative distribution function (CDF). The marginal log-likelihood (integrated over  $q_i$  using the empirical Bayes posterior of  $U_i$ ) is maximised by `nlminb` with analytic gradients. When the estimated frailty variance is negligible or no random covariates are specified, a fixed-effects probit is fitted by iteratively reweighted least squares. A clustered sandwich estimator provides standard errors.

**Stage 3 - Longitudinal Outcome.** The outcome model is

$$Y_i(t) = \beta^\top X_i^Y(t) + \theta^\top Z_i^Y(t) \kappa_i(t) + \varepsilon_i(t),$$

where  $\kappa_i(t) = E[U_i \mid \text{observation history up to } t]$  is the conditional expectation of the latent factor given the observation history. Estimation uses weighted least squares with time-varying risk-set centering, where the weights combine the marginal observation probability  $\varpi_i(t)$  and the visit intensity. A sandwich estimator provides standard errors for  $\beta$ .

## Value

A named list with three components, each itself a named list:

`visiting_process` A list with elements:

- `gamma_V` Named numeric vector of visiting-process regression coefficients from partial likelihood.
- `se_gamma` Named numeric vector of sandwich standard errors for `gamma_V`.
- `vcov_gamma` Numeric matrix; the sandwich variance-covariance matrix for `gamma_V`.
- `Lambda0_hat` Named numeric vector of Breslow baseline hazard increments  $d\hat{\Lambda}_0(t_k)$  at each unique discretised event time.
- `sigma2_zeta` Numeric scalar; estimated log-normal frailty variance on the log scale.
- `mu_Ui` Named numeric vector of empirical Bayes posterior means for the standardised latent factor  $U_i$ .
- `sigma2_Ui` Named numeric vector of empirical Bayes posterior variances for  $U_i$ .
- `Ci` Named numeric vector of per-subject censoring times.
- `mi` Named integer vector of per-subject visit counts.

`observation_process` A list with elements:

- `alpha_fixed` Named numeric vector of fixed-effect probit coefficients (including intercept).
- `se_alpha` Named numeric vector of sandwich standard errors for `alpha_fixed`.
- `vcov_obs` Numeric matrix; sandwich variance-covariance matrix for all observation-process parameters.
- `delta_latent_random_link` (Present only when random effects are fitted.) Named numeric vector of latent-link coefficients  $\delta$ .
- `se_delta` (Present only when random effects are fitted.) Named numeric vector of sandwich standard errors for `delta_latent_random_link`.

Sigma\_q (Present only when random effects are fitted.) Named numeric vector of estimated random-effect standard deviations.

longitudinal\_outcome A list with elements:

beta\_Y Named numeric vector of outcome fixed-effect coefficients.

theta\_Y Named numeric vector of outcome latent-link coefficients.

se\_beta\_naive Named numeric vector of sandwich standard errors for beta\_Y.

vcov\_joint Numeric matrix; sandwich variance-covariance matrix for the full coefficient vector  $c(\text{beta}_Y, \text{theta}_Y)$ .

## References

Yang, C.-H., Shi, X., and Mukherjee, B. (2026). "Joint Modeling of Longitudinal EHR Data with Shared Random Effects for Informative Visiting and Observation Processes." arXiv:2602.15374. doi:10.48550/arXiv.2602.15374.

## See Also

[CIMEHR\\_timevarying\\_integral](#) for a Gauss-Hermite quadrature variant with online filtering; [CIMEHR\\_timevarying\\_ou](#) for the Ornstein-Uhlenbeck pairwise composite likelihood extension; [sim\\_data\\_gen](#) for simulating data compatible with this model.

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_XO = "Age"
)
```

---

CIMEHR\_timevarying\_integral

*Clinical Informative Missingness for Electronic Health Records (CIMEHR) with Gauss-Hermite (GH) Quadrature – Time-Varying Variant*

---

## Description

Fits a three-stage semiparametric joint model for longitudinal Electronic Health Record (EHR) data, extending [CIMEHR](#) by replacing the closed-form marginal observation-probability approximation with Gauss-Hermite (GH) quadrature and an online Laplace filter. This enables exact integration over both the shared latent factor  $U_i$  and the observation-process random effects  $q_i$ , and supports a time-varying observation process through a lagged observation indicator.

Stage 1 estimates the visiting-process intensity via Andersen–Gill partial likelihood with a log-normal frailty. Stage 2 maximizes a Gauss-Hermite-integrated marginal log-likelihood for the probit observation model. Stage 3 estimates the longitudinal outcome coefficients by weighted least squares with time-varying risk-set centering and sequentially updated  $\omega_i(t)$  and  $\kappa_i(t)$ .

### Usage

```
CIMEHR_timevarying_integral(
  data,
  id_col = "id",
  time_col = "time",
  y_col = "Y",
  r_col = "R",
  r_lag_col = NULL,
  covars_visit_XV = NULL,
  covars_outcome_fixed_XY = NULL,
  covars_outcome_random_link_ZY = NULL,
  covars_obs_fixed_X0 = NULL,
  covars_obs_random_link_Z0 = NULL,
  gh_points_U = 7,
  gh_points_q = 5,
  gh_clip_eta = 12,
  clip_k = 12,
  filter_max_iter = 6,
  verbose = FALSE
)
```

### Arguments

<code>data</code>	A <code>data.frame</code> in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates. Rows with NA in the time column are treated as unvisited time points.
<code>id_col</code>	Character string giving the name of the subject identifier column. Default is "id".
<code>time_col</code>	Character string giving the name of the visit-time column. Default is "time".
<code>y_col</code>	Character string giving the name of the outcome column. Default is "Y".
<code>r_col</code>	Character string giving the name of the binary observation indicator column (1 = outcome observed, 0 = outcome missing at visit). Default is "R".
<code>r_lag_col</code>	Character string giving the name of a lagged observation indicator column. When non-NULL, a time-varying coefficient $\lambda$ on $R_{\{i, j-1\}}$ is added to the observation-process linear predictor, allowing the probability of observation to depend on whether the outcome was observed at the previous visit. When NULL (the default), the lagged indicator is constructed internally with first-visit values set to zero, but the $\lambda$ coefficient is omitted from the model.
<code>covars_visit_XV</code>	Character vector of covariate column names for the visiting-process intensity model. Default is NULL.

covars_outcome_fixed_XY	Character vector of covariate column names for the fixed-effect part of the longitudinal outcome model. No intercept is added automatically. At least one covariate is required. Default is NULL.
covars_outcome_random_link_ZY	Character vector of covariate column names that interact with the conditional expectation of the latent factor $U_i$ in the outcome model. An intercept is prepended automatically. Including <code>time_col</code> adds a time-varying link. Default is NULL.
covars_obs_fixed_X0	Character vector of covariate column names for the fixed-effect part of the observation-process probit model. An intercept is prepended automatically. Default is NULL.
covars_obs_random_link_Z0	Character vector of covariate column names receiving independent random effects in the observation process. An intercept is prepended automatically. Set to NULL or an empty vector for a fixed-effects-only probit. Default is NULL.
gh_points_U	Integer giving the number of Gauss–Hermite quadrature nodes for integrating over the standardised latent factor $U_i$ . More nodes improve accuracy at the cost of speed. Default is 7.
gh_points_q	Integer giving the number of Gauss–Hermite quadrature nodes for each dimension of the observation-process random effects $q_i$ . A tensor-product grid is formed when $q_i$ is multi-dimensional. Default is 5.
gh_clip_eta	Numeric scalar; linear predictor values in the probit model are clipped to $[-gh\_clip\_eta, gh\_clip\_eta]$ for numerical stability during Stage 2 optimisation. Default is 12.
clip_k	Numeric scalar; the standardised argument to $\Phi$ is clipped to $[-clip\_k, clip\_k]$ when computing marginal observation probabilities $\omega_i(t)$ and conditional expectations $\kappa_i(t)$ . Default is 12.
filter_max_iter	Integer giving the maximum number of Newton iterations for the Laplace approximation used in the online filter that updates the joint posterior of $(U_i, q_i)$ after each observation event. Default is 6.
verbose	Logical; if TRUE, progress messages are printed during estimation. Default is FALSE.

## Details

The model structure is the same as [CIMEHR](#) but the computational strategy differs in two key respects:

**Stage 2 – Gauss–Hermite Quadrature.** Instead of the Laplace-style marginal approximation used in [CIMEHR](#), the marginal observation-process log-likelihood is evaluated by Gauss–Hermite quadrature over  $U_i$  (centred on the empirical Bayes posterior from Stage 1) and over  $q_i$ . The quadrature grid has `gh_points_U` nodes for  $U_i$  and a tensor-product grid of `gh_points_q` nodes per dimension for  $q_i$ . The marginal negative log-likelihood is minimised with `nlm`. When `r_lag_col` is specified, an additional parameter  $\lambda$  captures dependence of observation on the previous observation status.

**Stage 3 – Online Filtering.** The marginal observation probability  $\omega_i(t)$  and conditional expectation  $\kappa_i(t)$  are computed via a sequential (online) Laplace filter that updates the joint posterior of  $(U_i, q_i)$

after each observation event. This produces time-varying versions of  $\omega$  and  $\kappa$  that condition on the full observation history up to each time point, rather than treating them as static subject-level quantities.

The visiting-process estimation (Stage 1) is identical to CIMEHR.

## Value

A named list with three components:

`visiting_process` A list with elements:

`gamma_V` Named numeric vector of visiting-process regression coefficients (Andersen–Gill partial likelihood).

`Lambda0_C` Numeric scalar; Breslow estimate of the cumulative baseline hazard over the full follow-up window.

`sigma2_zeta` Numeric scalar; estimated log-normal frailty variance on the log scale.

`mu_Ui` Named numeric vector of empirical Bayes posterior means for the standardised latent factor  $U_i$ .

`sigma2_Ui` Named numeric vector of empirical Bayes posterior variances for  $U_i$ .

`observation_process` A list with elements:

`alpha_fixed` Named numeric vector of fixed-effect probit coefficients (including intercept).

`delta_latent_random_link` (Present only when random effects are fitted.) Named numeric vector of latent-link coefficients  $\delta$ .

`Sigma_q` (Present only when random effects are fitted.) Numeric matrix; estimated random-effect covariance matrix (diagonal).

`lambda` (Present only when `r_lag_col` is non-NULL.) Numeric scalar; estimated coefficient for the lagged observation indicator.

`longitudinal_outcome` A list with elements:

`beta_Y` Named numeric vector of outcome fixed-effect coefficients.

`theta_Y` Named numeric vector of outcome latent-link coefficients.

## Note

Computation time increases with `gh_points_U`, `gh_points_q`, and the number of subjects. For very large datasets, consider using CIMEHR (which avoids quadrature) as a faster alternative.

## References

Yang, C.-H., Shi, X., and Mukherjee, B. (2026). "Joint Modeling of Longitudinal EHR Data with Shared Random Effects for Informative Visiting and Observation Processes." arXiv:2602.15374. doi:10.48550/arXiv.2602.15374.

## See Also

CIMEHR for the base version without quadrature; CIMEHR\_timevarying\_ou for the Ornstein-Uhlenbeck pairwise composite likelihood extension; sim\_data\_gen for simulating compatible data.

**Examples**

```

dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR_timevarying_integral(
  data           = dat,
  y_col         = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_XO = "Age",
  gh_points_U    = 2,
  gh_points_q    = 2
)
print(fit)

```

---

CIMEHR\_timevarying\_ou *Clinical Informative Missingness for Electronic Health Records (CIMEHR) with Ornstein–Uhlenbeck (OU) Pairwise Composite Likelihood (PCL) Extension – Time-Varying Variant*

---

**Description**

Fits a three-stage semiparametric joint model for longitudinal Electronic Health Record (EHR) data, extending [CIMEHR](#) by modelling temporal serial correlation in the observation process with a stationary Ornstein–Uhlenbeck (OU) process and estimating the observation-process parameters via Pairwise Composite Likelihood (PCL).

Stage 1 estimates the visiting-process intensity via Andersen–Gill partial likelihood with a log-normal frailty (identical to [CIMEHR](#)). Stage 2 maximises a pairwise composite log-likelihood over adjacent (or all) within-subject visit pairs, integrating over the bivariate probit structure induced by the OU process. An analytic gradient is provided for efficiency. Stage 3 estimates the longitudinal outcome coefficients by weighted least squares with risk-set centering and a robust sandwich variance estimator.

**Usage**

```

CIMEHR_timevarying_ou(
  data,
  id_col = "id",
  time_col = "time",
  y_col = "Y",
  r_col = "R",
  covars_visit_XV = NULL,
  covars_outcome_fixed_XY = NULL,
  covars_outcome_random_link_ZY = NULL,
  covars_obs_fixed_XO = NULL,
  covars_obs_random_link_ZO = NULL,
  pair_type = c("adjacent", "all"),
  phi_init = 0.1,

```

```

    phi_bounds = c(1e-04, 100),
    optim_method = "L-BFGS-B",
    verbose = FALSE
)

```

## Arguments

data	A data.frame in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates. Rows with NA in the time column are treated as unvisited time points.
id_col	Character string giving the name of the subject identifier column. Default is "id".
time_col	Character string giving the name of the visit-time column. Default is "time".
y_col	Character string giving the name of the outcome column. Default is "Y".
r_col	Character string giving the name of the binary observation indicator column (1 = outcome observed, 0 = outcome missing at visit). Default is "R".
covars_visit_XV	Character vector of covariate column names for the visiting-process intensity model. Default is NULL.
covars_outcome_fixed_XY	Character vector of covariate column names for the fixed-effect part of the longitudinal outcome model. No intercept is added automatically. At least one covariate is required. Default is NULL.
covars_outcome_random_link_ZY	Character vector of covariate column names that interact with the conditional expectation of the latent factor $U_i$ in the outcome model. An intercept is prepended automatically. Including time_col adds a time-varying link. Default is NULL.
covars_obs_fixed_XO	Character vector of covariate column names for the fixed-effect part of the observation-process probit model. An intercept is prepended automatically. Default is NULL.
covars_obs_random_link_ZO	Character vector of covariate column names receiving independent random effects in the observation process. An intercept is prepended automatically. Set to NULL for a fixed-effects-only probit (in which case PCL is skipped and a standard probit generalized linear model (GLM) is fitted). Default is NULL.
pair_type	Character string specifying which within-subject visit pairs to use for the pairwise composite likelihood. One of: "adjacent" (Default.) Use only temporally adjacent pairs $(j, j + 1)$ . This is computationally cheaper and typically sufficient. "all" Use all $\binom{m_i}{2}$ pairs per subject. More informative but $O(m_i^2)$ per subject.
phi_init	Numeric scalar giving the starting value for the OU decay parameter $\phi$ in the optimisation. Default is 0.1.
phi_bounds	Numeric vector of length 2 giving the lower and upper bounds for $\phi$ (on the original scale; internally log-transformed). Default is c(1e-4, 100).

optim_method	Character string specifying the optimisation method passed to <code>optim</code> . Default is "L-BFGS-B".
verbose	Logical; if TRUE, progress messages are printed during estimation. Default is FALSE.

## Details

The model extends `CIMEHR` by introducing an Ornstein–Uhlenbeck (OU) process  $\xi_i(t)$  in the observation model:

$$\tilde{R}_{ij} = \alpha^\top X_{ij}^O + \delta^\top Z_{ij}^O \cdot U_i + q_i^\top Z_{ij}^O + \xi_i(t_{ij}),$$

where  $R_{ij} = I(\tilde{R}_{ij} > 0)$ ,  $q_i \sim N(0, \Sigma_q)$ , and  $\xi_i$  is a stationary OU process with unit marginal variance and autocorrelation  $\text{Cor}(\xi_i(s), \xi_i(t)) = \exp(-\phi|t - s|)$ . The OU component captures temporal serial correlation in the observation process that is not explained by covariates or the shared latent factor.

**Stage 2 – Pairwise Composite Likelihood.** Because the full likelihood involves high-dimensional integrals over  $\xi_i$ , estimation proceeds via pairwise composite likelihood. For each within-subject visit pair  $(j, k)$  the bivariate probit probability  $P(R_{ij} = r_j, R_{ik} = r_k)$  is computed using `pbivnorm`, and the composite log-likelihood is

$$cl(\theta) = \sum_i \sum_{j < k} \log \Phi_2(d_j z_j, d_k z_k; d_j d_k \rho_{jk}),$$

where  $d_j = 2R_{ij} - 1$ ,  $z_j$  is the standardised marginal probit argument, and  $\rho_{jk}$  combines the latent factor, random-effect, and OU contributions to the pairwise correlation. An analytic gradient is provided and the composite log-likelihood is maximised with `optim` using the specified `optim_method`.

**Stages 1 and 3** are identical to `CIMEHR`, except that Stage 3 uses a static (non-filtered) version of  $\omega_i(t)$  and  $\kappa_i(t)$ .

S3 methods `print`, `summary`, `coef`, `vcov`, and `confint` are available for objects of class "CIMEHR\_timevarying\_ou".

## Value

An object of class "CIMEHR\_timevarying\_ou", which is a named list with the following components:

`visiting_process` A list with elements:

`gamma_v` Named numeric vector of visiting-process regression coefficients.

`Lambda0_hat` Named numeric vector of Breslow baseline hazard increments at each unique event time.

`sigma2_zeta` Numeric scalar; estimated log-normal frailty variance on the log scale.

`mu_Ui` Named numeric vector of empirical Bayes posterior means for  $U_i$ .

`sigma2_Ui` Named numeric vector of empirical Bayes posterior variances for  $U_i$ .

`observation_process` A list with elements:

`alpha_fixed` Named numeric vector of fixed-effect probit coefficients (including intercept).

`delta_latent_random_link` (Present only when random effects are fitted.) Named numeric vector of latent-link coefficients  $\delta$ .

Sigma\_q (Present only when random effects are fitted.) Named numeric vector of estimated random-effect variances  $\sigma_{q_i}^2$ .

phi\_OU (Present only when random effects are fitted.) Numeric scalar; estimated OU decay parameter  $\phi$ . Larger values indicate faster decay of serial correlation.

pcl\_value (Present only when random effects are fitted.) Numeric scalar; minimised negative pairwise composite log-likelihood.

convergence (Present only when random effects are fitted.) Integer convergence code from `optim`.

longitudinal\_outcome A list with elements:

beta\_Y Named numeric vector of outcome fixed-effect coefficients.

theta\_Y Named numeric vector of outcome latent-link coefficients.

se\_beta Named numeric vector of sandwich standard errors for beta\_Y.

vcov\_robust Numeric matrix; sandwich variance–covariance matrix for `c(beta_Y, theta_Y)`.

model\_info A list with elements:

n\_subjects Integer; number of unique subjects.

n\_visits Integer; total number of visits.

n\_observed Integer; total number of visits with  $R = 1$ .

pair\_type Character string echoing the `pair_type` argument.

median\_gap Numeric scalar; median time gap between consecutive visits within subjects.

call The matched call.

## References

Yang, C.-H., Shi, X., and Mukherjee, B. (2026). "Joint Modeling of Longitudinal EHR Data with Shared Random Effects for Informative Visiting and Observation Processes." arXiv:2602.15374. doi:10.48550/arXiv.2602.15374.

## See Also

[CIMEHR](#) for the base version without Ornstein–Uhlenbeck (OU) serial correlation; [CIMEHR\\_timevarying\\_integral](#) for the Gauss–Hermite quadrature variant; [sim\\_data\\_gen](#) for simulating data compatible with this model;

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR_timevarying_ou(
  data           = dat,
  y_col         = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age",
  pair_type     = "adjacent"
)
print(fit)
```

---

coef.CIMEHR	<i>Extract Coefficients from a Clinical Informative Missingness for Electronic Health Records (CIMEHR) Fit</i>
-------------	--

---

### Description

Extract Coefficients from a Clinical Informative Missingness for Electronic Health Records (CIMEHR) Fit

### Usage

```
## S3 method for class 'CIMEHR'
coef(object, type = c("outcome", "visiting", "observation"), ...)

## S3 method for class 'CIMEHR_timevarying_integral'
coef(object, type = c("outcome", "visiting", "observation"), ...)

## S3 method for class 'CIMEHR_timevarying_ou'
coef(object, type = c("outcome", "visiting", "observation"), ...)
```

### Arguments

object	A CIMEHR, CIMEHR_timevarying_integral, or CIMEHR_timevarying_ou object.
type	Which coefficients: "outcome" (default), "visiting", or "observation".
...	Ignored.

### Value

Named numeric vector.

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_XO = "Age"
)
coef(fit)
coef(fit, type = "visiting")
```

---

confint.CIMEHR      *Confidence Intervals for Clinical Informative Missingness for Electronic Health Records (CIMEHR) Outcome Coefficients*

---

## Description

Computes Wald confidence intervals using sandwich standard errors.

## Usage

```
## S3 method for class 'CIMEHR'  
confint(object, parm = NULL, level = 0.95, ...)  
  
## S3 method for class 'CIMEHR_timevarying_ou'  
confint(object, parm = NULL, level = 0.95, ...)
```

## Arguments

object	A CIMEHR or CIMEHR_timevarying_ou object.
parm	Character vector of parameter names, or NULL for all.
level	Confidence level. Default is 0.95.
...	Ignored.

## Value

Numeric matrix with lower and upper bounds, or NULL if standard errors are unavailable.

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]  
fit <- CIMEHR(  
  data           = dat,  
  y_col          = "log_HbA1c",  
  covars_visit_XV = "Age",  
  covars_outcome_fixed_XY = "Age",  
  covars_obs_fixed_XO = "Age"  
)  
confint(fit)  
confint(fit, parm = "Age", level = 0.90)
```

---

EHRJoint	<i>Joint Estimation for Electronic Health Record (EHR) Longitudinal Processes (EHRJoint)</i>
----------	--

---

## Description

Fits the three-stage joint estimator of Du, Shi, and Mukherjee (2025) for irregularly observed longitudinal Electronic Health Record (EHR) data with informative visiting and informative observation processes. Stage 1 estimates the visiting-process intensity by solving an Andersen-Gill partial-likelihood score equation. Stage 2 fits the observation indicator via a probit (default) or logistic Generalized Linear Model (GLM). Stage 3 estimates the longitudinal outcome coefficients from a Liang-type risk-set-centred estimating equation, augmented with a frailty-compensation covariate constructed from the Stage 1 estimates. Analytic standard errors are provided for all three stages: model-based for the visiting-process partial likelihood and for the observation-process GLM, and a clustered sandwich estimator for the outcome stage.

## Usage

```
EHRJoint(
  data,
  id = "id",
  time = "time",
  outcome = "Y",
  obs_indicator = "R",
  visit_covs = c("Z", "X"),
  long_covs = c("Z", "X"),
  random_covs = "Z",
  obs_link = c("probit", "logit"),
  sigma_sq_floor = 1e-06
)
```

## Arguments

<code>data</code>	A <code>data.frame</code> in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates. Rows with NA in the time column are treated as placeholder rows for zero-visit subjects.
<code>id</code>	Character string giving the name of the subject identifier column. Default is "id".
<code>time</code>	Character string giving the name of the visit-time column. Default is "time".
<code>outcome</code>	Character string giving the name of the outcome column. Default is "Y".
<code>obs_indicator</code>	Character string giving the name of the binary observation indicator column (1 = outcome observed, 0 = outcome missing at visit). Default is "R".
<code>visit_covs</code>	Character vector of covariate column names entering both the visiting-process intensity and the observation-process GLM. Default is <code>c("Z", "X")</code> .

long_covs	Character vector of covariate column names for the fixed-effect part of the longitudinal outcome model. Default is c("Z", "X").
random_covs	Character vector of covariate column names that interact with the latent factor in the outcome model (frailty compensation). An intercept column is prepended automatically. Default is "Z".
obs_link	Character string giving the link function for the observation-process GLM. One of "probit" (default, matching the common data-generating process) or "logit".
sigma_sq_floor	Positive numeric scalar; numerical threshold below which the estimated frailty variance is treated as zero. When the estimate falls below this floor, the frailty-compensation covariates are dropped and the outcome model reduces to the no-correction submodel. Default is 1e-6.

## Details

The estimator proceeds in three stages.

**Stage 1 - Visiting Process.** The visit intensity is

$$\lambda_i(t) = \eta_i \exp(\gamma^\top V_i) \lambda_0(t),$$

where  $\eta_i$  is a shared frailty. The coefficient vector  $\gamma$  is estimated by solving the Andersen-Gill partial-likelihood score equation

$$\sum_i \int [V_i - \bar{V}(\gamma; t)] dN_i(t) = 0,$$

where  $\bar{V}(\gamma; t)$  is the risk-set-weighted mean of  $V$ . Only visit rows enter the score; placeholder rows for zero-visit subjects are excluded. Standard errors are obtained from the inverse of the observed information matrix.

**Stage 2 - Observation Process.** The observation indicator is modelled by

$$P(R_{ij} = 1 \mid V_i) = g^{-1}(\alpha^\top V_i^*),$$

where  $V_i^* = (1, V_i)$  and  $g$  is the probit (default) or logistic link. The coefficient vector  $\alpha$  is estimated by `glm` and the model-based variance-covariance matrix is returned.

**Stage 3 - Longitudinal Outcome.** An empirical Bayes estimator of the latent loading  $B_i$  is constructed as

$$\hat{B}_i = \frac{1 + n_i \hat{\sigma}^2}{1 + \hat{\Lambda}_i \hat{\sigma}^2} - 1,$$

where  $n_i$  is the visit count and  $\hat{\Lambda}_i = \exp(\hat{\gamma}^\top V_i) \hat{\Lambda}_0(C)$ . The outcome model

$$Y_i(t) = \beta^\top X_i(t) + \theta^\top \hat{B}_i Q_i^*(t) + \varepsilon_i(t)$$

is estimated by solving the risk-set-centred estimating equation

$$\sum_{i,j} (M_{ij} - \bar{M})(Y_{ij} - M_{ij}^\top \beta^*) = 0,$$

with  $M_{ij} = (X_{ij}, \hat{B}_i Q_{ij}^*)$  and the centering weighted by  $n_i \hat{w}_i$  where  $\hat{w}_i$  is the predicted observation probability. The equation is linear in  $\beta^*$  and is solved directly by `solve`, with a Moore-Penrose pseudoinverse fallback (via `ginv`) on singular systems. A clustered sandwich estimator

$$\hat{V} = B^{-1} \left( \sum_i A_i A_i^\top \right) B^{-\top}, \quad B = H^\top M, \quad A_i = \sum_j H_{ij} (Y_{ij} - M_{ij}^\top \hat{\beta}^*)$$

is returned. This variance is conditional on the Stage 1 nuisance estimates and does not propagate uncertainty from  $(\hat{\gamma}, \hat{\sigma}^2, \hat{\Lambda}_0)$ .

If the estimated frailty variance  $\hat{\sigma}^2$  falls below `sigma_sq_floor`, the frailty-compensation block is dropped and the outcome model reduces to the no-correction submodel with only `long_covs`; `theta_hat` and `se_theta` are then returned as NA.

## Value

An object of class "EHRJoint", a named list with three components, each itself a named list:

`visiting_process` A list with elements:

- `gamma_hat` Named numeric vector of visiting-process regression coefficients.
- `se_gamma` Named numeric vector of model-based standard errors for `gamma_hat`, from the inverse observed information matrix.
- `vcov_gamma` Numeric matrix; the model-based variance-covariance matrix for `gamma_hat`.
- `converged` Logical; TRUE if `nleqslv` returned a successful termination code.

`observation_process` A list with elements:

- `alpha_hat` Named numeric vector of observation-process GLM coefficients (including intercept).
- `se_alpha` Named numeric vector of model-based standard errors for `alpha_hat`.
- `vcov_alpha` Numeric matrix; the model-based variance-covariance matrix for `alpha_hat`.
- `link` Character string; the link function used ("probit" or "logit").

`longitudinal_outcome` A list with elements:

- `beta_hat` Named numeric vector of outcome fixed-effect coefficients.
- `theta_hat` Named numeric vector of latent-link coefficients (the part of the augmented coefficient vector multiplying the frailty-compensation covariates). All NA when the frailty variance is below `sigma_sq_floor`.
- `se_beta` Named numeric vector of clustered sandwich standard errors for `beta_hat`.
- `se_theta` Named numeric vector of clustered sandwich standard errors for `theta_hat`. All NA when the frailty variance is below `sigma_sq_floor`.
- `vcov_joint` Numeric matrix; the full clustered sandwich variance-covariance matrix for the augmented coefficient vector `c(beta_hat, theta_hat)`.
- `Lambda_C` Numeric scalar; Breslow estimate of the cumulative baseline visiting hazard over the full follow-up window.
- `sigma_sq` Numeric scalar; method-of-moments estimate of the frailty variance  $\sigma^2$ , truncated below at zero.
- `sigma_sq_raw` Numeric scalar; the untruncated method-of-moments estimate of the frailty variance.

use\_B Logical; TRUE if the frailty-compensation covariates were retained in the outcome model ( $\text{sigma\_sq} > \text{sigma\_sq\_floor}$ ), FALSE otherwise.

cond\_LHS Numeric scalar; the L2 condition number of the Stage 3 estimating-equation left-hand-side matrix. Values larger than  $1e10$  trigger a warning and a pseudoinverse fallback.

call The matched call.

### Note

This function requires the **nleqslv** and **MASS** packages.

### References

Du, J., Shi, X., and Mukherjee, B. (2025). A new statistical approach for joint modeling of longitudinal outcomes measured in electronic health records with clinically informative presence and observation processes. arXiv:2410.13113. doi:10.48550/arXiv.2410.13113.

### See Also

[Joint\\_modeling\\_visiting\\_and\\_longitudinal\\_Liang](#) for a related joint model without an observation-process component; [CIMEHR](#) for the three-stage CIMEHR estimator with a full shared latent structure; [bootstrap](#) for bootstrap supplementation of the analytic standard errors.

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 60, ]
fit <- EHRJoint(
  dat,
  outcome      = "log_HbA1c",
  visit_covs   = c("Age", "NSES"),
  long_covs    = c("Age", "NSES"),
  random_covs  = "NSES"
)
```

---

extract\_coefficient     *Extract One or More Coefficients from a Fitted Method*

---

### Description

This helper returns estimates from a requested model stage. Use `parameter = NULL` to return all coefficients from that stage, or set `parameter` to a coefficient name such as "Age".

**Usage**

```
extract_coefficient(
  object,
  stage = c("outcome", "visiting", "observation"),
  parameter = NULL,
  ...
)

coef_stage(
  object,
  stage = c("outcome", "visiting", "observation"),
  parameter = NULL,
  ...
)
```

**Arguments**

object	A fitted CIMEHR package model object.
stage	One of "outcome", "visiting", or "observation".
parameter	Optional character vector of coefficient names to extract.
...	Ignored.

**Value**

A named numeric vector.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(data = dat, y_col = "log_HbA1c",
              covars_visit_XV = "Age",
              covars_outcome_fixed_XY = "Age",
              covars_obs_fixed_XO = "Age")
extract_coefficient(fit, stage = "outcome")
```

---

Inverse\_intensity\_rate\_ratio\_balancing

*Inverse Intensity Rate Ratio (IIRR) Estimator with Balancing Weights*

---

**Description**

Estimates marginal regression coefficients for a longitudinal outcome using the balancing-weights approach of Yiu and Su (2024). The visiting-process intensity is modelled via a partial-likelihood score equation, and the outcome coefficients are obtained from weighted estimating equations using two alternative weight sets: (i) balancing weights that satisfy covariate-balance constraints in the

spirit of entropy balancing, and (ii) conventional Maximum Likelihood Estimation (MLE) based inverse intensity weights.

A sensitivity parameter  $\phi$  allows the analyst to explore departures from the assumption that the visiting process is fully captured by the observed covariates.

## Usage

```
Inverse_intensity_rate_ratio_balancing(
  data,
  id = "id",
  time = "time",
  outcome = "Y",
  obs_indicator = "R",
  visit_covs = NULL,
  balance_covs = NULL,
  formula = NULL,
  phi = 0
)
```

## Arguments

<code>data</code>	A <code>data.frame</code> in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates.
<code>id</code>	Character string giving the name of the subject identifier column. Default is "id".
<code>time</code>	Character string giving the name of the visit-time column. Default is "time".
<code>outcome</code>	Character string giving the name of the outcome column. Default is "Y".
<code>obs_indicator</code>	Character string giving the name of the binary observation indicator column (1 = outcome observed at visit). Default is "R".
<code>visit_covs</code>	Character vector of covariate column names for the visiting-process intensity model. Default is NULL.
<code>balance_covs</code>	Character vector of covariate column names for which covariate-balance constraints are imposed. If NULL (the default), <code>visit_covs</code> is used.
<code>formula</code>	An optional <code>formula</code> specifying the design matrix for the longitudinal outcome model. When NULL (the default), the formula is constructed automatically as <code>outcome ~ visit_covs + time</code> from the column-name arguments. A user-supplied formula overrides this (e.g., <code>Y ~ Age + Gender + time</code> for explicit covariate sets).
<code>phi</code>	Numeric scalar; sensitivity parameter controlling departures from the semiparametric visiting-process model. When $\phi = 0$ (the default), the estimator coincides with the standard IIRR assumption. Non-zero $\phi$ tilts the balancing weights to explore sensitivity to unmeasured confounding of the visit process.

## Details

The estimator proceeds in four steps:

**Step 1 – Visiting-Process Coefficients.** The partial-likelihood score equation for the visiting-process coefficients  $\gamma$  is solved with `nleqslv`:

$$\sum_j [Z_{ij} - \bar{Z}(\gamma; t_j)] = 0.$$

**Step 2 – MLE Weights.** The MLE-based inverse intensity weight for observation  $(i, j)$  is

$$w_i^{\text{MLE}} = \frac{1}{\exp(\hat{\gamma}^\top Z_i) \hat{\Lambda}_0(C)},$$

where  $\hat{\Lambda}_0(C)$  is the Breslow cumulative baseline hazard estimate.

**Step 3 – Balancing Weights.** The balancing weights  $w_i^{\text{bal}}$  are found by solving a constrained entropy-minimisation problem:

$$\min \sum_i w_i \log(w_i) \quad \text{s.t.} \quad \sum_i w_i X_i = \bar{X},$$

where  $\bar{X}$  is the unweighted covariate mean and the constraint set is determined by `balance_covs`. When  $\phi \neq 0$ , the constraints are tilted via the sensitivity parameter. The dual problem is solved with `nleqslv` using Broyden's method.

**Step 4 – Outcome Estimation.** The outcome coefficients  $\beta$  are estimated by weighted least squares using both the balancing weights (`beta_bal`) and the MLE weights (`beta_mle`).

S3 methods `print` and `summary` are available for objects of class "Inverse\_intensity\_rate\_ratio\_balancing".

## Value

An object of class "Inverse\_intensity\_rate\_ratio\_balancing", which is a named list with the following components:

`beta_bal` Named numeric vector of outcome regression coefficients estimated with the proposed balancing weights.

`beta_mle` Named numeric vector of outcome regression coefficients estimated with conventional MLE-based inverse intensity weights (for comparison).

`gamma_hat` Named numeric vector of visiting-process regression coefficients, estimated by solving the partial-likelihood score equation with `nleqslv`.

`balancing_weights` Numeric vector of per-observation balancing weights.

`mle_weights` Numeric vector of per-observation MLE-based inverse intensity weights.

`breslow` Numeric scalar; Breslow estimate of the cumulative baseline hazard  $\Lambda_0(C)$ .

`constraints` Numeric vector of constraint residuals (ideally near zero when balance is achieved).

`phi` Numeric scalar echoing the sensitivity parameter.

`convergence` A list with elements:

`gamma` Integer; convergence code from `nleqslv` for the visiting-process estimation (1 = converged).

`balancing` Integer; convergence code from the balancing-weights optimisation (1 = converged).

**Note**

This function requires the **nleqslv** package.

**References**

Sean Yiu and Li Su (2025). Accommodating informative visit times for analysing irregular longitudinal data: a sensitivity analysis approach with balancing weights estimators. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 74(3), 824–843. doi:10.1093/jrssc/qlaf002.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_balancing(
  dat,
  outcome = "log_HbA1c",
  visit_covs = "Age"
)
```

---

Inverse\_intensity\_rate\_ratio\_weighting

*Inverse Intensity Rate Ratio (IIRR) Weighting Estimator*

---

**Description**

Estimates marginal regression coefficients for a longitudinal outcome by weighting each observation by the inverse of its estimated visit intensity. The visiting process is modelled via a Cox-type partial likelihood score equation, and the outcome coefficients are obtained by solving inverse-intensity-weighted estimating equations.

Only rows with observed outcomes ( $R == 1$  or  $R$  is NA) are used in both the visiting-process and outcome estimation.

**Usage**

```
Inverse_intensity_rate_ratio_weighting(
  data,
  id = "id",
  time = "time",
  outcome = "Y",
  obs_indicator = "R",
  visit_covs = NULL,
  formula = NULL
)
```

**Arguments**

data	A data.frame in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates.
id	Character string giving the name of the subject identifier column. Default is "id".
time	Character string giving the name of the visit-time column. Default is "time".
outcome	Character string giving the name of the outcome column. Default is "Y".
obs_indicator	Character string giving the name of the binary observation indicator column (1 = outcome observed at visit). Default is "R".
visit_covs	Character vector of covariate column names for the visiting-process intensity model. These enter the exponent of the multiplicative intensity. Default is NULL.
formula	An optional formula specifying the design matrix for the longitudinal outcome model. When NULL (the default), the formula is constructed automatically as <code>outcome ~ visit_covs + time</code> from the column-name arguments. A user-supplied formula overrides this (e.g., <code>Y ~ Age + Gender + time</code> for explicit covariate sets).

**Details**

The estimator proceeds in three steps:

**Step 1.** The visiting-process coefficients  $\gamma$  are estimated by solving the partial-likelihood score equation

$$\sum_j [Z_{ij} - \bar{Z}(\gamma; t_j)] = 0,$$

where  $\bar{Z}(\gamma; t) = \sum_i Z_i \exp(\gamma^\top Z_i) / \sum_i \exp(\gamma^\top Z_i)$  is the risk-set mean.

**Step 2.** Inverse intensity weights are computed as  $w_i = \exp(\hat{\gamma}^\top Z_i)$ .

**Step 3.** The outcome coefficients  $\beta$  are estimated by solving the weighted estimating equation

$$\sum_{i,j} w_i^{-1} X_{ij} (Y_{ij} - X_{ij}^\top \beta) = 0.$$

**Value**

An object of class "Inverse\_intensity\_rate\_ratio\_weighting", which is a named list with two components:

gamma\_hat Named numeric vector of visiting-process regression coefficients, estimated by solving the partial-likelihood score equation with `nleqslv`.

beta\_hat Named numeric vector of outcome model coefficients, estimated from inverse-intensity-weighted estimating equations solved with `nleqslv`.

**Note**

This function requires the `nleqslv` package.

No standard errors are returned. For inference, consider using `Inverse_intensity_rate_ratio_balancing` which provides balancing-weights estimates, or bootstrap the entire procedure.

## References

Petra Bůžková and Thomas Lumley (2007). Longitudinal data analysis for generalized linear models with follow-up dependent on outcome-related variables. *The Canadian Journal of Statistics*, 35(4), 485–500. doi:10.1002/cjs.5550350402.

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_weighting(
  dat,
  outcome = "log_HbA1c",
  visit_covs = "Age"
)
```

---

Joint\_modeling\_visiting\_and\_longitudinal\_Liang

*Joint Model for Visiting and Longitudinal (JMVL) Processes with Informative Presence (IP) Adjustment (Liang's Method)*

---

## Description

Fits a joint model for longitudinal data with an informative visiting process using the estimating-equations approach of Liang, Lu, and Ying (2009). The visiting-process intensity is modelled via a Cox-type partial-likelihood score equation, and the longitudinal outcome is estimated by solving Liang-type risk-set-centred estimating equations that incorporate an empirical Bayes estimate of the shared latent factor. Analytic standard errors are returned for both the visiting process (model-based, from the inverse information matrix) and the outcome (clustered sandwich, conditional on the Stage 1 nuisance estimates).

Unlike [EHRJoint](#), this method does not model a separate observation process; it assumes that the outcome is always observed at visits (or equivalently, filters to rows with R == 1 or R is NA).

## Usage

```
Joint_modeling_visiting_and_longitudinal_Liang(
  data,
  id = "id",
  time = "time",
  outcome = "Y",
  obs_indicator = "R",
  visit_covs = c("Z", "X"),
  long_covs = c("Z", "X"),
  random_covs = "Z"
)
```

**Arguments**

data	A data.frame in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates.
id	Character string giving the name of the subject identifier column. Default is "id".
time	Character string giving the name of the visit-time column. Default is "time".
outcome	Character string giving the name of the outcome column. Default is "Y".
obs_indicator	Character string giving the name of the binary observation indicator column. Rows with R == 1 or R is NA are retained; all others are dropped. Default is "R".
visit_covs	Character vector of covariate column names for the visiting-process intensity model. Default is c("Z", "X").
long_covs	Character vector of covariate column names for the fixed-effect part of the longitudinal outcome model. Default is c("Z", "X").
random_covs	Character vector of covariate column names that interact with the latent factor in the outcome model. An intercept column is prepended automatically. Default is "Z".

**Details**

The model consists of two linked components:

**Visiting Process.** The visit intensity is

$$\lambda_i(t) = \eta_i \exp(\gamma^\top Z_i) \lambda_0(t),$$

where  $\eta_i$  is a shared frailty. The coefficient vector  $\gamma$  is estimated by solving the partial-likelihood score equation, the cumulative baseline hazard  $\Lambda_0(C)$  by the Breslow estimator, and the frailty variance  $\sigma^2$  by method of moments. Standard errors come from the inverse observed information matrix.

**Outcome Model.** An empirical Bayes estimator of the latent loading  $B_i$  is constructed as

$$\hat{B}_i = \frac{1 + n_i \hat{\sigma}^2}{1 + \hat{\Lambda}_i \hat{\sigma}^2} - 1,$$

and the outcome model

$$Y_i(t) = \beta^\top X_i(t) + B_i Z_i^*(t) + \varepsilon_i(t)$$

is estimated by solving risk-set-centred estimating equations

$$\sum_{i,j} [(M_{ij} - \bar{M}(t_j))] (Y_{ij} - M_{ij}^\top \beta^*) = 0,$$

where  $M_{ij} = (X_{ij}, \hat{B}_i Z_{ij}^*)$  and  $\bar{M}(t)$  is the risk-set-weighted mean. Because this equation is linear in  $\beta^*$ , it admits a closed-form solution  $\hat{\beta}^* = (H^\top M)^{-1} H^\top Y$  with  $H = M - \bar{M}$ .

**Sandwich Variance.** The clustered sandwich estimator for  $\hat{\beta}^*$  is

$$\hat{V} = B^{-1} \left( \sum_i A_i A_i^\top \right) B^{-\top}, \quad B = H^\top M, \quad A_i = \sum_j H_{ij} (Y_{ij} - M_{ij}^\top \hat{\beta}^*).$$

This variance is conditional on the Stage 1 nuisance estimates; it does not propagate estimation uncertainty from  $(\hat{\gamma}, \hat{\sigma}^2, \hat{\Lambda}_0)$ .

This differs from `EHRJoint` in that no observation-process weights enter the estimating equations.

### Value

An object of class `"Joint_modeling_visiting_and_longitudinal_Liang"`, a named list with the following components:

`gamma_hat` Named numeric vector of visiting-process regression coefficients, estimated by solving the partial-likelihood score equation with `nleqslv`.

`se_gamma` Named numeric vector of model-based standard errors for `gamma_hat`, obtained from the inverse observed information matrix.

`vcov_gamma` Numeric matrix; the model-based variance-covariance matrix for `gamma_hat`.

`Lambda_C_est` Numeric scalar; Breslow estimate of the cumulative baseline hazard  $\Lambda_0(C)$  over the full follow-up window.

`sigma_sq_hat` Numeric scalar; method-of-moments estimate of the frailty variance  $\sigma^2$ . Constrained to be non-negative.

`beta_hat` Named numeric vector of outcome fixed-effect coefficients, estimated from the Liang-type estimating equations.

`se_beta` Named numeric vector of clustered sandwich standard errors for `beta_hat`. The sandwich estimator is conditional on the Stage 1 estimates  $(\hat{\gamma}, \hat{\sigma}^2, \hat{\Lambda}_0)$ .

`theta_hat` Named numeric vector of latent-link coefficients (the part of the augmented coefficient vector multiplying  $\hat{B}_i Z_{ij}^*$ ). NULL when  $\hat{\sigma}^2 = 0$ .

`se_theta` Named numeric vector of clustered sandwich standard errors for `theta_hat`. NULL when  $\hat{\sigma}^2 = 0$ .

`vcov_outcome` Numeric matrix; the full clustered sandwich variance-covariance matrix for the augmented coefficient vector `c(beta_hat, theta_hat)`.

`call` The matched call.

### Note

This function requires the `nleqslv` and `MASS` packages.

### References

Liang, Y., Lu, W., and Ying, Z. (2009). Joint modeling and analysis of longitudinal data with informative observation times. *Biometrics*, 65(2), 377–384. doi:10.1111/j.15410420.2008.01104.x.

### See Also

`EHRJoint` for a three-stage joint model that additionally incorporates an observation-process component; `CIMEHR` for the three-stage CIMEHR estimator with a full shared latent structure.

**Examples**

```

dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Joint_modeling_visiting_and_longitudinal_Liang(
  dat,
  outcome = "log_HbA1c",
  visit_covs = c("Age", "NSES"),
  long_covs = c("Age", "NSES"),
  random_covs = "NSES"
)

```

---

Linear\_increment\_IP     *Linear Increment (LI) Imputation with Informative Presence (IP) Adjustment*

---

**Description**

Implements a two-stage procedure that first imputes missing outcome values using the linear increment (LI) model of Aalen and Gunnes (2010), then applies an informative-presence (IP) adjustment method to the completed dataset. The LI model is fitted with `slim` and handles irregular observation times naturally by modelling increments rather than levels.

Unlike `Multiple_imputation_IP`, this approach produces a single completed dataset (no pooling required), making it computationally faster but unable to account for imputation uncertainty in standard errors.

**Usage**

```

Linear_increment_IP(
  data,
  id = "id",
  time = "time",
  outcome = "Y",
  obs_indicator = "R",
  li_formula = NULL,
  li_covariance = c("randomwalk", "independence", "exchangeable"),
  li_limit = ~1,
  ip_method = c("Linear_mixed_model", "Joint_modeling_visiting_and_longitudinal_Liang",
    "Inverse_intensity_rate_ratio_weighting", "Inverse_intensity_rate_ratio_balancing",
    "Pairwise_likelihood"),
  visit_covs = NULL,
  outcome_covs = NULL,
  random_covs = NULL,
  outcome_formula = NULL,
  random_formula = NULL,
  ip_args = list()
)

```

**Arguments**

data	A <code>data.frame</code> in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates. Rows with NA in the time column are treated as unvisited time points.
id	Character string giving the name of the subject identifier column. Default is "id".
time	Character string giving the name of the visit-time column. Default is "time".
outcome	Character string giving the name of the outcome column. Default is "Y".
obs_indicator	Character string giving the name of the binary observation indicator column (1 = outcome observed, 0 = outcome missing at visit). Default is "R".
li_formula	A <a href="#">formula</a> for the linear increment model. The left-hand side should be the outcome variable. Default is NULL, which constructs $Y \sim \langle \text{visit\_covs} \rangle + \langle \text{time} \rangle$ .
li_covariance	Character string specifying the covariance structure for the linear increment model. One of: "randomwalk" (Default.) Random-walk covariance, recommended for irregular data. The variance of increments is proportional to the time gap. "independence" Independent increments. "exchangeable" Exchangeable correlation structure.
li_limit	A one-sided <a href="#">formula</a> for the limit specification passed to <code>slim</code> . Default is $\sim 1$ (constant limit).
ip_method	Character string specifying the IP adjustment method applied to the completed dataset. One of: "Linear_mixed_model" Standard linear mixed model via <a href="#">Linear_mixed_model</a> (no IP adjustment; included for comparison). "Joint_modeling_visiting_and_longitudinal_Liang" Joint model of Liang et al. (2009) via <a href="#">Joint_modeling_visiting_and_longitudinal_Liang</a> . "Inverse_intensity_rate_ratio_weighting" Inverse intensity rate ratio weighting via <a href="#">Inverse_intensity_rate_ratio_balancing</a> (using maximum likelihood estimation (MLE) weights). "Inverse_intensity_rate_ratio_balancing" Balancing-weights estimator of Yiu and Su (2024) via <a href="#">Inverse_intensity_rate_ratio_balancing</a> . "Pairwise_likelihood" Pairwise composite likelihood via <a href="#">Pairwise_likelihood</a> .
visit_covs	Character vector of covariate column names for the visiting-process model (used by Liang and IRR methods). Default is NULL.
outcome_covs	Character vector of covariate column names for the outcome model in the Liang method. Default is NULL (uses <code>visit_covs</code> ).
random_covs	Character vector of covariate column names with random effects in the Liang method. Default is NULL (uses the first element of <code>visit_covs</code> ).
outcome_formula	A <a href="#">formula</a> for the longitudinal outcome model, used by linear mixed model ( <code>Linear_mixed_model</code> ), inverse intensity rate ratio (IRR), and pairwise likelihood (PairLik) methods. Default is NULL, which constructs $Y \sim \langle \text{outcome\_covs} \rangle + \langle \text{time} \rangle$ .

random_formula	A one-sided <a href="#">formula</a> for the Linear_mixed_model random effects (e.g., ~ 1   id). Default is NULL, which constructs ~ 1   <id>.
ip_args	A named list of additional arguments passed to the IP method. Common options include: <b>For IIRR methods:</b> phi (numeric; sensitivity parameter, default 0). <b>For PairLik:</b> pair_sample (integer; number of pairs to sample). <b>For Linear_mixed_model:</b> visit_adjust ("OA" or "VA").

## Details

The procedure has two stages:

**Stage 1 — Linear Increment Imputation.** The linear increment model (Aalen and Gunnes, 2010) models the change in outcome between consecutive observations:

$$\Delta Y_{ij} = (t_j - t_{j-1})(\beta_0 + \beta^\top X_{ij}) + \varepsilon_{ij},$$

where  $\text{Var}(\varepsilon_{ij}) \propto (t_j - t_{j-1})$  under the random-walk covariance structure. The model is fitted with [slim](#) on the subset of rows with observed outcomes, and predictions are generated for rows where the outcome is missing but visit time is known.

**Stage 2 — IP Adjustment.** The completed dataset is passed to the selected IP method for coefficient estimation.

S3 methods `print` and `summary` are available for objects of class "Linear\_increment\_IP\_result".

## Value

An object of class "Linear\_increment\_IP\_result", which is a named list with the following components:

`coefficients` Named numeric vector of estimated regression coefficients from the IP method.

`se` Named numeric vector of standard errors (when available from the IP method; NULL otherwise).

`ip_method` Character string echoing the IP method used.

`ip_args` List echoing the additional IP arguments.

`li_model` A list with elements `formula`, `covariance`, and `limit` describing the LI imputation model.

`n_imputed` Integer; number of missing outcome values that were imputed.

`imputed_data` A data.frame; the completed dataset after LI imputation.

`call` The matched call.

## References

Odd O. Aalen and Nina Gunnes (2010). A dynamic approach for reconstructing missing longitudinal data using the linear increments model. *Biostatistics*, 11(3), 453–472. doi:10.1093/biostatistics/kxq014.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Linear_increment_IP(dat,
  outcome      = "log_HbA1c",
  outcome_formula = log_HbA1c ~ Age + time,
  random_formula = ~ 1 | id,
  ip_method    = "Linear_mixed_model",
  visit_covs   = "Age")
```

---

Linear\_mixed\_model      *Fit a Linear Mixed Model for Longitudinal Data*

---

**Description**

Fits a linear mixed-effects model (LMM) to longitudinal data using [lme](#), with optional filtering by an observation indicator and optional visit-count adjustment. This provides a convenient wrapper for standard LMM, Observation-Adjusted LMM (OA-LMM), and Visit-Adjusted LMM (VA-LMM) estimators commonly used as benchmarks in informative-visiting analyses.

**Usage**

```
Linear_mixed_model(
  data,
  id = "id",
  fixed = Y ~ 1,
  random = ~1 | id,
  obs_indicator = NULL,
  visit_adjust = NULL,
  method = "REML",
  optimizer = "nlminb"
)
```

**Arguments**

data	A <code>data.frame</code> in long (panel) format containing longitudinal data with one row per subject-time combination.
id	Character string giving the name of the subject identifier column. This is used for the visit-count adjustment (if requested) and is referenced in the random-effects formula. Default is "id".
fixed	A <a href="#">formula</a> specifying the fixed-effects structure (e.g., $Y \sim \text{Age} + \text{Gender} + \text{time}$ ). When <code>visit_adjust</code> is non-NULL, the term <code>n_visits</code> is appended automatically. Default is $Y \sim 1$ .
random	A one-sided <a href="#">formula</a> specifying the random-effects structure using <b>nlme</b> syntax (e.g., $\sim 1 + \text{Age} \mid \text{id}$ ). Default is $\sim 1 \mid \text{id}$ .

obs_indicator	Character string giving the name of the observation indicator column. Rows where this column equals $\emptyset$ are dropped before fitting. Set to NULL to use all rows. Default is NULL.
visit_adjust	Character string specifying the type of visit-count adjustment. One of: NULL (Default.) No adjustment; a standard LMM is fitted. "OA" Observed-adjusted: the cumulative visit count <code>n_visits</code> is computed <i>after</i> filtering by <code>obs_indicator</code> . This adjusts for the number of observed (retained) visits. "VA" Visit-adjusted: the cumulative visit count <code>n_visits</code> is computed <i>before</i> filtering by <code>obs_indicator</code> . This adjusts for the total number of visits regardless of whether the outcome was observed. When either "OA" or "VA" is specified, the variable <code>n_visits</code> is automatically added to the fixed formula.
method	Character string specifying the estimation method passed to <code>lme</code> . One of "REML" (restricted maximum likelihood; default) or "ML" (maximum likelihood).
optimizer	Character string specifying the optimiser for the <code>lmeControl</code> object. One of "nlminb" (default) or "optim".

## Details

This function is a convenience wrapper that:

1. Optionally computes a within-subject cumulative visit counter (`n_visits`) either before or after filtering by the observation indicator.
2. Filters out rows where `obs_indicator ==  $\emptyset$`  (if specified).
3. Removes incomplete cases.
4. Fits a linear mixed model with `lme` using the specified fixed and random formulas, estimation method, and optimiser.

The OA-LMM and VA-LMM adjustments were proposed as simple corrections for informative visit times. OA-LMM conditions on the number of visits that yielded observed outcomes, while VA-LMM conditions on the total number of visits. Neither fully accounts for the visiting process but can reduce bias compared to a naive LMM.

## Value

A numeric matrix of fixed-effect estimates, identical to the `tTable` component of `summary(lme(...))`. The matrix has one row per fixed effect and columns `Value`, `Std.Error`, `DF`, `t-value`, and `p-value`.

## References

- Benjamin A. Goldstein, Nrupen A. Bhavsar, Matthew Phelan, and Michael J. Pencina (2016). Controlling for informed presence bias due to the number of health encounters in an electronic health record. *American Journal of Epidemiology*, 184(11), 847–855. doi:10.1093/aje/kww112.
- Nan M. Laird and James H. Ware (1982). Random-effects models for longitudinal data. *Biometrics*, 38(4), 963–974.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Linear_mixed_model(dat, fixed = log_HbA1c ~ Age + time,
  random = ~ 1 | id, obs_indicator = "R")
print(fit)
```

---

method\_comparisons      *Fit and Compare Multiple Methods*

---

**Description**

Fits any combination of implemented CIMEHR package methods and returns process-specific comparison tables. Each table contains one row per method and parameter. For methods that do not estimate a requested process, the table contains an NA row for that process. When true values are available, bias and RMSE columns are included.

**Usage**

```
method_comparisons(
  data,
  methods = "all",
  method_args = list(),
  printSE = TRUE,
  print95CI = TRUE,
  true_values = NULL,
  true_value_verbose = TRUE,
  which_balancing = c("balancing", "mle"),
  default_formula = log_HbA1c ~ Age + time,
  default_random_formula = ~1 | id,
  stop_on_error = FALSE,
  ...
)

Linear_increment(...)

Multiple_imputation(...)
```

**Arguments**

data	Long-format analysis data. This can also be a <code>sim_CIMEHR</code> object returned by <code>sim_data_gen()</code> , in which case <code>data\$long_data</code> is used and true simulation parameters are detected automatically.
methods	Character vector of method names to fit, or "all". Available names are returned by <code>available_comparison_methods()</code> . The aliases "Linear_increment" and "Multiple_imputation" call <code>Linear_increment_IP()</code> and <code>Multiple_imputation_IP()</code> , respectively.



```

    true_value_verbose = FALSE
  )
  cmp$tables$outcome

```

---

Multiple\_imputation\_IP

*Multiple Imputation (MI) with Informative Presence (IP) Adjustment*

---

## Description

Implements a two-stage procedure that first imputes missing outcome values using a multilevel model via [mice](#), then applies an informative-presence (IP) adjustment method to each imputed dataset and pools results. The imputation model accounts for the clustered structure of longitudinal data using a two-level specification (random intercept and optional random slopes).

## Usage

```

Multiple_imputation_IP(
  data,
  id = "id",
  time = "time",
  outcome = "Y",
  obs_indicator = "R",
  m = 5,
  maxit = 50,
  impute_method = c("2l.pan", "2l.lmer", "2l.norm"),
  impute_fixed = NULL,
  impute_random = NULL,
  impute_include_R = TRUE,
  ip_method = c("Linear_mixed_model", "Joint_modeling_visiting_and_longitudinal_Liang",
    "Inverse_intensity_rate_ratio_weighting", "Inverse_intensity_rate_ratio_balancing",
    "Pairwise_likelihood"),
  visit_covs = NULL,
  outcome_covs = NULL,
  random_covs = NULL,
  outcome_formula = NULL,
  random_formula = NULL,
  ip_args = list(),
  pool_method = c("mean", "rubin")
)

```

## Arguments

data	A data.frame in long (panel) format with one row per subject-time combination. Must contain columns for subject identifier, visit time, outcome, observation indicator, and all specified covariates. Rows with NA in the time column are treated as unvisited time points.
------	---

<code>id</code>	Character string giving the name of the subject identifier column. Default is "id".
<code>time</code>	Character string giving the name of the visit-time column. Default is "time".
<code>outcome</code>	Character string giving the name of the outcome column. Default is "Y".
<code>obs_indicator</code>	Character string giving the name of the binary observation indicator column (1 = outcome observed, 0 = outcome missing at visit). Default is "R".
<code>m</code>	Integer; number of multiply imputed datasets. Default is 5.
<code>maxit</code>	Integer; maximum number of iterations for <code>mice</code> . Default is 50.
<code>impute_method</code>	Character string specifying the <b>mice</b> imputation method for the outcome. One of:

"2l.pan" (Default.) Two-level model with heterogeneous within-group variance, implemented via **pan**.

"2l.lmer" Two-level model using `lmer`.

"2l.norm" Two-level normal model.

<code>impute_fixed</code>	Character vector of covariate column names to include as fixed effects in the imputation model. Default is NULL, which uses <code>c(visit_covs, time)</code> . The imputation model for $Y$ is:
---------------------------	---

$$Y_{ij} = \beta_0 + \sum_k \beta_k X_{kij} + b_{0i} + \sum_l b_{li} Z_{lij} + \varepsilon_{ij},$$

where fixed effects include an intercept plus all variables in `impute_fixed`.

<code>impute_random</code>	Character vector of covariate column names to include as random slopes in the imputation model. Default is NULL (random intercept only). For example, <code>c("Age")</code> adds a random slope for Age: $b_{0i} + b_{1i} \text{Age}_{ij}$ .
----------------------------	--

<code>impute_include_R</code>	Logical; whether to include the observation indicator R as a predictor in the imputation model. Default is TRUE. Recommended because R may be informative about missing $Y$ values under missing at random (MAR).
-------------------------------	---

<code>ip_method</code>	Character string specifying the IP adjustment method applied to each completed dataset. One of: "Linear_mixed_model" Standard linear mixed model via <code>Linear_mixed_model</code> (no IP adjustment; for comparison). "Joint_modeling_visiting_and_longitudinal_Liang" Joint model of Liang et al. (2009) via <code>Joint_modeling_visiting_and_longitudinal_Liang</code> . "Inverse_intensity_rate_ratio_weighting" Inverse intensity rate ratio weighting via <code>Inverse_intensity_rate_ratio_balancing</code> (MLE weights). "Inverse_intensity_rate_ratio_balancing" Balancing-weights estimator of Yiu and Su (2024) via <code>Inverse_intensity_rate_ratio_balancing</code> . "Pairwise_likelihood" Pairwise composite likelihood via <code>Pairwise_likelihood</code> .
------------------------	---

<code>visit_covs</code>	Character vector of covariate column names for the visiting-process model (used by Liang and inverse intensity rate ratio methods). Default is NULL.
-------------------------	--

<code>outcome_covs</code>	Character vector of covariate column names for the outcome model in the Liang method. Default is NULL (uses <code>visit_covs</code> ).
---------------------------	--

random_covs	Character vector of covariate column names with random effects in the Liang method. Default is NULL (uses the first element of visit_covs).
outcome_formula	A <a href="#">formula</a> for the longitudinal outcome model, used by linear mixed model (Linear_mixed_model), inverse intensity rate ratio (IIRR), and pairwise likelihood (PairLik) methods. Default is NULL, which constructs $Y \sim \langle \text{outcome\_covs} \rangle + \langle \text{time} \rangle$ .
random_formula	A one-sided <a href="#">formula</a> for the Linear_mixed_model random effects. Default is NULL, which constructs $\sim 1 \mid \langle \text{id} \rangle$ .
ip_args	A named list of additional arguments passed to the IP method. Common options include:  <b>For IIRR methods:</b> phi (numeric; sensitivity parameter, default 0). balance_covs (character vector). <b>For PairLik:</b> pair_sample (integer; number of pairs to sample). family ("gaussian" or "binomial"). <b>For Linear_mixed_model:</b> method ("REML" or "ML"). optimizer ("nlminb" or "optim"). visit_adjust (NULL, "OA", or "VA").
pool_method	Character string specifying the method for pooling estimates across imputations. One of:  "mean" (Default.) Simple averaging of point estimates with Monte Carlo standard errors. "rubin" Rubin's rules: combines within-imputation and between-imputation variance for valid inference.

## Details

The procedure has three stages:

**Stage 1 — Multiple Imputation.** Missing outcome values (where  $R = 0$  but time is not NA) are imputed using a multilevel model via [mice](#). Unvisited rows (time is NA) are separated, imputation is performed on the visited subset, and unvisited rows are reattached afterwards. The predictor matrix is configured so that only the outcome is imputed and only the specified covariates are used as predictors.

**Stage 2 — IP-Adjusted Analysis.** Each completed dataset is analysed by the selected IP method. Failed fits are discarded with a warning.

**Stage 3 — Pooling.** Point estimates are combined across successful imputations. Under pool\_method = "rubin", Rubin's rules are applied: the total variance is  $T = \bar{W} + (1 + 1/m)B$ , where  $\bar{W}$  is the mean within-imputation variance and  $B$  is the between-imputation variance.

S3 methods print are available for objects of class "Multiple\_imputation\_IP\_result".

## Value

An object of class "Multiple\_imputation\_IP\_result", which is a named list with the following components:

coefficients Named numeric vector of pooled regression coefficients.

se Named numeric vector of pooled standard errors.

by\_imputation List of length  $m$ , each element a list with coef and (optionally) se from one imputed dataset.

ip\_method Character string echoing the IP method used.

ip\_args List echoing the additional IP arguments.

impute\_method Character string echoing the imputation method.

impute\_model A list with elements fixed, random, and include\_R describing the imputation model.

m Integer; number of successful imputations.

m\_requested Integer; number of imputations originally requested.

call The matched call.

## References

Donald B. Rubin (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons. doi:10.1002/9780470316696.

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Multiple_imputation_IP(dat,
  outcome      = "log_HbA1c",
  outcome_formula = log_HbA1c ~ Age + time,
  random_formula = ~ 1 | id,
  ip_method    = "Linear_mixed_model",
  visit_covs   = "Age",
  m = 1, maxit = 1)
```

---

Pairwise\_likelihood    *Pairwise Composite Likelihood (PCL) Generalized Linear Model (GLM) Estimation*

---

## Description

Fits a Generalized Linear Model (GLM) using Pairwise Composite Likelihood (PCL). For each pair of subjects, the contribution to the composite log-likelihood is based on the difference  $Y_i - Y_j$  (Gaussian family) or a logistic-style softplus function (binomial family). This approach yields consistent slope estimates without requiring a correctly specified within-cluster correlation structure, and is robust to informative cluster sizes.

The intercept is not identifiable in pairwise differences and is automatically removed if present in the formula.

**Usage**

```
Pairwise_likelihood(
  data,
  formula,
  id_col = "id",
  time_col = "time",
  y_col = "Y",
  family = c("gaussian", "binomial"),
  start = NULL,
  pair_sample = NULL,
  pair_seed = 1L,
  hessian = TRUE,
  verbose = FALSE
)
```

**Arguments**

<code>data</code>	A <code>data.frame</code> containing longitudinal or clustered data. Rows with missing values in the outcome, covariates, or identifier columns are removed.
<code>formula</code>	A <a href="#">formula</a> specifying the model (e.g., $Y \sim \text{Age} + \text{time}$ ). Must be supplied by the user; there is no default. If an intercept is present it is removed automatically with a message (when <code>verbose = TRUE</code> ).
<code>id_col</code>	Character string giving the name of the subject/cluster identifier column. Default is "id".
<code>time_col</code>	Character string giving the name of the time variable column. Default is "time".
<code>y_col</code>	Character string giving the name of the outcome column. Default is "Y".
<code>family</code>	Character string specifying the response distribution. One of: "gaussian" (Default.) Pairwise residual sum of squares: $\sum 0.5 (Y_i - Y_j - \mu_i + \mu_j)^2$ . "binomial" Logistic-style softplus: $\sum -\log(1 + \exp((Y_i - Y_j)(\mu_j - \mu_i)))$ . Requires $Y \in \{0, 1\}$ .
<code>start</code>	Optional numeric vector of starting values for the regression coefficients. Default is NULL, which uses Ordinary Least Squares (Gaussian) or GLM (binomial) estimates.
<code>pair_sample</code>	Optional positive integer; if the total number of subject pairs exceeds this value, a random subsample of pairs is used. Default is NULL (use all pairs).
<code>pair_seed</code>	Integer seed for reproducible pair subsampling. Default is 1.
<code>hessian</code>	Logical; if TRUE, the numeric Hessian is computed at the optimum via <a href="#">hessian</a> and used to obtain a naive variance–covariance matrix. Default is TRUE.
<code>verbose</code>	Logical; if TRUE, progress messages are printed. Default is FALSE.

**Details**

The pairwise composite likelihood objective aggregates contributions from all (or a subsample of) between-subject pairs. For each pair of subjects  $(i, j)$  and all cross-products of their observations, the contribution is:

**Gaussian:**

$$l_{ij}(\beta) = -\frac{1}{2} \sum_{s \in i} \sum_{t \in j} [(Y_s - Y_t) - (\mu_s - \mu_t)]^2,$$

where  $\mu_s = X_s^\top \beta$ . Minimising the total gives the PCL estimator.

**Binomial:**

$$l_{ij}(\beta) = \sum_{s \in i} \sum_{t \in j} -\log[1 + \exp((Y_s - Y_t)(\mu_t - \mu_s))].$$

The inner loops are implemented in C++ via **Rcpp** for efficiency. Optimization is performed with `optim` using BFGS.

S3 methods `print` and `summary` are available for objects of class "Pairwise\_likelihood".

**Value**

An object of class "Pairwise\_likelihood", which is a named list with the following components:

`call` The matched call.

`family` Character string echoing the response family.

`formula` The formula used (intercept removed if applicable).

`coef` Named numeric vector of estimated regression coefficients.

`loglik` Numeric scalar; composite log-likelihood at the optimum (Gaussian: negated pairwise residual sum of squares; binomial: pairwise softplus sum).

`converged` Logical; TRUE if `optim` converged.

`hessian` (When `hessian = TRUE`.) Numeric matrix; Hessian of the objective at the optimum.

`vcov_naive` (When `hessian = TRUE`.) Numeric matrix; inverse Hessian, a naive (model-based) variance-covariance estimator for `coef`.

`n` Integer; number of observations used.

`p` Integer; number of regression coefficients.

`G` Integer; number of unique subjects/clusters.

`k_pairs` Integer; number of subject pairs used.

`pair_sample` Integer or NULL; echoing the subsampling parameter.

**Note**

This function requires the **Rcpp** and **numDeriv** packages. The inline C++ functions `pw_total_gaussian_rcpp` and `pw_total_softplus_rcpp` are compiled on first use.

**References**

Weining Shen, Suyu Liu, Yong Chen, and Jing Ning (2019). Regression analysis of longitudinal data with outcome-dependent sampling and informative censoring. *Scandinavian Journal of Statistics*, 46(3), 831–847. doi:10.1111/sjos.12373.

Yong Chen, Jing Ning, and Chunyan Cai (2015). Regression analysis of longitudinal data with irregular and informative observation times. *Biostatistics*, 16(4), 727–739. doi:10.1093/biostatistics/kxv008.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30 &
  sim_ehr_data$R == 1 & !is.na(sim_ehr_data$R), ]
fit <- Pairwise_likelihood(dat,
  formula = log_HbA1c ~ Age + time,
  y_col = "log_HbA1c",
  pair_sample = 50, hessian = FALSE)
```

---

print.CIMEHR	<i>Print a Clinical Informative Missingness for Electronic Health Records (CIMEHR) Fit</i>
--------------	--

---

**Description**

Prints outcome fixed-effect estimates with sandwich standard errors and Wald confidence intervals.

**Usage**

```
## S3 method for class 'CIMEHR'
print(x, digits = 4, level = 0.95, ...)
```

**Arguments**

x	A CIMEHR object returned by <a href="#">CIMEHR</a> .
digits	Number of significant digits. Default is 4.
level	Confidence level for Wald intervals. Default is 0.95.
...	Ignored.

**Value**

x, invisibly.

**See Also**

[summary.CIMEHR](#), [CIMEHR](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data = dat,
  y_col = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age"
)
print(fit)
```

```
print.cimehr_bootstrap
      Print a Bootstrap Result
```

---

### Description

Prints bootstrap standard errors and percentile confidence intervals for outcome coefficients.

### Usage

```
## S3 method for class 'cimehr_bootstrap'
print(x, digits = 4, ...)
```

### Arguments

x	A cimehr_bootstrap object from <a href="#">bootstrap</a> .
digits	Number of significant digits. Default is 4.
...	Ignored.

### Value

x, invisibly.

### See Also

[summary.cimehr\\_bootstrap](#), [bootstrap](#)

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_weighting(dat,
      outcome = "log_HbA1c",
      visit_covs = "Age")
bs <- bootstrap(fit, data = dat, B = 3, seed = 1)
print(bs)
```

---

```
print.CIMEHR_timevarying_integral
      Print a Clinical Informative Missingness for Elec-
      tronic Health Records with Gauss-Hermite Quadrature
      (CIMEHR_timevarying_integral) Fit
```

---

## Description

Prints outcome fixed-effect estimates from the Gauss-Hermite quadrature variant. Standard errors are not available for this method.

## Usage

```
## S3 method for class 'CIMEHR_timevarying_integral'
print(x, digits = 4, level = 0.95, ...)
```

## Arguments

x	A CIMEHR_timevarying_integral object returned by <a href="#">CIMEHR_timevarying_integral</a> .
digits	Number of significant digits. Default is 4.
level	Confidence level for Wald intervals. Default is 0.95.
...	Ignored.

## Value

x, invisibly.

## See Also

[summary.CIMEHR\\_timevarying\\_integral](#), [CIMEHR\\_timevarying\\_integral](#)

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$tid <= 20, ]
fit <- CIMEHR_timevarying_integral(
  data = dat,
  y_col = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age",
  gh_points_U = 2,
  gh_points_q = 2
)
print(fit)
```

---

```
print.CIMEHR_timevarying_ou
    Print a Clinical Informative Missingness for Electronic Health
    Records with Ornstein-Uhlenbeck Pairwise Composite Likelihood
    (CIMEHR_timevarying_ou) Fit
```

---

### Description

Prints outcome fixed-effect estimates with robust standard errors from the Ornstein-Uhlenbeck pairwise composite likelihood variant.

### Usage

```
## S3 method for class 'CIMEHR_timevarying_ou'
print(x, digits = 4, level = 0.95, ...)
```

### Arguments

x	A CIMEHR_timevarying_ou object returned by <a href="#">CIMEHR_timevarying_ou</a> .
digits	Number of significant digits. Default is 4.
level	Confidence level for Wald intervals. Default is 0.95.
...	Ignored.

### Value

x, invisibly.

### See Also

[summary.CIMEHR\\_timevarying\\_ou](#), [CIMEHR\\_timevarying\\_ou](#)

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR_timevarying_ou(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age",
  pair_type      = "adjacent"
)
print(fit)
```

---

print.EHRJoint	<i>Print a Joint Estimation for Electronic Health Record (EHR) Longitudinal Processes (EHRJoint) Fit</i>
----------------	--

---

## Description

Prints outcome fixed-effect estimates from the joint model.

## Usage

```
## S3 method for class 'EHRJoint'  
print(x, digits = 4, level = 0.95, ...)
```

## Arguments

x	An EHRJoint object returned by <a href="#">EHRJoint</a> .
digits	Number of significant digits. Default is 4.
level	Confidence level for Wald intervals. Default is 0.95.
...	Ignored.

## Value

x, invisibly.

## See Also

[summary.EHRJoint](#), [EHRJoint](#)

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]  
fit <- EHRJoint(  
  dat,  
  outcome = "log_HbA1c",  
  visit_covs = c("Age", "NSES"),  
  long_covs = c("Age", "NSES"),  
  random_covs = "NSES"  
)  
print(fit)
```

---

```
print.Inverse_intensity_rate_ratio_balancing
      Print an Inverse Intensity Rate Ratio (IIRR) Estimator with Balancing
      Weights Fit
```

---

### **Description**

Prints outcome estimates from both balancing and maximum likelihood estimation (MLE) weights.

### **Usage**

```
## S3 method for class 'Inverse_intensity_rate_ratio_balancing'
print(x, digits = 4, level = 0.95, ...)
```

### **Arguments**

<code>x</code>	An <code>Inverse_intensity_rate_ratio_balancing</code> object.
<code>digits</code>	Number of significant digits. Default is 4.
<code>level</code>	Confidence level for Wald intervals. Default is 0.95.
<code>...</code>	Ignored.

### **Value**

`x`, invisibly.

### **See Also**

[summary.Inverse\\_intensity\\_rate\\_ratio\\_balancing](#), [Inverse\\_intensity\\_rate\\_ratio\\_balancing](#)

### **Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_balancing(
  dat,
  outcome = "log_HbA1c",
  visit_covs = "Age"
)
print(fit)
```

---

```
print.Inverse_intensity_rate_ratio_weighting
      Print an Inverse Intensity Rate Ratio (IIRR) Weighting Estimator Fit
```

---

### Description

Prints outcome fixed-effect estimates from the inverse intensity rate ratio weighting estimator.

### Usage

```
## S3 method for class 'Inverse_intensity_rate_ratio_weighting'
print(x, digits = 4, level = 0.95, ...)
```

### Arguments

x	An Inverse_intensity_rate_ratio_weighting object.
digits	Number of significant digits. Default is 4.
level	Confidence level for Wald intervals. Default is 0.95.
...	Ignored.

### Value

x, invisibly.

### See Also

[summary.Inverse\\_intensity\\_rate\\_ratio\\_weighting](#), [Inverse\\_intensity\\_rate\\_ratio\\_weighting](#)

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_weighting(
  dat,
  outcome = "log_HbA1c",
  visit_covs = "Age"
)
print(fit)
```

---

```
print.Joint_modeling_visiting_and_longitudinal_Liang
      Print a Joint Model for Visiting and Longitudinal (JMVL) Processes
      (Liang) Fit
```

---

**Description**

Prints outcome fixed-effect estimates from the Liang–Lu–Ying joint estimating equations approach.

**Usage**

```
## S3 method for class 'Joint_modeling_visiting_and_longitudinal_Liang'
print(x, digits = 4, level = 0.95, ...)
```

**Arguments**

<code>x</code>	A <code>Joint_modeling_visiting_and_longitudinal_Liang</code> object.
<code>digits</code>	Number of significant digits. Default is 4.
<code>level</code>	Confidence level for Wald intervals. Default is 0.95.
<code>...</code>	Ignored.

**Value**

`x`, invisibly.

**See Also**

[summary.Joint\\_modeling\\_visiting\\_and\\_longitudinal\\_Liang](#), [Joint\\_modeling\\_visiting\\_and\\_longitudinal\\_Liang](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Joint_modeling_visiting_and_longitudinal_Liang(
  dat,
  outcome      = "log_HbA1c",
  visit_covs   = c("Age", "NSES"),
  long_covs    = c("Age", "NSES"),
  random_covs  = "NSES"
)
print(fit)
```

---

```
print.Linear_increment_IP_result
```

*Print a Linear Increment (LI) Imputation with Informative Presence (IP) Adjustment Fit*

---

## Description

Prints outcome coefficients from the linear increment imputation with informative presence adjustment. Standard errors and confidence intervals are not reported; see the bootstrap note in the output.

## Usage

```
## S3 method for class 'Linear_increment_IP_result'  
print(x, digits = 4, ...)
```

## Arguments

x	A <code>Linear_increment_IP_result</code> object returned by <a href="#">Linear_increment_IP</a> .
digits	Number of significant digits. Default is 4.
...	Ignored.

## Value

x, invisibly.

## See Also

[summary.Linear\\_increment\\_IP\\_result](#), [Linear\\_increment\\_IP](#)

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]  
fit <- Linear_increment_IP(data = dat,  
                           outcome      = "log_HbA1c",  
                           outcome_formula = log_HbA1c ~ Age + time,  
                           random_formula  = ~ 1 | id,  
                           ip_method      = "Linear_mixed_model",  
                           visit_covs    = "Age")  
  
print(fit)
```

---

```
print.Linear_mixed_model_result
```

*Print a Linear Mixed Model for Longitudinal Data Result*

---

### Description

Prints the fixed-effects table from a linear mixed model fit.

### Usage

```
## S3 method for class 'Linear_mixed_model_result'  
print(x, digits = 4, level = 0.95, ...)
```

### Arguments

<code>x</code>	A <code>Linear_mixed_model_result</code> object returned by <a href="#">Linear_mixed_model</a> .
<code>digits</code>	Number of significant digits. Default is 4.
<code>level</code>	Confidence level for Wald intervals. Default is 0.95.
<code>...</code>	Ignored.

### Value

`x`, invisibly.

### See Also

[summary.Linear\\_mixed\\_model\\_result](#), [Linear\\_mixed\\_model](#)

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]  
fit <- Linear_mixed_model(dat, fixed = log_HbA1c ~ Age + time,  
  random = ~ 1 | id, obs_indicator = "R")  
print(fit)
```



---

```
print.Pairwise_likelihood
```

*Print a Pairwise Composite Likelihood (PCL) Generalized Linear Model (Pairwise\_likelihood) Fit*

---

## Description

Prints coefficients with naive standard errors (from the inverse Hessian) when available.

## Usage

```
## S3 method for class 'Pairwise_likelihood'  
print(x, digits = 4, level = 0.95, ...)
```

## Arguments

x	A Pairwise_likelihood object returned by <a href="#">Pairwise_likelihood</a> .
digits	Number of significant digits. Default is 4.
level	Confidence level for Wald intervals. Default is 0.95.
...	Ignored.

## Value

x, invisibly.

## See Also

[summary.Pairwise\\_likelihood](#), [Pairwise\\_likelihood](#)

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30 &  
  sim_ehr_data$R == 1 & !is.na(sim_ehr_data$R), ]  
fit <- Pairwise_likelihood(dat,  
  formula = log_HbA1c ~ Age + time,  
  y_col = "log_HbA1c",  
  pair_sample = 50, hessian = FALSE)  
print(fit)
```

---

```
print.Summary_stat_result
```

*Print a Summary-Statistic Regression for Longitudinal Data Result*

---

## Description

Prints coefficients from the summary-statistic regression.

## Usage

```
## S3 method for class 'Summary_stat_result'  
print(x, digits = 4, ...)
```

## Arguments

x	A <code>Summary_stat_result</code> object returned by <a href="#">Summary_stat</a> .
digits	Number of significant digits. Default is 4.
...	Ignored.

## Value

x, invisibly.

## See Also

[summary.Summary\\_stat\\_result](#), [Summary\\_stat](#)

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]  
fit <- Summary_stat(dat, outcome = "log_HbA1c", formula = ~ Age)  
print(fit)
```

---

sim\_data\_gen

*Simulate Longitudinal Electronic Health Record (EHR) Data with Informative Visiting and Observation Processes*

---

## Description

Generates synthetic longitudinal data from a joint model that includes an informative visiting process (Poisson with log-normal frailty), an informative observation process (probit with latent-factor-linked random effects), and a linear longitudinal outcome model with latent-factor-linked random effects. By default, no Ornstein-Uhlenbeck (OU) serial-correlation component is added; an optional OU component is enabled by setting  $\phi > 0$ .

Covariates are user-specified through the `covariates` argument as a named list, and per-process coefficients are passed as named numeric vectors keyed by those covariate names. This lets the same generator produce arbitrary covariate sets (e.g., generic Z/X, or EHR-style Age/Gender/NSES) without changes to the function. See the Examples section.

The data-generating mechanism matches the model specification in Yang, Shi, and Mukherjee (2026).

## Usage

```
sim_data_gen(
  n,
  time.start = 0,
  time.end = 60,
  seed = 123,
  covariates = NULL,
  gamma = c(intercept = -2.2, Z = 1, X = 1),
  sigma_zeta = 1,
  alpha = c(intercept = 1.5, Z = -0.5, X = -0.3),
  obs_random_covs = "Z",
  delta = c(intercept = -0.3, Z = -0.5),
  sigma_q = c(intercept = 0.3, Z = 0.2),
  phi = 0,
  beta = c(intercept = -2, Z = -0.5, X = 0.5),
  beta_t = 0.1,
  outcome_random_covs = "Z",
  theta = c(intercept = 0.5, Z = 0.3),
  Sigma_b = matrix(c(1, 0, 0, 2), 2, 2),
  sigma_y = 1,
  verbose = TRUE
)
```

## Arguments

<code>n</code>	Integer; number of subjects to simulate.
<code>time.start</code>	Numeric scalar; start of the observation window. Default is 0.
<code>time.end</code>	Numeric scalar; end of the observation window (the constant censoring time used for all subjects). Default is 60.
<code>seed</code>	Integer; random seed for reproducibility. Default is 123.
<code>covariates</code>	Named list specifying the baseline covariates to generate. Each element is itself a list with at least a type entry, which must be either "continuous" or "binary". Continuous covariates accept additional fields <code>dist</code> ("normal" or

"uniform"; default "normal"), mean/sd (for normal), min/max (for uniform), and standardize (logical; whether to z-standardize after generation, default FALSE). Binary covariates accept prob, the Bernoulli success probability (default 0.5). When covariates = NULL (the default), a minimal generic spec `list(Z = list(type = "binary", prob = 0.5), X = list(type = "continuous", dist = "normal", mean = 0, sd = 1))` is used so that the function works out of the box for the package's internal tests and existing examples.

gamma	Named numeric vector of visiting-process log-intensity coefficients. Must include an intercept entry; remaining names should match a subset of the covariates declared in covariates. Covariates declared but not named in gamma are excluded from the visiting model (coefficient zero).
sigma_zeta	Positive numeric scalar; standard deviation of the log-normal frailty on the log scale. Default is 1.0.
alpha	Named numeric vector of observation-process probit fixed-effect coefficients. Must include an intercept; remaining names should be a subset of covariates.
obs_random_covs	Character vector naming the covariates whose observation-process random slopes are linked to the shared latent factor. An intercept random effect is always included and need not be listed here. The default is "Z", matching the default delta and sigma_q dimensions; pass character(0) for an intercept-only random effect.
delta	Named numeric vector of latent-link coefficients for the observation process. Must include an intercept; the remaining names should match obs_random_covs.
sigma_q	Named numeric vector of residual standard deviations for the observation-process random effects. Must include an intercept; the remaining names should match obs_random_covs.
phi	Nonnegative numeric scalar controlling the optional OU serial-correlation component in the observation process. When $\phi > 0$ , the probit error is a stationary OU process with unit marginal variance and decay parameter $\phi$ . When $\phi = 0$ (the default), the probit error reduces to iid $N(0, 1)$ , giving the standard probit observation model.
beta	Named numeric vector of outcome fixed-effect coefficients (excluding the time slope). Must include intercept; remaining names should be a subset of covariates.
beta_t	Numeric scalar; coefficient for the linear time effect in the outcome model. Default is 0.1.
outcome_random_covs	Character vector naming the covariates whose outcome-model random slopes are linked to the shared latent factor. An intercept random effect is always included. The default is "Z".
theta	Named numeric vector of outcome latent-link coefficients. Must include intercept; remaining names should match outcome_random_covs.
Sigma_b	Numeric positive-definite matrix; the residual covariance of the outcome random effects. Dimension must equal $1 + \text{length}(\text{outcome\_random\_covs})$ . Default is the $2 \times 2$ matrix <code>matrix(c(1, 0, 0, 2), 2, 2)</code> for a single random slope.
sigma_y	Positive numeric scalar; residual standard deviation of the outcome measurement error. Default is 1.

verbose Logical; if TRUE (the default), print a short simulation summary via [message](#).

### Value

An object of class "sim\_CIMEHR", a named list with components long\_data (the long-format data.frame returned to the user; contains id, the declared covariates, time, R, Y, and C), latent (a subject-level data.frame with the latent factor  $U_i$ , frailty  $\eta_i$ , and random effects, retained for diagnostics and not merged into long\_data), prop\_no\_visit, obs\_rate, and params (echoes all data-generating parameters for use as true\_values in method\_comparisons()).

### Latent quantities

The latent variables  $U_i$ ,  $\eta_i$ ,  $q_i$ , and  $b_i$  are not returned in long\_data because they are unobservable in real EHR data; including them would obscure the analyst's view of what a user actually has. They remain available in the latent component for simulation diagnostics.

### See Also

[CIMEHR](#), [CIMEHR\\_timevarying\\_integral](#), [CIMEHR\\_timevarying\\_ou](#).

### Examples

```
# Generic two-covariate setup (back-compatible with earlier package versions)
dat <- sim_data_gen(n = 30, seed = 42)
head(dat$long_data)

# EHR-style multi-covariate setup
ehr_covs <- list(
  Age      = list(type = "continuous", dist = "uniform",
                 min = 18, max = 99, standardize = TRUE),
  Gender   = list(type = "binary", prob = 0.5),
  Marital  = list(type = "binary", prob = 0.4),
  Black    = list(type = "binary", prob = 0.30),
  Hispanic = list(type = "binary", prob = 0.20),
  NSES     = list(type = "continuous", dist = "normal",
                 mean = 0, sd = 1, standardize = FALSE)
)
dat_ehr <- sim_data_gen(
  n = 50, seed = 1, covariates = ehr_covs,
  gamma = c(intercept = -2.0, Age = 0.10, Marital = -0.05,
            Black = 0.15, Hispanic = -0.05, NSES = 0.12),
  alpha = c(intercept = 0.5, Age = 0.10, Gender = -0.10, NSES = 0.20),
  obs_random_covs = "NSES",
  delta  = c(intercept = -0.3, NSES = -0.5),
  sigma_q = c(intercept = 0.3, NSES = 0.2),
  beta   = c(intercept = -2.0, Age = 0.05, Gender = 0.10,
            Black = 0.15, Hispanic = 0.05, NSES = -0.25),
  outcome_random_covs = "NSES",
  theta  = c(intercept = 0.5, NSES = 0.3)
)
head(dat_ehr$long_data)
```

---

`sim_ehr_data`*Simulated Electronic Health Record (EHR) Longitudinal Data*

---

## Description

A pre-generated long-format longitudinal dataset with 2000 subjects, designed to look like an outpatient EHR extract for HbA1c. The dataset is one replicate from the package's Monte Carlo simulation framework (`simulations/simulation_study.R`), specifically the `continuous_phi0` scenario (focal  $Z = \text{NSES}$ ,  $\phi = 0$ ). It is generated by `sim_data_gen` using an EHR-style covariate spec (Age, Gender, Marital, Black, Hispanic, NSES) and the back-transform helper that adds the `log_HbA1c` outcome column. The latent factor  $U_i$ , frailty  $\eta_i$ , and per-process random effects that the simulation generates internally are not retained in this user-facing dataset, mirroring the situation in real EHR data.

## Usage

`sim_ehr_data`

## Format

A `data.frame` with the following columns:

**id** Integer subject identifier.

**Age** Standardized age (z-score). Used by the modeling functions.

**Gender** Binary indicator (1 = male, 0 = female).

**Marital** Binary indicator (1 = married, 0 = not).

**Black** Binary indicator (1 = Black race, 0 = otherwise).

**Hispanic** Binary indicator (1 = Hispanic ethnicity, 0 = otherwise).

**NSES** Continuous neighborhood socioeconomic status indicator (standardized median household income; mean 0, SD 1).

**time** Visit time on the 0 to 120 window. NA for subjects with no visits.

**R** Observation indicator at the visit (1 = HbA1c recorded, 0 = not recorded).

**log\_HbA1c** Natural log of HbA1c, the recommended outcome for analysis. Built from the standardized linear outcome  $Y$  from the data-generating model via  $\log\_HbA1c = \log(5.6) + 0.05 * Y$ . NA when  $R = 0$ .

**C** Censoring time (end of follow-up window, 120 for all subjects).

## Details

True parameter values used to generate the data are stored as an attribute and can be retrieved with `attr(sim_ehr_data, "true_params")`.

**Source**

Generated by [sim\\_data\\_gen](#) with the EHR-style covariate spec described in `data-raw/sim_ehr_data.R`, using the `continuous_phi0` scenario from `simulations/simulation_study.R` (`n = 2000`, `time.end = 120`, `phi = 0`, focal Z = NSES, seed = 21260509), then post-processed to add `log_HbA1c`.

**See Also**

[sim\\_data\\_gen](#) to generate new datasets with custom covariate sets and parameters.

**Examples**

```
data(sim_ehr_data)
head(sim_ehr_data)

# True data-generating parameters (named lists keyed by covariate name)
true <- attr(sim_ehr_data, "true_params")
true$beta
```

---

summary.CIMEHR	<i>Summarise a Clinical Informative Missingness for Electronic Health Records (CIMEHR) Fit</i>
----------------	--

---

**Description**

Displays estimates, sandwich standard errors, and Wald confidence intervals for the requested layer(s): visiting process, observation process, and/or longitudinal outcome.

**Usage**

```
## S3 method for class 'CIMEHR'
summary(
  object,
  level = 0.95,
  digits = 4,
  specify = c("all", "visiting", "observation", "outcome"),
  ...
)
```

**Arguments**

object	A CIMEHR object returned by <a href="#">CIMEHR</a> .
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; which process layer to display. One of "all" (default), "visiting", "observation", or "outcome".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.CIMEHR](#), [CIMEHR](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age"
)
summary(fit)
summary(fit, specify = "visiting")
summary(fit, specify = "observation")
summary(fit, specify = "outcome")
```

---

summary.cimehr\_bootstrap

*Summarise a Bootstrap Result*

---

**Description**

Displays bootstrap standard errors, percentile confidence intervals, and a summary of the replicate distribution for outcome coefficients.

**Usage**

```
## S3 method for class 'cimehr_bootstrap'
summary(object, digits = 4, ...)
```

**Arguments**

object	A cimehr_bootstrap object from <a href="#">bootstrap</a> .
digits	Number of significant digits. Default is 4.
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.cimehr\\_bootstrap](#), [bootstrap](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_weighting(dat,
                                             outcome = "log_HbA1c",
                                             visit_covs = "Age")
bs <- bootstrap(fit, data = dat, B = 3, seed = 1)
summary(bs)
```

---

```
summary.CIMEHR_timevarying_integral
```

*Summarise a Clinical Informative Missingness for Electronic Health Records with Gauss-Hermite Quadrature (CIMEHR\_timevarying\_integral) Fit*

---

**Description**

Displays estimates for the requested layer(s) of the Gauss-Hermite variant. Standard errors are not available for this method.

**Usage**

```
## S3 method for class 'CIMEHR_timevarying_integral'
summary(
  object,
  level = 0.95,
  digits = 4,
  specify = c("all", "visiting", "observation", "outcome"),
  ...
)
```

**Arguments**

object	A CIMEHR_timevarying_integral object returned by <a href="#">CIMEHR_timevarying_integral</a> .
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; which process layer to display. One of "all" (default), "visiting", "observation", or "outcome".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.CIMEHR\\_timevarying\\_integral](#), [CIMEHR\\_timevarying\\_integral](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$tid <= 20, ]
fit <- CIMEHR_timevarying_integral(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age",
  gh_points_U    = 2,
  gh_points_q    = 2
)
summary(fit)
summary(fit, specify = "outcome")
```

---

```
summary.CIMEHR_timevarying_ou
```

*Summarise a Clinical Informative Missingness for Electronic Health Records with Ornstein-Uhlenbeck Pairwise Composite Likelihood (CIMEHR\_timevarying\_ou) Fit*

---

**Description**

Displays estimates for the requested layer(s) of the Ornstein-Uhlenbeck variant, including the OU decay parameter. Robust standard errors are shown where available.

**Usage**

```
## S3 method for class 'CIMEHR_timevarying_ou'
summary(
  object,
  level = 0.95,
  digits = 4,
  specify = c("all", "visiting", "observation", "outcome"),
  ...
)
```

**Arguments**

object	A CIMEHR_timevarying_ou object returned by <a href="#">CIMEHR_timevarying_ou</a> .
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; which process layer to display. One of "all" (default), "visiting", "observation", or "outcome".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.CIMEHR\\_timevarying\\_ou](#), [CIMEHR\\_timevarying\\_ou](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR_timevarying_ou(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age",
  pair_type      = "adjacent"
)
summary(fit)
summary(fit, specify = "observation")
```

---

summary.EHRJoint	<i>Summarise a Joint Estimation for Electronic Health Record (EHR) Longitudinal Processes (EHRJoint) Fit</i>
------------------	--

---

**Description**

Displays estimates for the requested layer(s): visiting, observation, and/or outcome.

**Usage**

```
## S3 method for class 'EHRJoint'
summary(
  object,
  level = 0.95,
  digits = 4,
  specify = c("all", "visiting", "observation", "outcome"),
  ...
)
```

**Arguments**

object	An EHRJoint object.
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; which process layer to display. One of "all" (default), "visiting", "observation", or "outcome".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.EHRJoint](#), [EHRJoint](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- EHRJoint(
  dat,
  outcome      = "log_HbA1c",
  visit_covs   = c("Age", "NSES"),
  long_covs    = c("Age", "NSES"),
  random_covs  = "NSES"
)
summary(fit)
```

---

```
summary.Inverse_intensity_rate_ratio_balancing
```

*Summarise an Inverse Intensity Rate Ratio (IIRR) Estimator with Balancing Weights Fit*

---

**Description**

Displays visiting-process and outcome estimates under both balancing and MLE weights, including the sensitivity parameter.

**Usage**

```
## S3 method for class 'Inverse_intensity_rate_ratio_balancing'
summary(
  object,
  level = 0.95,
  digits = 4,
  specify = c("all", "visiting", "outcome"),
  ...
)
```

**Arguments**

object	An Inverse_intensity_rate_ratio_balancing object.
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; which process layer to display. One of "all" (default), "visiting", or "outcome".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.Inverse\\_intensity\\_rate\\_ratio\\_balancing](#), [Inverse\\_intensity\\_rate\\_ratio\\_balancing](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_balancing(
  dat,
  outcome = "log_HbA1c",
  visit_covs = "Age"
)
summary(fit)
summary(fit, specify = "outcome")
```

---

```
summary.Inverse_intensity_rate_ratio_weighting
```

*Summarise an Inverse Intensity Rate Ratio (IIRR) Weighting Estimator Fit*

---

**Description**

Displays visiting-process and outcome estimates.

**Usage**

```
## S3 method for class 'Inverse_intensity_rate_ratio_weighting'
summary(
  object,
  level = 0.95,
  digits = 4,
  specify = c("all", "visiting", "outcome"),
  ...
)
```

**Arguments**

object	An Inverse_intensity_rate_ratio_weighting object.
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; which process layer to display. One of "all" (default), "visiting", or "outcome".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.Inverse\\_intensity\\_rate\\_ratio\\_weighting](#), [Inverse\\_intensity\\_rate\\_ratio\\_weighting](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Inverse_intensity_rate_ratio_weighting(
  dat,
  outcome = "log_HbA1c",
  visit_covs = "Age"
)
summary(fit)
summary(fit, specify = "visiting")
```

---

```
summary.Joint_modeling_visiting_and_longitudinal_Liang
```

*Summarise a Joint Model for Visiting and Longitudinal (JMVL) Processes (Liang) Fit*

---

**Description**

Displays visiting-process and outcome estimates.

**Usage**

```
## S3 method for class 'Joint_modeling_visiting_and_longitudinal_Liang'
summary(
  object,
  level = 0.95,
  digits = 4,
  specify = c("all", "visiting", "outcome"),
  ...
)
```

**Arguments**

object	A <code>Joint_modeling_visiting_and_longitudinal_Liang</code> object.
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; which process layer to display. One of "all" (default), "visiting", or "outcome".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.Joint\\_modeling\\_visiting\\_and\\_longitudinal\\_Liang](#), [Joint\\_modeling\\_visiting\\_and\\_longitudinal\\_Liang](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Joint_modeling_visiting_and_longitudinal_Liang(
  dat,
  outcome      = "log_HbA1c",
  visit_covs   = c("Age", "NSES"),
  long_covs    = c("Age", "NSES"),
  random_covs  = "NSES"
)
summary(fit)
```

---

```
summary.Linear_increment_IP_result
```

*Summarise a Linear Increment (LI) Imputation with Informative Presence (IP) Adjustment Fit*

---

**Description**

Displays imputation model details and outcome coefficients. Standard errors and confidence intervals are not reported; see the bootstrap note in the output.

**Usage**

```
## S3 method for class 'Linear_increment_IP_result'
summary(object, digits = 4, specify = c("all", "outcome"), ...)
```

**Arguments**

object	A Linear_increment_IP_result object.
digits	Number of significant digits. Default is 4.
specify	Character; only "all" or "outcome" are meaningful for this two-stage fit (only the outcome stage is reported). Default is "all".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.Linear\\_increment\\_IP\\_result](#), [Linear\\_increment\\_IP](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Linear_increment_IP(data = dat,
  outcome      = "log_HbA1c",
  outcome_formula = log_HbA1c ~ Age + time,
  random_formula = ~ 1 | id,
  ip_method     = "Linear_mixed_model",
  visit_covs    = "Age")

summary(fit)
```

---

```
summary.Linear_mixed_model_result
```

*Summarise a Linear Mixed Model for Longitudinal Data Result*

---

**Description**

Displays the fixed-effects coefficient table.

**Usage**

```
## S3 method for class 'Linear_mixed_model_result'
summary(object, level = 0.95, digits = 4, specify = c("all", "outcome"), ...)
```

**Arguments**

object	A <code>Linear_mixed_model_result</code> object.
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; only "all" or "outcome" are meaningful for this single-layer fit. Default is "all".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.Linear\\_mixed\\_model\\_result](#), [Linear\\_mixed\\_model](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Linear_mixed_model(dat, fixed = log_HbA1c ~ Age + time,
  random = ~ 1 | id, obs_indicator = "R")
summary(fit)
```

---

```
summary.Multiple_imputation_IP_result
  Summarise a Multiple Imputation (MI) with Informative Presence (IP)
  Adjustment Fit
```

---

**Description**

Displays imputation settings and pooled outcome coefficients. Standard errors and confidence intervals are not reported; see the bootstrap note in the output.

**Usage**

```
## S3 method for class 'Multiple_imputation_IP_result'
summary(object, digits = 4, specify = c("all", "outcome"), ...)
```

**Arguments**

object	A Multiple_imputation_IP_result object.
digits	Number of significant digits. Default is 4.
specify	Character; only "all" or "outcome" are meaningful for this two-stage fit (only the pooled outcome stage is reported). Default is "all".
...	Ignored.

**Value**

object, invisibly.

**See Also**

[print.Multiple\\_imputation\\_IP\\_result](#), [Multiple\\_imputation\\_IP](#)

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Multiple_imputation_IP(data = dat,
  outcome = "log_HbA1c",
  outcome_formula = log_HbA1c ~ Age + time,
  random_formula = ~ 1 | id,
  ip_method = "Linear_mixed_model",
  visit_covs = "Age",
```

```
summary(fit)                                m = 1, maxit = 1)
```

---

```
summary.Pairwise_likelihood
```

*Summarise a Pairwise Composite Likelihood (PCL) Generalized Linear Model (Pairwise\_likelihood) Fit*

---

## Description

Displays model dimensions, convergence status, and the coefficient table with standard errors when available.

## Usage

```
## S3 method for class 'Pairwise_likelihood'
summary(object, level = 0.95, digits = 4, specify = c("all", "outcome"), ...)
```

## Arguments

object	A Pairwise_likelihood object.
level	Confidence level for Wald intervals. Default is 0.95.
digits	Number of significant digits. Default is 4.
specify	Character; only "all" or "outcome" are meaningful for this single-layer fit. Default is "all".
...	Ignored.

## Value

object, invisibly.

## See Also

[print.Pairwise\\_likelihood, Pairwise\\_likelihood](#)

## Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30 &
  sim_ehr_data$R == 1 & !is.na(sim_ehr_data$R), ]
fit <- Pairwise_likelihood(dat,
  formula = log_HbA1c ~ Age + time,
  y_col = "log_HbA1c",
  pair_sample = 50, hessian = FALSE)
summary(fit)
```

---

```
summary.Summary_stat_result
      Summarise a Summary-Statistic Regression for Longitudinal Data Re-
      sult
```

---

### Description

Displays the regression coefficient table.

### Usage

```
## S3 method for class 'Summary_stat_result'
summary(object, digits = 4, specify = c("all", "outcome"), ...)
```

### Arguments

object	A Summary_stat_result object.
digits	Number of significant digits. Default is 4.
specify	Character; only "all" or "outcome" are meaningful for this single-layer fit. Default is "all".
...	Ignored.

### Value

object, invisibly.

### See Also

[print.Summary\\_stat\\_result](#), [Summary\\_stat](#)

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Summary_stat(dat, outcome = "log_HbA1c", formula = ~ Age)
summary(fit)
```

---

summary\_observation     *Summarise the Observation Process of a Fitted Method*

---

**Description**

Summarise the Observation Process of a Fitted Method

**Usage**

```
summary_observation(object, ...)
```

**Arguments**

object	A fitted CIMEHR package model object.
...	Additional arguments passed to the model-specific summary method.

**Value**

The input object, invisibly.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data           = dat,
  y_col         = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_X0 = "Age"
)
summary_observation(fit)
```

---

summary\_outcome     *Summarise the Outcome Process of a Fitted Method*

---

**Description**

Summarise the Outcome Process of a Fitted Method

**Usage**

```
summary_outcome(object, ...)
```

**Arguments**

object            A fitted CIMEHR package model object.  
 ...              Additional arguments passed to the model-specific summary method.

**Value**

The input object, invisibly.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data           = dat,
  y_col         = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_XO = "Age"
)
summary_outcome(fit)
```

---

 Summary\_stat

*Summary-Statistic Regression for Longitudinal Data*


---

**Description**

Reduces longitudinal data to a single summary statistic per subject (e.g., mean, median, min, max of the outcome), then fits a cross-sectional linear model on the summarized data. This provides a simple baseline analysis that ignores the within-subject correlation structure and visit process entirely.

**Usage**

```
Summary_stat(
  data,
  id = "id",
  outcome = "Y",
  stat = "mean",
  formula = ~1,
  cov_stat = "mean"
)
```

**Arguments**

data            A data.frame in long (panel) format with one row per subject-time combination.  
 id             Character string giving the name of the subject identifier column. Default is "id".

outcome	Character string giving the name of the outcome column. Default is "Y".
stat	Character string specifying the summary statistic to apply to the outcome within each subject. One of "min", "max", "mean", or "median". Default is "mean".
formula	A one-sided <a href="#">formula</a> specifying the covariates for the linear model (e.g., $\sim X + Z$ ). The outcome (as summarized) is used as the response. Default is $\sim 1$ (intercept only).
cov_stat	Summary statistic to apply to time-varying covariates within each subject before fitting. Can be a single character string (applied to all covariates) or a named list mapping covariate names to statistic strings (e.g., <code>list(X = "mean", W = "median")</code> ). Covariates not named in the list default to "mean". Default is "mean".

### Details

The function proceeds as follows:

1. The outcome is aggregated to one value per subject using `stat` (e.g., subject-level means).
2. Each covariate in the formula is aggregated to one value per subject using `cov_stat`.
3. A standard linear model is fitted on the subject-level summary data.

This approach discards all within-subject temporal information and does not account for informative visiting or observation processes. It is included primarily as a naive benchmark for comparison with methods such as [CIMEHR](#).

### Value

A numeric matrix of regression coefficients, identical to the coefficients component of `summary(lm(...))`, with columns Estimate, Std. Error, t value, and  $\Pr(>|t|)$ .

### Examples

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- Summary_stat(dat, outcome = "log_HbA1c", formula = ~ Age)
```

---

summary\_visiting

*Summarise the Visiting Process of a Fitted Method*

---

### Description

Summarise the Visiting Process of a Fitted Method

### Usage

```
summary_visiting(object, ...)
```

**Arguments**

object            A fitted CIMEHR package model object.  
 ...              Additional arguments passed to the model-specific summary method.

**Value**

The input object, invisibly.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_XO = "Age"
)
summary_visiting(fit)
```

---

vcov.CIMEHR	<i>Variance-Covariance Matrix for Clinical Informative Missingness for Electronic Health Records (CIMEHR) Outcome Coefficients</i>
-------------	--

---

**Description**

Returns the sandwich variance-covariance matrix for the outcome model.

**Usage**

```
## S3 method for class 'CIMEHR'
vcov(object, ...)

## S3 method for class 'CIMEHR_timevarying_ou'
vcov(object, ...)
```

**Arguments**

object            A CIMEHR or CIMEHR\_timevarying\_ou object.  
 ...              Ignored.

**Value**

Numeric matrix.

**Examples**

```
dat <- sim_ehr_data[sim_ehr_data$id <= 30, ]
fit <- CIMEHR(
  data           = dat,
  y_col          = "log_HbA1c",
  covars_visit_XV = "Age",
  covars_outcome_fixed_XY = "Age",
  covars_obs_fixed_XO = "Age"
)
vcov(fit)
```

# Index

- \* **datasets**
  - sim\_ehr\_data, 61
- available\_comparison\_methods, 3
- bootstrap, 4, 23, 46, 63, 64
- CIMEHR, 4, 6, 7, 10, 12–14, 16, 17, 23, 31, 45, 60, 62, 63, 77
- CIMEHR\_timevarying\_integral, 4, 6, 10, 10, 17, 47, 60, 64, 65
- CIMEHR\_timevarying\_ou, 4, 6, 10, 13, 14, 48, 60, 65, 66
- coef.CIMEHR, 18
- coef.CIMEHR\_timevarying\_integral (coef.CIMEHR), 18
- coef.CIMEHR\_timevarying\_ou (coef.CIMEHR), 18
- coef\_stage (extract\_coefficient), 23
- confint.CIMEHR, 19
- confint.CIMEHR\_timevarying\_ou (confint.CIMEHR), 19
- EHRJoint, 4, 6, 20, 29, 31, 49, 67
- extract\_coefficient, 23
- formula, 25, 28, 33–35, 41, 43, 77
- ginv, 22
- glm, 21
- hessian, 43
- Inverse\_intensity\_rate\_ratio\_balancing, 4, 6, 24, 28, 33, 40, 50, 68
- Inverse\_intensity\_rate\_ratio\_weighting, 4, 6, 27, 51, 69
- Joint\_modeling\_visiting\_and\_longitudinal\_Liang, 4, 6, 23, 29, 33, 40, 52, 70
- Linear\_increment (method\_comparisons), 37
- Linear\_increment\_IP, 5, 32, 53, 71
- Linear\_mixed\_model, 33, 35, 40, 54, 71
- lm, 77
- lme, 35, 36
- lmeControl, 36
- lmer, 40
- message, 60
- method\_comparisons, 37
- mice, 39–41
- Multiple\_imputation (method\_comparisons), 37
- Multiple\_imputation\_IP, 5, 32, 39, 55, 72
- nleqslv, 22, 26, 28, 31
- nlminb, 9, 12
- optim, 16, 17, 44
- Pairwise\_likelihood, 33, 40, 42, 56, 73
- pbivnorm, 16
- print.CIMEHR, 45, 63
- print.cimehr\_bootstrap, 6, 46, 64
- print.CIMEHR\_timevarying\_integral, 47, 65
- print.CIMEHR\_timevarying\_ou, 48, 66
- print.EHRJoint, 49, 67
- print.Inverse\_intensity\_rate\_ratio\_balancing, 50, 68
- print.Inverse\_intensity\_rate\_ratio\_weighting, 51, 69
- print.Joint\_modeling\_visiting\_and\_longitudinal\_Liang, 52, 70
- print.Linear\_increment\_IP\_result, 53, 71
- print.Linear\_mixed\_model\_result, 54, 71
- print.Multiple\_imputation\_IP\_result, 55, 72

print.Pairwise\_likelihood, [56](#), [73](#)  
print.Summary\_stat\_result, [57](#), [74](#)

sim\_data\_gen, [10](#), [13](#), [17](#), [57](#), [61](#), [62](#)  
sim\_ehr\_data, [61](#)  
slim, [32–34](#)  
solve, [22](#)  
summary.CIMEHR, [45](#), [62](#)  
summary.cimehr\_bootstrap, [6](#), [46](#), [63](#)  
summary.CIMEHR\_timevarying\_integral,  
[47](#), [64](#)  
summary.CIMEHR\_timevarying\_ou, [48](#), [65](#)  
summary.EHRJoint, [49](#), [66](#)  
summary.Inverse\_intensity\_rate\_ratio\_balancing,  
[50](#), [67](#)  
summary.Inverse\_intensity\_rate\_ratio\_weighting,  
[51](#), [68](#)  
summary.Joint\_modeling\_visiting\_and\_longitudinal\_Liang,  
[52](#), [69](#)  
summary.Linear\_increment\_IP\_result, [53](#),  
[70](#)  
summary.Linear\_mixed\_model\_result, [54](#),  
[71](#)  
summary.Multiple\_imputation\_IP\_result,  
[55](#), [72](#)  
summary.Pairwise\_likelihood, [56](#), [73](#)  
summary.Summary\_stat\_result, [57](#), [74](#)  
summary\_observation, [75](#)  
summary\_outcome, [75](#)  
Summary\_stat, [57](#), [74](#), [76](#)  
summary\_visiting, [77](#)

vcov.CIMEHR, [78](#)  
vcov.CIMEHR\_timevarying\_ou  
(vcov.CIMEHR), [78](#)