

Package: cfrnow (via r-universe)

October 13, 2026

Title Real-Time Case Fatality Ratio via a Mixture-Cure Survival Model

Version 0.4.0

Description Estimates a real-time case fatality ratio (CFR) from line-list data using a Bayesian mixture-cure survival model, registered as an 'epidist' model type so the CFR and the onset-to-death delay both take 'brms' formulas. Each case is fatal with probability equal to the CFR and, when fatal, dies at an interval-censored onset-to-death delay; cases still alive at the observation cut-off are right-censored, which corrects the downward bias of the naive deaths / cases ratio in real time. The delay is co-estimated using the 'primarycensored' censored-distribution machinery.

License MIT + file LICENSE

URL <https://epiforecasts.io/cfrnow/>,
<https://github.com/epiforecasts/cfrnow>

BugReports <https://github.com/epiforecasts/cfrnow/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends distspec ($\geq 0.1.0$), R ($\geq 4.1.0$)

Imports brms, checkmate, epidist, lifecycle, posterior,
primarycensored ($\geq 1.5.2$), rlang, stats

Suggests cmdstanr, data.table, ggplot2, knitr, rmarkdown, testthat ($\geq 3.0.0$)

VignetteBuilder knitr

Remotes epinowcast/epidist, paul-buerkner/brms

Additional_repositories <https://stan-dev.r-universe.dev>

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libicu-dev

Repository <https://epiforecasts.r-universe.dev>
Date/Publication 2026-10-05 16:50:24 UTC
RemoteUrl <https://github.com/epiforecasts/cfrnow>
RemoteRef v0.4.0
RemoteSha df7b1da22c40a6beea1fd7983ae76860bfaac555

Contents

as_epidist_cure_model	2
fit_cfr	3
posterior_cumulative_incidence	5
posterior_prob_death	7
pp_check_cfr	8
prepare_cfr_data	10
simulate_linelist	11
summary.cfrnow_fit	13
Index	15

as_epidist_cure_model	<i>Build an epidist_cure_model from a line list or prepare_cfr_data() output</i>
-----------------------	--

Description

Produces the one-row-per-case data frame the model needs: y (a delay for deaths and recoveries, the follow-up time for survivors), an outcome code, and the primary (pwindow) and secondary (swindow) censoring widths. When the input carries timed recoveries the result is flagged for the two-outcome fit via a use_recovery attribute.

Usage

as_epidist_cure_model(data)

Arguments

data	Either the list returned by prepare_cfr_data() or a data frame already carrying y, outcome, pwindow and swindow.
------	--

Value

A data frame of class epidist_cure_model.

fit_cfr

*Fit the real-time mixture-cure CFR model***Description**

Estimates a real-time case fatality ratio from line-list data. Each case is fatal with probability prob and, when fatal, dies at an onset-to-death delay; cases still alive at the cut-off are right-censored, correcting the downward bias of the naive deaths / cases ratio. The delay and the prob prior are given as **distspec** distributions.

Usage

```
fit_cfr(
  data,
  delay = LogNormal(meanlog = Normal(2, 1), sdlog = Normal(0.5, 0.3)),
  prob_prior = Beta(1, 1),
  recovery_delay = NULL,
  loss_prior = NULL,
  formula = mu ~ 1,
  ...,
  cfr_prior = lifecycle::deprecated()
)
```

Arguments

data	Output of <code>prepare_cfr_data()</code> , or an <code>epidist_cure_model</code> / data frame with y, outcome, pwindow, swindow.
delay	Onset-to-death delay as a distspec distribution (<code>distspec::LogNormal()</code> , <code>distspec::Gamma()</code> or <code>distspec::Weibull()</code>) whose native parameters are fixed numbers or <code>Normal()</code> priors. A max (e.g. <code>LogNormal(Normal(2.4, 0.2), Normal(0.5, 0.15), max = 30)</code>) truncates the delay there, as it does everywhere else in distspec : recorded delays then run from 0 to max - 1 days.
prob_prior	Prior on prob as a <code>distspec::Beta()</code> . Defaults to <code>Beta(1, 1)</code> .
recovery_delay	Optional onset-to-recovery delay (same form as delay) for the two-outcome fit; may use a different family from delay.
loss_prior	Optional prior on the probability that a case is lost to follow-up, so its outcome never reaches the line list. A <code>distspec::Beta()</code> gives one probability whatever the outcome; a list with death and recovery entries, each a <code>Beta()</code> or a fixed number, gives them their own (e.g. <code>list(death = 0, recovery = Beta(1, 1))</code> where every death is recorded). Needs recorded recoveries and a <code>recovery_delay</code> . <code>NULL</code> (the default) assumes every outcome is eventually recorded.
formula	A brms formula for the delay location mu and, optionally, prob (<code>prob ~ ...</code>). Defaults to <code>mu ~ 1</code> . <code>prob_prior</code> normally lands on the prob intercept; when the prob formula drops the intercept (e.g. <code>prob ~ 0 + group</code> , one logit-probability

per group) it is placed on those coefficients instead. It then applies to every prob coefficient, so an intercept-free formula should carry only factor terms. A `cfr ~ ...` sub-formula is still accepted and translated to `prob ~ ...` with a deprecation warning.

... Passed to `epidist::epidist()` and on to `brms::brm()` (e.g. chains, iter, backend, seed).

`cfr_prior` **[Deprecated]** Use `prob_prior` instead.

Details

The delay's native parameters may each be a fixed number (held fixed; fixing the whole delay gives the Ghani/Nishiura estimator) or a `Normal()` prior (co-estimated). The family (`LogNormal()`, `Gamma()` or `Weibull()`) sets the delay distribution. `prob_prior` is a `Beta()`; it matters because `prob` is weakly identified early on (`Beta(1, 1)` is uniform, `Beta(1, 9)` favours a low probability, `Beta(6.6, 13.4)` suits a high-fatality pathogen).

`loss_prior` adds a probability that a case is lost to follow-up, so its outcome never reaches the line list. A recorded death or recovery then also says the case was kept, and a case still unresolved at the cut-off was either lost or genuinely unresolved. Without it, a case unresolved far longer than the delays allow has almost no probability, which pulls the delay's tail out or stops the sampler from starting.

Whether loss can be estimated depends on how it relates to the outcome. One probability for both (`loss_prior = Beta(1, 1)`) is identified by the recorded deaths, the recorded recoveries and the cases that stay unresolved, and it assumes a lost case's CFR matches everyone else's. Fixing one half (`list(death = 0, recovery = Beta(1, 1))`, where a death is always written down and a discharge may not be) is identified too. Estimating both leaves one parameter unidentified: the priors decide the split and `fit_cfr()` warns, so treat that fit as a sensitivity analysis. A `loss_prior` needs recorded recoveries and a `recovery_delay` to time them, because only timed recoveries separate loss from the outcome probability; without them, `fit_cfr()` stops.

Where a case has a date it was last known unresolved, censoring it there through `prepare_cfr_data()`'s `last_contact_date` uses that timing instead of inferring the loss.

A delay's `max` truncates it: the likelihood renormalises the distribution over $[0, \text{max})$ days, matching how **distspec** discretises the same object, so a delay recorded as `max` days or longer has no probability. The estimated parameters still describe the untruncated family, which is what `summary()` reports as `delay_mean` and `delay_sd`. A recorded delay outside the bound, or a case unresolved for as long as every bound that could still apply to it, cannot come from such a model, so `fit_cfr()` stops rather than letting the sampler fail.

The model is fitted through `epidist::epidist()`, so covariates (or a smooth time effect) can be put on `prob` or the delay through formula, e.g. `formula = brms::bf(mu ~ 1, prob ~ age)`. When the line list carries recovery dates, pass a `recovery_delay` to fit the two-outcome model that also times recoveries.

`prob` is a case fatality ratio when the line list runs from onset to death, as `prepare_cfr_data()` expects; fed a line list with a different outcome (e.g. hospitalisation), the same model fits that outcome's probability instead.

Value

A `brmsfit` with class `cfrnow_fit`; summarise with `summary()`.

See Also

Other fit: [posterior_cumulative_incidence\(\)](#), [posterior_prob_death\(\)](#), [pp_check_cfr\(\)](#), [summary.cfrnow_fit\(\)](#)

Examples

```
## Not run:
ll <- simulate_linelist(n = 500, cfr = 0.4, delay = LogNormal(2.4, 0.5))
d <- prepare_cfr_data(ll, obs_time = max(ll$onset_date) - 5)
otd <- LogNormal(meanlog = Normal(2.41, 0.2), sdlog = Normal(0.51, 0.15))

# co-estimated delay
fit <- fit_cfr(d, delay = otd, prob_prior = Beta(1, 1), backend = "cmdstanr")
summary(fit)

# fixed delay (Ghani/Nishiura), a gamma family, and a prob covariate
fit_cfr(d, delay = LogNormal(meanlog = 2.41, sdlog = 0.51))
fit_cfr(d, delay = Gamma(shape = Normal(3.3, 1), rate = Normal(0.26, 0.08)))
fit_cfr(d, delay = otd, formula = brms::bf(mu ~ 1, prob ~ group))

## End(Not run)
```

posterior_cumulative_incidence

Posterior cumulative incidence and hazards of death and recovery

Description

Turns a fit's outcome probability and delay distributions into the quantities a competing-risks analysis reports, for each covariate pattern in newdata. The cumulative incidence of death and of recovery by time t since onset are

$$I_D(t) = p F_D(t), \quad I_R(t) = (1 - p) F_R(t)$$

where p is prob and F_D and F_R are the distribution functions of the death and recovery delays, censored for the day of onset and truncated at each delay's max as in the likelihood.

Usage

```
posterior_cumulative_incidence(
  object,
  newdata = NULL,
  times = NULL,
  type = c("cumulative_incidence", "hazard", "subdistribution_hazard"),
  pwindow = 1
)
```

Arguments

object	A <code>cfrnow_fit</code> from <code>fit_cfr()</code> .
newdata	A data frame with the covariates of the prob and delay formulas, one row per covariate pattern. Defaults to the distinct covariate patterns among the fitted cases, in the order they first appear in <code>object\$data</code> (a single row for a fit without covariates).
times	Days since onset at which to evaluate. Defaults to every day from 0 to the longest follow-up in the fitted data.
type	"cumulative_incidence" (the default), "hazard" for the cause-specific hazard or "subdistribution_hazard".
pwindow	Width of the onset window, in days, that the delays are censored over. Defaults to a single day.

Details

The hazards are daily, matching the daily resolution of the data. The cause-specific hazard of death on day t is the probability of dying during $[t, t + 1)$ among cases still unresolved at t ,

$$h_D(t) = \frac{I_D(t+1) - I_D(t)}{1 - I_D(t) - I_R(t)},$$

and the subdistribution (Fine-Gray) hazard keeps cases that have recovered in the risk set,

$$\tilde{h}_D(t) = \frac{I_D(t+1) - I_D(t)}{1 - I_D(t)}.$$

Recovery works the same way. Hazard ratios between two covariate patterns are ratios of these draws. They vary with t , since the model does not assume proportional hazards; covariates act on prob and on the delays separately.

A fit without a `recovery_delay` returns death only. Its survivors never resolve in the model, so the cause-specific and subdistribution hazards coincide. Loss to follow-up only changes which outcomes get recorded and plays no part here.

Value

A data frame with one row per posterior draw, row of `newdata`, time and outcome: `.draw`, `row` (the row of `newdata`), `time`, `outcome` ("death" or "recovery") and `value`. A hazard is NA at a time when every case has already resolved.

See Also

Other fit: `fit_cfr()`, `posterior_prob_death()`, `pp_check_cfr()`, `summary.cfrnow_fit()`

Examples

```
## Not run:
ll <- simulate_linelist(n = 500, cfr = 0.4, delay = LogNormal(2.4, 0.5))
ll$group <- sample(c("a", "b"), nrow(ll), replace = TRUE)
d <- prepare_cfr_data(ll,
```

```

    obs_time = max(l1$onset_date) - 5,
    covariates = "group"
  )
  fit <- fit_cfr(d,
    delay = LogNormal(Normal(2.4, 0.2), Normal(0.5, 0.15)),
    formula = brms::bf(mu ~ group, prob ~ group)
  )
  groups <- data.frame(group = c("a", "b"))

  # cumulative incidence of death by day 7, 14 and 28 in each group
  ci <- posterior_cumulative_incidence(fit, groups, times = c(7, 14, 28))
  aggregate(value ~ row + time, ci, stats::median)

  # cause-specific hazard ratio of death, group b vs a, over time
  h <- posterior_cumulative_incidence(fit, groups,
    times = 0:28,
    type = "hazard"
  )
  a <- h[h$outcome == "death" & h$row == 1, ]
  b <- h[h$outcome == "death" & h$row == 2, ]
  a$hr <- b$value / a$value
  aggregate(hr ~ time, a, stats::median)

  ## End(Not run)

```

posterior_prob_death *Posterior probability of death for each case*

Description

`brms::posterior_linpred()` gives each case's prob, the probability of death before anything about the case's outcome has been observed. For a case still unresolved at the cut-off after y days of follow-up, this updates prob with that follow-up by Bayes' rule (the censored branch of the cure likelihood):

$$\pi = \frac{p S_D(y)}{p S_D(y) + (1 - p) S_R(y)}$$

where S_D and S_R are the survivor functions of the death and recovery delays, censored for the day of onset and truncated at each delay's max, and $S_R = 1$ when the fit has no recovery_delay. A fit with a `loss_prior` replaces each survivor function S with $1 + (1 - l) S$, where l is the probability that a case with that outcome is lost to follow-up. A case already resolved by the cut-off has a deterministic π : 1 for an observed death, 0 for a recovery or a resolved non-death.

Usage

```
posterior_prob_death(object)
```

Arguments

`object` A `cfrnow_fit` from `fit_cfr()`.

Details

A case with little follow-up (small y) has $S_D(y)$ close to 1, so its π_i sits close to its (covariate-specific) prob; only once it has been followed for close to the typical delay does π_i move away from prob towards 0 or 1. Averaging π_i over cases grouped by onset date therefore gives the expected fatal fraction for each onset period without needing a time trend in prob. For intervals on the fraction that actually dies, draw each case's outcome from its π_i before averaging. Estimates for the most recent periods, whose cases have had little follow-up, lean towards the fitted prob for those cases' covariates rather than their eventual outcome.

Works for any of the supported delay families and any prob or delay formula, since the per-case parameters come from `posterior_linpred()`.

Value

A `draws_df` with one π_i column per case, in the row order of `object$data`, holding posterior draws of the probability of death.

See Also

Other fit: `fit_cfr()`, `posterior_cumulative_incidence()`, `pp_check_cfr()`, `summary.cfrnow_fit()`

Examples

```
## Not run:
ll <- simulate_linelist(n = 500, cfr = 0.4, delay = LogNormal(2.4, 0.5))
d <- prepare_cfr_data(ll, obs_time = max(ll$onset_date) - 5)
fit <- fit_cfr(d, delay = LogNormal(Normal(2.4, 0.2), Normal(0.5, 0.15)))
pi <- posterior_prob_death(fit)

# real-time CFR by onset week: draw each case's outcome from its pi, then
# take the fatal fraction of the cases in each week
pi <- posterior::as_draws_matrix(pi)
fatal <- matrix(rbinom(length(pi), 1, pi), nrow(pi))
week <- cut(fit$cfrnow$onset, "week")
by_week <- vapply(split(seq_along(week), week), function(i) {
  rowMeans(fatal[, i, drop = FALSE])
}, numeric(nrow(fatal)))
apply(by_week, 2, stats::quantile, probs = c(0.025, 0.5, 0.975))

## End(Not run)
```

pp_check_cfr

Posterior-predictive check for a cfrnow fit

Description

Draws replicate line-list outcomes from the posterior and compares them with the data the model was fit to, replaying the real-time observation process (the onset-to-event timing and the cut-off censoring). Two checks are available: the count of observed deaths by the cut-off (plus recoveries for a two-outcome fit), and the distribution of the observed onset-to-death delays.

Usage

```
pp_check_cfr(object, type = c("counts", "delay"), ndraws = 100)
```

Arguments

object	A cfrnow_fit from <code>fit_cfr()</code> .
type	Which check to plot: "counts" (default) or "delay".
ndraws	Number of posterior draws to replicate over. Defaults to 100, or fewer if the fit has fewer draws.

Details

The check reuses the fit's own posterior draws of the CFR and the delay, so it works for covariate and time-varying prob ~ ... fits as well as intercept-only ones. It needs the observation cut-off, which `fit_cfr()` records when the data come from `prepare_cfr_data()`; each case's follow-up horizon is built from its own cut-off, so a per-case obs_time (e.g. one cut-off per reporting site) is replayed correctly. A retrospective fit (obs_time = NULL) has no truncation to replay, so every fatal case shows up as a death.

Only the quantities the model generates are checked: the observed death (and recovery) counts and the observed onset-to-death delays. The split of the remaining cases into censored versus untimed-resolved is not part of the generative model, so it is left out.

The check is stochastic: it subsamples posterior draws and simulates outcomes, so call `set.seed()` beforehand for reproducible plots.

Value

A ggplot object.

See Also

Other fit: `fit_cfr()`, `posterior_cumulative_incidence()`, `posterior_prob_death()`, `summary.cfrnow_fit()`

Examples

```
## Not run:
ll <- simulate_linelist(n = 500, cfr = 0.4, delay = LogNormal(2.4, 0.5))
d <- prepare_cfr_data(ll, obs_time = max(ll$onset_date) - 5)
fit <- fit_cfr(d, delay = LogNormal(Normal(2.4, 0.2), Normal(0.5, 0.15)))
pp_check_cfr(fit, type = "counts")
pp_check_cfr(fit, type = "delay")

## End(Not run)
```

prepare_cfr_data

Build model inputs from a line list

Description

Classifies each case at an observation cut-off into an observed death (with an interval-censored onset-to-death delay), a resolved non-death, or a right-censored survivor still unresolved at the cut-off, and returns the pieces `fit_cfr()` passes to Stan. A case counts as resolved non-death if it has a `recovery_date` on or before the cut-off (or, in a retrospective fit, if it simply never died); such a case contributes the cure term rather than being censored, so recording recoveries tightens the estimate.

Usage

```
prepare_cfr_data(
  linelist,
  obs_time = NULL,
  covariates = character(),
  t0 = NULL,
  max_delay = 60,
  last_contact_date = NULL
)
```

Arguments

<code>linelist</code>	A data frame with an <code>onset_date</code> column, an optional <code>onset_lower/onset_upper</code> onset window, a <code>death_date</code> column (NA for cases that have not died; use the date the death was notified, i.e. when it entered the data, so real-time censoring absorbs any reporting lag), and an optional <code>recovery_date</code> column (NA unless the case is a recorded non-fatal recovery). Dates may be Date or coercible.
<code>obs_time</code>	Real-time cut-off, or NULL for a retrospective fit in which every recorded death counts and survivors are treated as fully resolved. A single Date (or coercible) is shared by every case; a vector with one entry per row of <code>linelist</code> gives each case its own cut-off; or a string naming a <code>linelist</code> column holding those per-case cut-offs. In real time, a case with a recovery on or before its own <code>obs_time</code> is resolved; one still alive and unresolved is right-censored; and a death dated after its own <code>obs_time</code> is treated as not-yet-known (right-censored).
<code>covariates</code>	Character vector of <code>linelist</code> column names to carry through to the per-case model rows, so they can be used in a <code>prob ~ ...</code> formula. The onset date is always carried as <code>onset</code> ; for a time-varying CFR derive a time term from it (e.g. <code>week</code>) and pass <code>prob ~ s(week)</code> .
<code>t0</code>	Optional time origin (Date). Defaults to <code>min(onset) - max_delay</code> .
<code>max_delay</code>	Plausibility filter for data-entry errors, in days: a death record implying a negative onset-to-death delay, or one longer than <code>max_delay</code> , is dropped as a likely mis-keyed date. This only screens records; it does not bound or truncate the

onset-to-death delay the model fits, so set it comfortably above the longest credible delay to avoid discarding genuine long-delay deaths (which would bias the delay short). It also sets the default origin, $t_0 = \min(\text{onset}) - \text{max_delay}$.

last_contact_date

Optional column name in linelist holding the last date a case was known to be alive and unresolved (a transfer, a discharge against advice, the last ward note). A case with no recorded outcome is censored there instead of at obs_time, so the follow-up the model sees stops where the data do. NA means followed to the cut-off, and the column is ignored for a case whose death or recovery was recorded, which was followed until that happened.

Details

Onset is taken over the day-window [onset_lower, onset_upper] when those columns are present, defaulting to a one-day window at onset_date for a case whose window is missing (NA), or for every case when the columns are absent. In real time, a window that closes after the case's cut-off is cut back to the cut-off, since the case was already known then. Deaths and recoveries are recorded to the day.

Records that cannot be used are dropped with a warning: a missing onset, an inverted onset window (onset_upper < onset_lower), a death with an impossible onset-to-death delay (negative, or longer than max_delay), or a recovery dated before onset. In real time, a case whose onset falls after its own obs_time is not yet known and is excluded with a message.

Data often reaches the analyst at different times by site, so obs_time may give each case its own cut-off rather than one shared by the whole line list: a case's deaths, recoveries, follow-up and exclusion are all judged against its own cut-off, not the latest one in the data.

Value

A cfrnow_data list with the aggregated model inputs (n_death, death_delay, death_width, n_recovery, recovery_delay, recovery_width, n_cens, censor_time, censor_width, n_resolved, n_cases, n_deaths, n_recoveries, t0, obs_time, one per kept case) and a cases data frame with one row per kept case (y, outcome, pwindow, swindow, onset, obs_time, follow_up (days watched, Inf in a retrospective fit) and any requested covariates), which [as_epidist_cure_model\(\)](#) turns into the model frame.

Examples

```
ll <- simulate_linelist(n = 50, delay = LogNormal(2.4, 0.5))
prepare_cfr_data(ll, obs_time = as.Date("2026-02-01"))
```

Description

Draws onset dates over a window, marks each case fatal with probability `cfr`, and gives fatal cases an onset-to-death delay drawn from `delay`. Pass a recovery delay to also simulate onset-to-recovery times for the non-fatal cases and add a `recovery_date` column. Delays are `distspec` distributions with fixed parameters, matching what `fit_cfr()` takes; you can specify them by mean and sd (e.g. `LogNormal(mean = 12.75, sd = 7)`). Returns a line list with `onset_date`, `death_date` (NA for non-fatal cases) and, when recovery is given, `recovery_date` (NA for fatal cases). The full, untruncated outcomes are simulated; pass the result to `prepare_cfr_data()` with an `obs_time` to induce the real-time truncation.

Usage

```
simulate_linelist(
  n = 200,
  cfr = 0.5,
  delay,
  recovery = NULL,
  onset_start = as.Date("2026-01-01"),
  onset_days = 60
)
```

Arguments

<code>n</code>	Number of cases.
<code>cfr</code>	True case fatality ratio.
<code>delay</code>	Onset-to-death delay: a <code>distspec</code> distribution (<code>distspec::LogNormal()</code> , <code>distspec::Gamma()</code> or <code>distspec::Weibull()</code>) with fixed parameters.
<code>recovery</code>	Optional onset-to-recovery delay (same form as <code>delay</code>); when given, non-fatal cases get a <code>recovery_date</code> .
<code>onset_start</code>	First possible onset date.
<code>onset_days</code>	Width of the onset window (days); onsets are uniform over it.

Details

Data are generated to match the model's daily interval-censoring: each case's true onset falls uniformly within its recorded day, and the event day is the floor of the continuous onset-plus-delay time. So the recorded day-level delays are exactly a doubly-interval-censored draw, which makes the simulator suitable for checking calibration, not only rough recovery.

Value

A data frame with `onset_date`, `death_date` and, if recovery is given, `recovery_date`.

Examples

```
simulate_linelist(
  n = 5, cfr = 0.6,
  delay = LogNormal(mean = 12.75, sd = 7)
)
```

summary.cfrnow_fit *Summarise a mixture-cure CFR fit*

Description

Reports the corrected CFR and the onset-to-death delay (mean and sd, in days) as posterior quantiles with convergence diagnostics (rhat, ess_bulk). A fit with a loss_prior also reports the estimated probability of being lost to follow-up, as loss, or as loss_death and loss_recovery when the two outcomes have their own. The naive deaths / cases ratio is returned as an attribute; in real time it underestimates the corrected CFR because not every fatal case has died by the cut-off.

Usage

```
## S3 method for class 'cfrnow_fit'
summary(
  object,
  probs = c(0.025, 0.5, 0.975),
  info_tol = 0.9,
  ascertainment_ratio = 1,
  ...
)

## S3 method for class 'cfrnow_fit'
print(x, ...)
```

Arguments

object	A cfrnow_fit from <code>fit_cfr()</code> .
probs	Quantiles to report.
info_tol	Low-information threshold: flag when the CFR posterior sd is more than this fraction of the prior sd. Defaults to 0.9.
ascertainment_ratio	Ratio r of the ascertainment probability of fatal to non-fatal cases (see Details). A single positive number; defaults to 1 (no correction).
...	Unused.
x	A cfrnow_fit.

Details

When few deaths have resolved (a young outbreak) the CFR is only weakly identified and its posterior stays close to the prior. This is reported via the `cfr_low_information` attribute: TRUE when the CFR posterior sd exceeds `info_tol` times the prior sd.

For a `prob ~ group` fit the CFR varies by group, so one `prob[<group>]` row is reported per group (grouping predictors must be factors or characters), and the `cfr_low_information` flag is NA (only defined for a single CFR).

The CFR the model fits is the fatality risk among *ascertained* cases. When ascertainment is outcome-dependent – fatal and non-fatal cases entering the line list at different rates – this differs from the population CFR. `ascertainment_ratio` (r) is the ratio of the ascertainment probability of fatal to non-fatal cases; the reported CFR is shifted on the logit scale by $-\log(r)$, so $r > 1$ (fatal cases over-ascertained) lowers it and $r < 1$ (e.g. deaths not linked back to cases) raises it. It is supplied, not fitted, and defaults to 1 (no correction); because the correction is a post-hoc logit shift, sweep a range of r to show its leverage rather than trusting a single value.

Value

A data frame with one row per quantity: `prob` (or one `prob[<group>]` row per group for a `prob ~ group` fit), `delay_mean` and `delay_sd`, `recovery_mean` and `recovery_sd` for a two-outcome fit, and `loss` (or `loss_death` and `loss_recovery`) for a fit with an estimated `loss_prior`. It carries `naive_cfr`, `n_cases`, `n_deaths`, `cfr_prior_sd`, `cfr_low_information` and `ascertainment_ratio` attributes.

See Also

Other fit: [fit_cfr\(\)](#), [posterior_cumulative_incidence\(\)](#), [posterior_prob_death\(\)](#), [pp_check_cfr\(\)](#)

Index

- * **cure_model**
 - as_epidist_cure_model, [2](#)
- * **fit**
 - fit_cfr, [3](#)
 - posterior_cumulative_incidence, [5](#)
 - posterior_prob_death, [7](#)
 - pp_check_cfr, [8](#)
 - summary.cfrnow_fit, [13](#)
- as_epidist_cure_model, [2](#)
- as_epidist_cure_model(), [11](#)
- brms::brm(), [4](#)
- brms::posterior_linpred(), [7](#)
- distspec::Beta(), [3](#)
- distspec::Gamma(), [3](#), [12](#)
- distspec::LogNormal(), [3](#), [12](#)
- distspec::Weibull(), [3](#), [12](#)
- epidist::epidist(), [4](#)
- fit_cfr, [3](#)
- fit_cfr(), [6–10](#), [12–14](#)
- posterior_cumulative_incidence, [5](#)
- posterior_cumulative_incidence(), [5](#), [8](#),
[9](#), [14](#)
- posterior_prob_death, [7](#)
- posterior_prob_death(), [5](#), [6](#), [9](#), [14](#)
- pp_check_cfr, [8](#)
- pp_check_cfr(), [5](#), [6](#), [8](#), [14](#)
- prepare_cfr_data, [10](#)
- prepare_cfr_data(), [2–4](#), [9](#), [12](#)
- print.cfrnow_fit(summary.cfrnow_fit),
[13](#)
- set.seed(), [9](#)
- simulate_linelist, [11](#)
- summary(), [4](#)
- summary.cfrnow_fit, [13](#)
- summary.cfrnow_fit(), [5](#), [6](#), [8](#), [9](#)