

# Package ‘coarseDataTools’

February 14, 2012

**Version** 0.5.1

**Date** 2011-04-20

**Title** A collection of functions to help with analysis of coarse infectious disease data

**Author** Nicholas G. Reich <nick.reich@gmail.com>, Justin Lessler <jlessler@jhsph.edu>

**Maintainer** Nicholas G. Reich <nick.reich@gmail.com>

**License** GPL (>= 2)

**Depends** survival

**Description** This package contains functions to analyze coarse data. Its development was motivated by applications to infectious disease: in particular, problems with estimating the incubation period and the case fatality ratio of a given disease. Sample data files are included in the package.

**Repository** CRAN

**Date/Publication** 2011-04-24 06:07:07

## R topics documented:

dic.fit . . . . .	2
dic.getSE . . . . .	3
EMforCFR . . . . .	4
exp.win.lengths . . . . .	6
fluA.inc.per . . . . .	7
fwFuncs . . . . .	8
get.obs.type . . . . .	9
LikelihoodFuncs . . . . .	9
loglik . . . . .	10
precision.simulation . . . . .	11
ProfileLiks . . . . .	13
run.Estep . . . . .	14

run.Mstep . . . . .	15
SEM.variance . . . . .	15
simulated.outbreak.deaths . . . . .	17

<b>Index</b>	<b>18</b>
--------------	-----------

---

dic.fit	<i>Fitting log-normal models to doubly interval-censored data</i>
---------	---

---

## Description

This function fits a log-normal model to doubly interval-censored survival data. It was developed with the application to incubation periods in mind. The data can be a mixture of doubly interval-censored, single interval-censored or exact observations from a single univariate distribution.

## Usage

```
dic.fit(dat, start.log.sigma = log(log(2)), opt.method = "L-BFGS-B", mu.int = c(log(0.5), log(13)), log
```

## Arguments

dat	a matrix with columns named "EL", "ER", "SL", "SR", corresponding to the left (L) and right (R) endpoints of the windows of possible exposure (E) and symptom onset (S). Also, a "type" column must be specified and have entries with 0, 1, or 2, corresponding to doubly interval-censored, single interval-censored or exact observations, respectively.
start.log.sigma	the log-log-scale starting value for the dispersion, used in a profile likelihood maximization which determines the starting values for the full likelihood maximization routine.
opt.method	the optimization method to use in the optim() function
mu.int	the log-scale interval of possible median values (in the same units as the observations in dat). Narrowing this interval can help speed up convergence of the algorithm, but care must be taken so that possible values are not excluded or that the maximization does not return a value at an endpoint of this interval.
log.sigma.int	the log-log-scale interval of possible dispersion values. See notes for mu.int above.
ptiles	percentiles of the distribution to be estimated in this routine. Note that the median and dispersion are estimated by default. See the ests matrix in the returned list below for more information.
...	other parameters to be passed to optim() which runs the full likelihood maximization.

## Details

See instructions in the arguments above.

**Value**

A list is returned with elements as follows:

ests	a matrix summarizing the results of fitting the model. Rows correspond to the median, dispersion and then percentiles specified by the ptils argument. Columns correspond to the point estimate, the lower and upper bounds on the 95% confidence interval and the standard error of the point estimate. If the maximization does not converge, this matrix is filled with NAs.
conv	A value of 1 indicates successful convergence; 0 indicates unsuccessful convergence.
MSG	The error message returned from optim() if the routine fails to converge.
Sig.log.scale	The inverse of the hessian matrix for the likelihood surface at the MLE. Used to determine the standard errors for the percentiles.

**Author(s)**

Nicholas G. Reich

**References**

Reich NG et al. *Statistics in Medicine*. Estimating incubation periods with coarse data. 2009. <http://www3.interscience.wiley.com/journal/122507367/abstract>

Lessler J et al. *Lancet Infectious Diseases*. Incubation periods of acute respiratory viral infections: a systematic review. <http://www.thelancet.com/journals/laninf/article/PIIS1473309909700696/abstract>

**Examples**

```
data(fluA.inc.per)
dic.fit(fluA.inc.per)
```

---

dic.getSE	<i>Calculates the standard errors for estimates from dic.fit() using the delta method.</i>
-----------	--

---

**Description**

This function is used at the end of the `dic.fit` routine to calculate the standard errors of point estimates for the desired characteristics of the log-normal distribution. It is not meant to be run by end users.

**Usage**

```
dic.getSE(mu, log.s, Sig, ptils)
```

**Arguments**

<code>mu</code>	the log of the median of the assumed underlying distribution, i.e. the mean of the underlying normal distribution.
<code>log.s</code>	the log-log of the dispersion of the assumed underlying distribution, i.e. the log of the standard deviation of the underlying normal distribution.
<code>Sig</code>	The inverse of the hessian matrix for the likelihood surface at the MLE.
<code>ptiles</code>	percentiles of the distribution to be estimated in this routine. Note that the median and dispersion are estimated by default. See the ests matrix in the returned list below for more information.

**Value**

A vector of standard errors for (in this order) the 50th percentile, the dispersion, and any percentile specified in `ptiles`.

**Author(s)**

Nicholas G. Reich

**See Also**

[dic.fit](#)

---

EMforCFR	<i>A function to estimate the relative case fatality ratio when reporting rates are time-varying and deaths are lagged because of survival time.</i>
----------	--

---

**Description**

This function implements an EM algorithm to estimate the relative case fatality ratio between two groups when reporting rates are time-varying and deaths are lagged because of survival time.

**Usage**

```
EMforCFR(assumed.nu, alpha.start.values, full.data, max.iter = 50,
         verb = FALSE, tol = 1e-10, SEM.var = TRUE)
```

**Arguments**

<code>assumed.nu</code>	a vector of probabilities corresponding to the survival distribution, i.e. $\text{nu}[i] = \Pr(\text{surviving } i \text{ days} \mid \text{fatal case})$
<code>alpha.start.values</code>	a vector starting values for the reporting rate parameter of the GLM model. This must have length which corresponds to one less than the number of unique integer values of <code>full.dat[, "new.times"]</code> .
<code>full.data</code>	A matrix of observed data. See description below.

<code>max.iter</code>	The maximum number of iterations for the EM algorithm and the accompanying SEM algorithm (if used).
<code>verb</code>	An indicator for whether the function should print results as it runs.
<code>tol</code>	A tolerance to use to test for convergence of the EM algorithm.
<code>SEM.var</code>	If TRUE, the SEM algorithm will be run in addition to the EM algorithm to calculate the variance of the parameter estimates.

### Details

The data matrix `full.data` must have the following columns:

"grp": a 1 or a 2 indicating which of the two groups,  $j$ , the observation is for.

"new.times": an integer value representing the time,  $t$ , of observation.

"R": the count of recovered cases with onset at time  $t$  in group  $j$ .

"D": the count of deaths which occurred at time  $t$  in group  $j$  (note that these deaths did not have disease onset at time  $t$  but rather died at time  $t$ ).

"N": the total cases at  $t, j$ , or the sum of R and D columns.

### Value

A list with the following elements

<code>naive.rel.cfr</code>	the naive estimate of the relative case fatality ratio
<code>glm.rel.cfr</code>	the reporting-rate-adjusted estimate of the relative case fatality ratio
<code>EM.rel.cfr</code>	the lag-adjusted estimate of the relative case fatality ratio
<code>EM.re.cfr.var</code>	the variance for the log-scale lag-adjusted estimator taken from the final M-step
<code>EM.rel.cfr.var.SEM</code>	the Supplemented EM algorithm variance for the log-scale lag-adjusted estimator
<code>EM.rel.cfr.chain</code>	a vector of the EM algorithm iterates of the lag-adjusted relative CFR estimates
<code>EMiter</code>	the number of iterations needed for the EM algorithm to converge
<code>EMconv</code>	indicator for convergence of the EM algorithm. 0 indicates all parameters converged within <code>max.iter</code> iterations. 1 indicates that the estimate of the relative case fatality ratio converged but other did not. 2 indicates that the relative case fatality ratio did not converge.
<code>SEMconv</code>	indicator for convergence of SEM algorithm. Same scheme as <code>EMconv</code> .
<code>ests</code>	the coefficient estimates for the model
<code>ests.chain</code>	a matrix with all of the coefficient estimates, at each EM iteration
<code>DM</code>	the DM matrix from the SEM algorithm
<code>DMiter</code>	a vector showing how many iterations it took for the variance component to converge in the SEM algorithm

**Author(s)**

Nicholas G. Reich

**Examples**

```
## This is code from the CFR vignette provided in the documentation.

data(simulated.outbreak.deaths)
min.cases <- 10
N.1 <- simulated.outbreak.deaths[1:60, "N"]
N.2 <- simulated.outbreak.deaths[61:120, "N"]
first.t <- min(which(N.1 > min.cases & N.2 > min.cases))
last.t <- max(which(N.1 > min.cases & N.2 > min.cases))
idx.for.Estep <- first.t:last.t
new.times <- 1:length(idx.for.Estep)
simulated.outbreak.deaths <- cbind(simulated.outbreak.deaths, new.times = NA)
simulated.outbreak.deaths[c(idx.for.Estep, idx.for.Estep + 60), "new.times"] <- rep(new.times, + 2)
assumed.nu = c(0, 0.3, 0.4, 0.3)
alpha.start <- rep(0, 22)

## caution! this next line may take several minutes (5-10, depending on
##   the speed of your machine) to run.
## Not run: cfr.est = EMforCFR(assumed.nu = assumed.nu, alpha.start.values = alpha.start,
## full.data = simulated.outbreak.deaths, verb = FALSE, SEM.var = TRUE, max.iter = 500, tol = 1e-05)
## End(Not run)
```

---

exp.win.lengths

*Exposure window lengths from an influenza outbreak at a NYC school*


---

**Description**

A numeric vector of exposure window lengths taken from a dataset of doubly interval-censored incubation period observations. All observations came from a NYC public school. The outbreak has been described in full in Lessler et al. (see citation below).

**Usage**

```
data(exp.win.lengths)
```

**Format**

A numeric vector with 134 positive values. Each value represents an exposure window length from an observation of the incubation period for that individual. The exposure window length is the length of time during which exposure could have occurred. For example, if an individual could have been exposed anytime between 6am on Monday to 6am on Wednesday, her exposure window length would be 2 days.

**Source**

Lessler J et al. New England Journal of Medicine. Outbreak of 2009 Pandemic Influenza A (H1N1) at a New York City School. 2009. 361(27):2628-2636. <http://content.nejm.org/cgi/content/full/361/27/2628>

**Examples**

```
data(exp.win.lengths)
summary(exp.win.lengths)
hist(exp.win.lengths)
```

---

fluA.inc.per

*Coarse incubation period data for influenza A*

---

**Description**

These observations on the incubation period of influenza A come from a variety of sources, and were gathered for a literature review. They report doubly interval-censored, single interval-censored or exact observations for the incubation period.

**Usage**

```
data(fluA.inc.per)
```

**Format**

A data frame with 151 observations on the following 7 variables.

author the name of the primary author for the source of the observation

year the year of the study which is the source of the observation

EL the earliest possible time of infection

ER the latest possible time of infection

SL the earliest possible time of symptom onset

SR the latest possible time of symptom onset

type an indicator of the type of observation: 0 for doubly interval-censored, 1 for single-interval censored, 2 for exact

**Source**

Lessler J, Reich NG, Brookmeyer R, Perl TM, Nelson KE, Cummings DAT. (2009) A systematic review of the incubation periods of acute respiratory viral infections. Lancet Infectious Diseases. 9(5):291-300.

**Examples**

```
data(fluA.inc.per)
head(fluA.inc.per)
```

---

fwFuncs	<i>Functions to calculate the product of the density and the likelihood weight functions.</i>
---------	---

---

### Description

The fw1() and fw3() functions are not meant to be called by the end user but are used by `dic.fit` when calculating the likelihood of a given observation. For a given time, `t`, they calculate the likelihood, given a set of parameters (`mu` and `sigma`) and the observed endpoints of exposure and symptom onset windows. Details on the likelihood functions can be found in Reich et al. (2009) – see reference below.

### Usage

```
fw1(t, EL, ER, SL, SR, mu, sigma)
fw3(t, EL, ER, SL, SR, mu, sigma)
```

### Arguments

<code>t</code>	a real number, representing time.
<code>EL</code>	the left endpoint of the exposure window.
<code>ER</code>	the right endpoint of the exposure window.
<code>SL</code>	the left endpoint of the symptom onset window.
<code>SR</code>	the right endpoint of the symptom onset window.
<code>mu</code>	the log of the median of the assumed underlying distribution, i.e. the mean of the underlying normal distribution.
<code>sigma</code>	the log-log of the dispersion of the assumed underlying distribution, i.e. the log of the standard deviation of the underlying normal distribution.

### Value

fw1() returns the value for the first third of the integral for the doubly interval-censored likelihood calculation. fw3() returns the value for the last third of the integral for the doubly interval-censored likelihood calculation.

### Author(s)

Nicholas G. Reich

### References

Reich NG et al. Statistics in Medicine. Estimating incubation periods with coarse data. 2009. <http://www3.interscience.wiley.com/journal/122507367/abstract>

---

get.obs.type	<i>Generation of data types</i>
--------------	---------------------------------

---

**Description**

This function makes an educated guess at what type of data (doubly interval-censored, single interval-censored or exact) each observation in a doubly interval-censored dataset is.

**Usage**

```
get.obs.type(dat)
```

**Arguments**

dat	a matrix with columns named "EL", "ER", "SL", "SR", corresponding to the left (L) and right (R) endpoints of the windows of possible exposure (E) and symptom onset (S).
-----	--

**Value**

type	A numeric vector with entries corresponding to each row of 'dat'. The values 0, 1, or 2, correspond to doubly interval-censored, single interval-censored or exact observations, respectively.
------	--

**Author(s)**

Justin Lessler

**Examples**

```
data(fluA.inc.per)
get.obs.type(fluA.inc.per)
```

---

LikelihoodFuncs	<i>Functions to calculate the likelihood for a given, possibly coarse, survival-time observation.</i>
-----------------	---

---

**Description**

The functions documented here calculate the likelihood for a given survival-time observation. Observations may be doubly interval-censored, single interval-censored or exact. Several helper functions are included. None of these functions are meant to be run by end users, as they are all used by the function [dic.fit](#).

**Usage**

```
lik(mu, sigma, EL, ER, SL, SR, type)
diclik(mu, sigma, EL, ER, SL, SR)
diclik2(mu, sigma, EL, ER, SL, SR)
diclik2.helper1(x, SL, SR, mu, sigma)
diclik2.helper2(x, SR, mu, sigma)
siclik(mu, sigma, EL, ER, SL, SR)
exactlik(mu, sigma, EL, ER, SL, SR)
```

**Arguments**

mu	the log of the median of the assumed underlying distribution, i.e. the mean of the underlying normal distribution.
sigma	the log of the dispersion of the assumed underlying distribution, i.e. the standard deviation of the underlying normal distribution.
EL	the left endpoint of the exposure window.
ER	the right endpoint of the exposure window.
SL	the left endpoint of the symptom onset window.
SR	the right endpoint of the symptom onset window.
type	a value indicating one of three data types: 0 = doubly interval-censored; 1 = single interval-censored; 2 = exact.
x	a real number, used as an argument over which to integrate.

**Value**

Each of these functions returns a likelihood (or a portion of a likelihood calculation). Again, none of these are meant to be called by the end user.

**Author(s)**

Nicholas G. Reich

**See Also**

[dic.fit](#)

---

loglik	<i>General likelihood function used in the maximization performed in dic.fit().</i>
--------	---

---

**Description**

This function calculates the negative log-likelihood of a dataset of coarse observations and parameters for a log-normal distribution. This function is not meant to be called by the end user but is rather to be used by [dic.fit](#) when fitting a log-normal distribution to coarse data.

**Usage**

```
loglik(pars, dat)
```

**Arguments**

pars	A vector with parameters for a log-normal distribution. pars[1] is the mean of the underlying normal distribution assumed for the data. pars[2] is the log of the standard deviation of the underlying normal distribution.
dat	A dataset as described in the documentation for <a href="#">dic.fit</a> .

**Value**

The negative log likelihood of the entire dataset. The negative log-likelihood is returned so that the default settings of `optim()` (i.e. to minimize and not to maximize) can be used to find the maximum likelihood estimate.

**Author(s)**

Nicholas G Reich

**See Also**

[dic.fit](#)

---

precision.simulation *Simulate incubation period analyses with coarse data*

---

**Description**

These functions simulate coarse incubation period data sets and analyze them. The goal is for these simulations to provide evidence for how much information a given dataset contains about a characteristic of the incubation period distribution.

**Usage**

```
precision.simulation(N,  
  med = 2,  
  disp = 1.3,  
  percentile = 0.5,  
  nsim = 100,  
  exact.data = FALSE,  
  pct.type.A = 0.5,  
  exp.win.dat = NULL,  
  verb = FALSE)  
precision.simulation.coarse(N,  
  med,  
  disp,
```

```

    percentile,
    nsim,
    pct.type.A,
    exp.win.dat,
    verb)
precision.simulation.exact(N,
    med,
    disp,
    percentile,
    nsim,
    verb)
generate.coarse.data(N,
    med,
    disp,
    pct.type.A,
    exp.win.dat)

```

### Arguments

N	Overall sample size for the datasets to be simulated.
med	Median for the assumed log normal distribution of the incubation periods.
disp	Dispersion for the assumed log normal distribution of the incubation periods.
percentile	Percentile of the incubation period distribution which we want to estimate.
nsim	Number of datasets to analyze in the simulation.
exact.data	Either TRUE/FALSE. Indicates whether the data generated should be coarsened at all. If TRUE, pct.type.A and exp.win.dat are ignored.
pct.type.A	Percent of the N observations that are assumed to be type A data. If N*pct.type.A is not an integer, it will be rounded to the nearest integer.
exp.win.dat	A vector of exposure window lengths. Defaults to the observed window lengths from Lessler et al. (see below).
verb	If TRUE, a message with the system time and iteration number will be printed ten times during the simulation run.

### Value

The simulation.precision functions return a matrix with four columns and nsim rows. The "ests" column gives the estimated percentiles for the incubation period distribution. The "SE" column gives the standard error for the estimate. The "conv" column is 1 if the doubly interval-censored likelihood maximization converged. Otherwise, it is 0. The "bias" column gives the estimated percentile - true percentile.

The generate.coarse.data function returns a matrix with data suitable for analysis by the dic.fit() function.

### Author(s)

Nicholas G. Reich

## References

Lessler J et al. New England Journal of Medicine. Outbreak of 2009 Pandemic Influenza A (H1N1) at a New York City School. 2009. 361(27):2628-2636. <http://content.nejm.org/cgi/content/full/361/27/2628>

Reich NG et al. Statistics in Medicine. Estimating incubation periods with coarse data. 2009. <http://www3.interscience.wiley.com/journal/122507367/abstract>

## See Also

[dic.fit](#)

## Examples

```
a <- precision.simulation(N=100, nsim=10, exp.win.dat=c(1,2), verb=TRUE)
head(a)
summary(a)
```

---

ProfileLiks

*Profile likelihood helper functions used by dic.fit().*

---

## Description

Calculates the likelihood of a dataset given parameters of a log-normal distribution. These functions are not meant to be called directly by the end user but are used by the [dic.fit](#) function to determine starting values for the log-scale median and log-log-scale dispersion of the distribution.

## Usage

```
pl.mu(mu, log.sigma, dat)
pl.sigma(log.sigma, mu, dat)
```

## Arguments

mu	the log of the median of the log-normal distribution.
log.sigma	the log-log (i.e. the log taken twice) of the dispersion of the log-normal distribution.
dat	a dataset, as described in the help file of <a href="#">dic.fit</a> .

## Value

the negative log-likelihood for the dataset, given the parameters mu and log.sigma.

## Note

Two versions of the function are needed because the function optimize only can optimize on the first parameter specified by the function. So `pl.mu()` is used to find the value of mu that maximizes the likelihood and `pl.sigma()` is used to find the value of the dispersion (on the log-log scale) that maximizes the likelihood.

**Author(s)**

Nicholas G. Reich

**See Also**

[dic.fit](#)

---

run.Estep

*The expectation step for a specialized EM algorithm*

---

**Description**

The Expectation- or E-step for the EM algorithm implemented in EMforCFR(). Not meant to be called by users.

**Usage**

```
run.Estep(alpha, full.data, nlag, assumed.nu)
```

**Arguments**

alpha	Current estimates of the alpha parameters from the GLM model.
full.data	A matrix of observed data. See description in EMforCFR helpfile.
nlag	The number of time units for lagged data. Corresponds to length(assumed.nu).
assumed.nu	a vector of probabilities corresponding to the survival distribution, i.e. $\text{nu}[i]=\text{Pr}(\text{surviving } i \text{ days} \mid \text{fatal case})$

**Value**

A data matrix with the same format as full.data from the EMforCFR() documentation.

**Author(s)**

Nicholas G. Reich

**See Also**

[EMforCFR](#)

---

run.Mstep	<i>The maximization step for a specialized EM algorithm</i>
-----------	---

---

**Description**

The Maximization- or M-step for the EM algorithm implemented in EMforCFR(). Not meant to be called by users.

**Usage**

```
run.Mstep(dat)
```

**Arguments**

dat                    data matrix passed from EMforCFR().

**Value**

A list with two components

phi                    fitted vector of parameters

Var                    variance-covariance matrix from the fitted model

**Author(s)**

Nicholas G. Reich

**See Also**

[EMforCFR](#)

---

SEM.variance	<i>Implementation of the Supplemented EM algorithm</i>
--------------	--

---

**Description**

This function is meant to be run only through the function EMforCFR() and is used to calculate the variance via the Supplemented EM algorithm (see Meng and Rubin, 1991)

**Usage**

```
SEM.variance(full.data, dat, phi, max.iter, tol, nlag, alpha.start.values, assumed.nu)
```

**Arguments**

<code>full.data</code>	A matrix of observed data. See description in EMforCFR helpfile.
<code>dat</code>	A data frame.
<code>phi</code>	A vector of fitted parameters from the final EM iteration.
<code>max.iter</code>	The maximum number of iterations for SEM algorithm.
<code>tol</code>	A tolerance to use to test for convergence of the EM algorithm.
<code>nlag</code>	The number of time units for lagged data. Corresponds to <code>length(assumed.nu)</code> .
<code>alpha.start.values</code>	a vector starting values for the reporting rate parameter of the GLM model. This must have length which corresponds to one less than the number of unique integer values of <code>full.dat[, "new.times"]</code> .
<code>assumed.nu</code>	a vector of probabilities corresponding to the survival distribution, i.e. $\text{nu}[i] = \text{Pr}(\text{surviving } i \text{ days} \mid \text{fatal case})$

**Value**

A list with the following components

<code>DM</code>	The estimate of the variance-covariance matrix for the model parameters. Only converged rows are returned.
<code>DMiter</code>	A vector whose <i>i</i> th entry is the number of iterations needed for convergence of the <i>i</i> th row of the DM matrix.
<code>loop.idx</code>	If not NULL, the values correspond to the original indices of DM which have been omitted because of lack of convergence.

**Author(s)**

Nicholas G. Reich

**References**

Meng, X.L. and Rubin, D.B. JASA. 1991: 86 (416), 899-909.

**See Also**

[EMforCFR](#)

---

`simulated.outbreak.deaths`*Simulated case and death reports from a fictional outbreak*

---

**Description**

This dataset provides reported counts of cases and deaths occurring at different time points across a simulated outbreak. Details of the data simulation algorithm are provided in the manuscript "Estimating case fatality ratios from infectious disease surveillance data" (Reich et al., under review, available upon request).

**Usage**

```
data(simulated.outbreak.deaths)
```

**Format**

`time` time, t, after start of outbreak

`grp` an categorical variable indicating membership in one of two groups of a covariate, j

`R` number of recovered cases reported at the given t and j

`D` number of deaths reported at the given t and j

`N` total number of cases and deaths reported at t and j, or D+R

**References**

Reich NG, Lessler J, Brookmeyer R. Estimating case fatality ratios from infectious disease surveillance data. [currently under review but available from the authors upon request]

**Examples**

```
data(simulated.outbreak.deaths)
head(simulated.outbreak.deaths)
plot(simulated.outbreak.deaths[simulated.outbreak.deaths[, "grp"]==1, "D"], type="l")
```

# Index

- \*Topic **EM algorithm**
    - SEM.variance, 15
  - \*Topic **case fatality ratio**
    - EMforCFR, 4
  - \*Topic **coarse data**
    - EMforCFR, 4
  - \*Topic **datasets**
    - exp.win.lengths, 6
    - fluA.inc.per, 7
  - \*Topic **incomplete data**
    - EMforCFR, 4
  - \*Topic **infectious disease**
    - EMforCFR, 4
  - \*Topic **models**
    - dic.fit, 2
  - \*Topic **supplemented EM algorithm**
    - SEM.variance, 15
  - \*Topic **survival**
    - dic.fit, 2
    - get.obs.type, 9
- dic.fit, 2, 3, 4, 8–11, 13, 14
- dic.getSE, 3
- diclik (LikelihoodFuncs), 9
- diclik2 (LikelihoodFuncs), 9
- EMforCFR, 4, 14–16
- exactlik (LikelihoodFuncs), 9
- exp.win.lengths, 6
- fluA.inc.per, 7
- fw1 (fwFuncs), 8
- fw3 (fwFuncs), 8
- fwFuncs, 8
- generate.coarse.data  
(precision.simulation), 11
- get.obs.type, 9
- lik (LikelihoodFuncs), 9
- LikelihoodFuncs, 9
- loglik, 10
- pl.mu (ProfileLiks), 13
- pl.sigma (ProfileLiks), 13
- precision.simulation, 11
- ProfileLiks, 13
- run.Estep, 14
- run.Mstep, 15
- SEM.variance, 15
- siclik (LikelihoodFuncs), 9
- simulated.outbreak.deaths, 17