# Package 'CLVTools'

October 19, 2021

**Title** Tools for Customer Lifetime Value Estimation

**Version** 0.8.1

**Date** 2021-10-13

**Depends** R (>= 3.5.0), methods

**Description** A set of state-of-the-art probabilistic modeling approaches to derive estimates of individual customer lifetime values (CLV).
Commonly, probabilistic approaches focus on modelling 3 processes, i.e. individuals' attrition, transaction, and spending process.
Latent customer attrition models, which are also known as ``buy-'til-you-die models", model the attrition as well as the transaction process.
They are used to make inferences and predictions about transactional patterns of individual customers such as their future purchase behavior.
Moreover, these models have also been used to predict individuals' long-term engagement in activities such as playing an online game or
posting to a social media platform. The spending process is usually modelled by a separate probabilistic model. Combining these results yields in
lifetime values estimates for individual customers.
This package includes fast and accurate implementations of various probabilistic models for non-contractual settings
(e.g., grocery purchases or hotel visits). All implementations support time-invariant covariates, which can be used to control for e.g.,
socio-demographics. If such an extension has been proposed in literature, we further provide the possibility to control for time-varying
covariates to control for e.g., seasonal patterns.
Currently, the package includes the following latent attrition models to model individuals' attrition and transaction process:
[1] Pareto/NBD model (Pareto/Negative-Binomial-Distribution),
[2] the Extended Pareto/NBD model (Pareto/Negative-Binomial-Distribution with time-varying covariates),
[3] the BG/NBD model (Beta-Gamma/Negative-Binomial-Distribution) and the
[4] GGom/NBD (Gamma-Gompertz/Negative-Binomial-Distribution).
Further, we provide an implementation of the Gamma/Gamma model to model the spending process of individuals.

**Imports** data.table (>= 1.12.0), ggplot2 (>= 3.2.0), lubridate (>= 1.7.8), Matrix (>= 1.2-17), MASS, optimx (>= 2019-12.02),

Rcpp(>= 0.12.12), stats, utils

**Suggests** covr, future, knitr, rmarkdown, testthat

**License** GPL-3

**URL** https://github.com/bachmannpatrick/CLVTools

**BugReports** https://github.com/bachmannpatrick/CLVTools/issues

**NeedsCompilation** yes

**SystemRequirements** C++11

**LinkingTo** Rcpp, RcppArmadillo (>= 0.9.500.2.0), RcppGSL (>= 0.3.7)

**LazyLoad** yes

**Encoding** UTF-8

**Collate** 'CLVTools.R' 'RcppExports.R' 'all_generics.R'
'class_clv_time.R' 'class_clv_data.R' 'class_clv_model.R'
'class_clv_fitted.R' 'class_clv_fitted_transactions.R'
'class_clv_model_nocorrelation.R' 'class_clv_model_bgnbd.R'
'class_clv_bgnbd.R' 'class_clv_fitted_transactions_staticcov.R'
'class_clv_data_staticcovariates.R'
'class_clv_model_bgnbd_staticcov.R'
'class_clv_bgnbd_staticcov.R'
'class_clv_data_dynamiccovariates.R'
'class_clv_fitted_spending.R'
'class_clv_fitted_transactions_dynamiccov.R'
'class_clv_model_gg.R' 'class_clv_gg.R'
'class_clv_model_ggomnbd_nocov.R' 'class_clv_ggomnbd.R'
'class_clv_model_ggomnbd_staticcov.R'
'class_clv_ggomnbd_staticcov.R'
'class_clv_model_withcorrelation.R' 'class_clv_model_pnbd.R'
'class_clv_model_pnbd_staticcov.R'
'class_clv_model_pnbd_dynamiccov.R' 'class_clv_pnbd.R'
'class_clv_pnbd_dynamiccov.R' 'class_clv_pnbd_staticcov.R'
'class_clv_time_date.R' 'class_clv_time_datetime.R'
'class_clv_time_days.R' 'class_clv_time_hours.R'
'class_clv_time_weeks.R' 'class_clv_time_years.R'
'clv_template_controlflow_estimate.R'
'clv_template_controlflow_predict.R' 'data.R'
'f_DoExpectation.R' 'f_clvdata_inputchecks.R'
'f_clvfitted_inputchecks.R' 'f_generics_clvdata.R'
'f_generics_clvfitted.R' 'f_generics_clvfitted_estimate.R'
'f_generics_clvfittedspending.R'
'f_generics_clvfittedtransactions.R'
'f_generics_clvfittedtransactionsdyncov.R'
'f_generics_clvfittedtransactionsstaticcov.R'
'f_generics_clvfittedtransactionsstaticcov_estimate.R'
'f_generics_clvpnbddyncov.R' 'f_interface_bgbb.R'
'f_interface_bgnbd.R' 'f_interface_clvdata.R'
'f_interface_gg.R' 'f_interface_ggomnbd.R' 'f_interface_pnbd.R'

'f_interface_predict_clvfittedspending.R'
'f_interface_predict_clvfittedtransactions.R'
'f_interface_setdynamiccovariates.R'
'f_interface_setstaticcovariates.R' 'f_s3generics_clvdata.R'
'f_s3generics_clvdata_dynamiccov.R'
'f_s3generics_clvdata_plot.R'
'f_s3generics_clvdata_staticcov.R' 'f_s3generics_clvfitted.R'
'f_s3generics_clvfittedspending_plot.R'
'f_s3generics_clvfittedtransactions_plot.R'
'f_s3generics_clvfittedtransactions_staticcov.R'
'f_s3generics_clvtime.R' 'interlayer_callLL.R'
'interlayer_callnextinterlayer.R' 'interlayer_constraints.R'
'interlayer_correlation.R' 'interlayer_manager.R'
'interlayer_regularization.R' 'pnbd_dyncov_ABCD.R'
'pnbd_dyncov_BkSum.R' 'pnbd_dyncov_CET.R' 'pnbd_dyncov_DECT.R'
'pnbd_dyncov_LL.R' 'pnbd_dyncov_createwalks.R'
'pnbd_dyncov_expectation.R' 'pnbd_dyncov_makewalks.R'
'pnbd_dyncov_palive.R'

**RoxygenNote** 7.1.2

**VignetteBuilder** knitr

**Author** Patrick Bachmann [cre, aut],
Niels Kuebler [aut],
Markus Meierer [aut],
Jeffrey Naef [aut],
Elliot Oblander [aut],
Patrik Schilter [aut]

**Maintainer** Patrick Bachmann <patrick.bachmann@business.uzh.ch>

**Repository** CRAN

**Date/Publication** 2021-10-18 22:50:09 UTC

# R topics documented:

---

`CLVTools-package` *Customer Lifetime Value Tools*

---

## Description

CLVTools is a toolbox for various probabilistic customer attrition models for non-contractual settings. It provides a framework, which is capable of unifying different probabilistic customer attrition models. This package provides tools to estimate the number of future transactions of individual customers as well as the probability of customers being alive in future periods. Further, the average spending by customers can be estimated. Multiplying the future transactions conditional on being alive and the predicted individual spending per transaction results in an individual CLV value.

The implemented models require transactional data from non-contractual businesses (i.e. customers' purchase history).

## Author(s)

**Maintainer**: Patrick Bachmann `<patrick.bachmann@business.uzh.ch>`

Authors:

- Niels Kuebler `<niels.kuebler@uzh.ch>`
- Markus Meierer `<markus.meierer@business.uzh.ch>`

- Jeffrey Naef <naef@stat.math.ethz.ch>
- Elliot Oblander <eoblander23@gsb.columbia.edu>
- Patrik Schilter <patrik.schilter@gmail.com>


## See Also

Development for CLVTools can be followed via the GitHub repository at https://github.com/bachmannpatrick/CLVTools.

## Examples

```
data("cdnow")

# Create a CLV data object, split data in estimation and holdout sample
clv.data.cdnow <- clvdata(data.transactions = cdnow, date.format = "ymd",
                          time.unit = "week", estimation.split = 39, name.id = "Id")

# summary of data
summary(clv.data.cdnow)

# Fit a PNBD model without covariates on the first 39 periods
pnbd.cdnow <- pnbd(clv.data.cdnow,
                   start.params.model = c(r=0.5, alpha=8, s=0.5, beta=10))
# inspect fit
summary(pnbd.cdnow)

# Predict 10 periods (weeks) ahead from estimation end
#   and compare to actuals in this period
pred.out <- predict(pnbd.cdnow, prediction.end = 10)

# Plot the fitted model to the actual repeat transactions
plot(pnbd.cdnow)
```

---

# apparelDynCov *Time-varying Covariates for the Apparel Retailer Dataset*

---

## Description

This simulated data contains direct marketing information on all 250 customers in the "apparelTrans" dataset. This information can be used as time-varying covariates.

## Usage

```
data("apparelDynCov")
```

## Format

A data.table with 20500 rows and 5 variables

**Id** Customer Id

**Cov.Date** Date of contextual factor

**Marketing** Direct marketing variable: 1 if customer was contacted with direct marketing in this time period

**Gender** 0=male, 1=female

**Channel** Acquisition channel: 0=online, 1=offline

---

# apparelStaticCov *Time-invariant Covariates for the Apparel Retailer Dataset*

---

## Description

This simulated data contains additional demographic information on all 250 customers in the "apparelTrans" dataset. This information can be used as time-invariant covariates.

## Usage

```
data("apparelStaticCov")
```

## Format

A `data.table` with 250 rows and 3 variables:

**Id** Customer Id

**Gender** 0=male, 1=female

**Channel** Acquisition channel: 0=online, 1=offline

---

# apparelTrans *Apparel Retailer Dataset*

---

## Description

This is a simulated dataset containing the entire purchase history of customers made their first purchase at an apparel retailer on January 3rd 2005. In total the dataset contains 250 customers who made 3648 transactions between January 2005 and mid July 2006.

## Usage

```
data("apparelTrans")
```

### Format

A `data.table` with 2353 rows and 3 variables:

`Id` Customer Id

`Date` Date of purchase

`Price` Price of purchase

---

## bgbb *BG/BB models - Work In Progress*

---

### Description

Fits BG/BB models on transactional data with static and without covariates. Not yet implemented.

### Usage

```
## S4 method for signature 'clv.data'
bgbb(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
  verbose = TRUE,
  ...
)

## S4 method for signature 'clv.data.static.covariates'
bgbb(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
  verbose = TRUE,
  names.cov.life = c(),
  names.cov.trans = c(),
  start.params.life = c(),
  start.params.trans = c(),
  names.cov.constr = c(),
  start.params.constr = c(),
  reg.lambdas = c(),
  ...
)

## S4 method for signature 'clv.data.dynamic.covariates'
bgbb(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
```

```
  verbose = TRUE,
  names.cov.life = c(),
  names.cov.trans = c(),
  start.params.life = c(),
  start.params.trans = c(),
  names.cov.constr = c(),
  start.params.constr = c(),
  reg.lambdas = c(),
  ...
)
```

## Arguments

| | |
|---|---|
| clv.data | The data object on which the model is fitted. |
| start.params.model | Named start parameters containing the optimization start parameters for the model without covariates. |
| optimx.args | Additional arguments to control the optimization which are forwarded to optimx::optimx. If multiple optimization methods are specified, only the result of the last method is further processed. |
| verbose | Show details about the running of the function. |
| ... | Ignored |
| names.cov.life | Which of the set Lifetime covariates should be used. Missing parameter indicates all covariates shall be used. |
| names.cov.trans | Which of the set Transaction covariates should be used. Missing parameter indicates all covariates shall be used. |
| start.params.life | Named start parameters containing the optimization start parameters for all lifetime covariates. |
| start.params.trans | Named start parameters containing the optimization start parameters for all transaction covariates. |
| names.cov.constr | Which covariates should be forced to use the same parameters for the lifetime and transaction process. The covariates need to be present as both, lifetime and transaction covariates. |
| start.params.constr | Named start parameters containing the optimization start parameters for the constraint covariates. |
| reg.lambdas | Named lambda parameters used for the L2 regularization of the lifetime and the transaction covariate parameters. Lambdas have to be >= 0. |

## Value

No value is returned.

bgnbd *BG/NBD models*

## Description

Fits BG/NBD models on transactional data without and with static covariates.

## Usage

```
## S4 method for signature 'clv.data'
bgnbd(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
  verbose = TRUE,
  ...
)

## S4 method for signature 'clv.data.static.covariates'
bgnbd(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
  verbose = TRUE,
  names.cov.life = c(),
  names.cov.trans = c(),
  start.params.life = c(),
  start.params.trans = c(),
  names.cov.constr = c(),
  start.params.constr = c(),
  reg.lambdas = c(),
  ...
)
```

## Arguments

| | |
|---|---|
| `clv.data` | The data object on which the model is fitted. |
| `start.params.model` | Named start parameters containing the optimization start parameters for the model without covariates. |
| `optimx.args` | Additional arguments to control the optimization which are forwarded to `optimx::optimx`. If multiple optimization methods are specified, only the result of the last method is further processed. |
| `verbose` | Show details about the running of the function. |
| `...` | Ignored |

`names.cov.life` Which of the set Lifetime covariates should be used. Missing parameter indicates all covariates shall be used.

`names.cov.trans` Which of the set Transaction covariates should be used. Missing parameter indicates all covariates shall be used.

`start.params.life` Named start parameters containing the optimization start parameters for all lifetime covariates.

`start.params.trans` Named start parameters containing the optimization start parameters for all transaction covariates.

`names.cov.constr` Which covariates should be forced to use the same parameters for the lifetime and transaction process. The covariates need to be present as both, lifetime and transaction covariates.

`start.params.constr` Named start parameters containing the optimization start parameters for the constraint covariates.

`reg.lambdas` Named lambda parameters used for the L2 regularization of the lifetime and the transaction covariate parameters. Lambdas have to be >= 0.

## Details

Model parameters for the BG/NBD model are `r`, `alpha`, `a`, and `b`.
`r`: shape parameter of the Gamma distribution of the purchase process.
`alpha`: scale parameter of the Gamma distribution of the purchase process.
`a`: shape parameter of the Beta distribution of the dropout process.
`b`: shape parameter of the Beta distribution of the dropout process.

If no start parameters are given, r = 1, alpha = 3, a = 1, b = 3 is used. All model start parameters are required to be > 0. If no start values are given for the covariate parameters, 0.1 is used.

Note that the DERT expression has not been derived (yet) and it consequently is not possible to calculated values for DERT and CLV.

**The BG/NBD model:** The BG/NBD is an "easy" alternative to the Pareto/NBD model that is easier to implement. The BG/NBD model slight adapts the behavioral "story" associated with the Pareto/NBD model in order to simplify the implementation. The BG/NBD model uses a beta-geometric and exponential gamma mixture distributions to model customer behavior. The key difference to the Pareto/NBD model is that a customer can only churn right after a transaction. This simplifies computations significantly, however has the drawback that a customer cannot churn until he/she makes a transaction. The Pareto/NBD model assumes that a customer can churn at any time.

**BG/NBD model with static covariates:** The standard BG/NBD model captures heterogeneity was solely using Gamma distributions. However, often exogenous knowledge, such as for example customer demographics, is available. The supplementary knowledge may explain part of the heterogeneity among the customers and therefore increase the predictive accuracy of the model. In addition, we can rely on these parameter estimates for inference, i.e. identify and quantify effects of contextual factors on the two underlying purchase and attrition processes. For technical details we refer to the technical note by Fader and Hardie (2007).

The likelihood function is the likelihood function associated with the basic model where alpha, a, and b are replaced with alpha = alpha0*exp(-g1z1), a = a_0*exp(g2z2), and b = b0*exp(g3z2) while r remains unchanged. Note that in the current implementation, we constrain the covariate parameters and data for the lifetime process to be equal (g2=g3 and z2=z3).

## Value

Depending on the data object on which the model was fit, bgnbd returns either an object of class clv.bgnbd or clv.bgnbd.static.cov.

The function summary can be used to obtain and print a summary of the results. The generic accessor functions coefficients, vcov, fitted, logLik, AIC, BIC, and nobs are available.

## References

Fader PS, Hardie BGS, Lee, KL (2005). ""Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model" Marketing Science, 24(2), 275–284.

Fader PS, Hardie BGS (2013). "Overcoming the BG/NBD Model's #NUM! Error Problem" URL http://brucehardie.com/notes/027/bgnbd_num_error.pdf.

Fader PS, Hardie BGS (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

## See Also

clvdata to create a clv data object, SetStaticCovariates to add static covariates to an existing clv data object.

predict to predict expected transactions, probability of being alive, and customer lifetime value for every customer

plot to plot the unconditional expectation as predicted by the fitted model

The generic functions vcov, summary, fitted.

## Examples

```
data("apparelTrans")
clv.data.apparel <- clvdata(apparelTrans, date.format = "ymd",
                            time.unit = "w", estimation.split = 40)

# Fit standard bgnbd model
bgnbd(clv.data.apparel)

# Give initial guesses for the model parameters
bgnbd(clv.data.apparel,
     start.params.model = c(r=0.5, alpha=15, a = 2, b=5))


# pass additional parameters to the optimizer (optimx)
#    Use Nelder-Mead as optimization method and print
#    detailed information about the optimization process
```

```
apparel.bgnbd <- bgnbd(clv.data.apparel,
                    optimx.args = list(method="Nelder-Mead",
                                       control=list(trace=6)))

# estimated coefs
coef(apparel.bgnbd)

# summary of the fitted model
summary(apparel.bgnbd)

# predict CLV etc for holdout period
predict(apparel.bgnbd)

# predict CLV etc for the next 15 periods
predict(apparel.bgnbd, prediction.end = 15)


# To estimate the bgnbd model with static covariates,
#   add static covariates to the data
data("apparelStaticCov")
clv.data.static.cov <-
 SetStaticCovariates(clv.data.apparel,
                     data.cov.life = apparelStaticCov,
                     names.cov.life = c("Gender", "Channel"),
                     data.cov.trans = apparelStaticCov,
                     names.cov.trans = c("Gender", "Channel"))

# Fit bgnbd with static covariates
bgnbd(clv.data.static.cov)

# Give initial guesses for both covariate parameters
bgnbd(clv.data.static.cov, start.params.trans = c(Gender=0.75, Channel=0.7),
                   start.params.life  = c(Gender=0.5, Channel=0.5))

# Use regularization
bgnbd(clv.data.static.cov, reg.lambdas = c(trans = 5, life=5))

# Force the same coefficient to be used for both covariates
bgnbd(clv.data.static.cov, names.cov.constr = "Gender",
                   start.params.constr = c(Gender=0.5))

# Fit model only with the Channel covariate for life but
# keep all trans covariates as is
bgnbd(clv.data.static.cov, names.cov.life = c("Channel"))
```

## Description

Calculates the expected number of transactions in a given time period based on a customer's past transaction behavior and the BG/NBD model parameters.

- `bgnbd_nocov_CET` Conditional Expected Transactions without covariates
- `bgnbd_staticcov_CET` Conditional Expected Transactions with static covariates

## Usage

```
bgnbd_nocov_CET(r, alpha, a, b, dPeriods, vX, vT_x, vT_cal)

bgnbd_staticcov_CET(
  r,
  alpha,
  a,
  b,
  dPeriods,
  vX,
  vT_x,
  vT_cal,
  vCovParams_trans,
  vCovParams_life,
  mCov_trans,
  mCov_life
)
```

## Arguments

| | |
|---|---|
| `r` | shape parameter of the Gamma distribution of the purchase process |
| `alpha` | scale parameter of the Gamma distribution of the purchase process |
| `a` | shape parameter of the Beta distribution of the lifetime process |
| `b` | shape parameter of the Beta distribution of the lifetime process |
| `dPeriods` | number of periods to predict |
| `vX` | Frequency vector of length n counting the numbers of purchases. |
| `vT_x` | Recency vector of length n. |
| `vT_cal` | Vector of length n indicating the total number of periods of observation. |
| `vCovParams_trans` | Vector of estimated parameters for the transaction covariates. |
| `vCovParams_life` | Vector of estimated parameters for the lifetime covariates. |
| `mCov_trans` | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |
| `mCov_life` | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |

## Details

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_trans at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_life at the respective position.

## Value

Returns a vector containing the conditional expected transactions for the existing customers in the BG/NBD model.

## References

Fader PS, Hardie BGS, Lee, KL (2005). ""Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model" Marketing Science, 24(2), 275–284.

Fader PS, Hardie BGS (2013). "Overcoming the BG/NBD Model's #NUM! Error Problem" URL http://brucehardie.com/notes/027/bgnbd_num_error.pdf.

Fader PS, Hardie BGS (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

---

# bgnbd_expectation *BG/NBD: Unconditional Expectation*

---

## Description

Computes the expected number of repeat transactions in the interval (0, vT_i] for a randomly selected customer, where 0 is defined as the point when the customer came alive.

## Usage

```
bgnbd_nocov_expectation(r, alpha, a, b, vT_i)

bgnbd_staticcov_expectation(r, vAlpha_i, vA_i, vB_i, vT_i)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process |
| alpha | scale parameter of the Gamma distribution of the purchase process |
| a | shape parameter of the Beta distribution of the lifetime process |
| b | shape parameter of the Beta distribution of the lifetime process |
| vT_i | Number of periods since the customer came alive |
| vAlpha_i | Vector of individual parameters alpha |
| vA_i | Vector of individual parameters a |
| vB_i | Vector of individual parameters b |

## Value

Returns the expected transaction values according to the chosen model.

## References

Fader PS, Hardie BGS, Lee, KL (2005). ""Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model" Marketing Science, 24(2), 275–284.

Fader PS, Hardie BGS (2013). "Overcoming the BG/NBD Model's #NUM! Error Problem" URL http://brucehardie.com/notes/027/bgnbd_num_error.pdf.

Fader PS, Hardie BGS (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

---

# bgnbd_LL *BG/NBD: Log-Likelihood functions*

---

## Description

Calculates the Log-Likelihood values for the BG/NBD model with and without covariates.

The function `bgnbd_nocov_LL_ind` calculates the individual LogLikelihood values for each customer for the given parameters.

The function `bgnbd_nocov_LL_sum` calculates the LogLikelihood value summed across customers for the given parameters.

The function `bgnbd_staticcov_LL_ind` calculates the individual LogLikelihood values for each customer for the given parameters and covariates.

The function `bgnbd_staticcov_LL_sum` calculates the individual LogLikelihood values summed across customers.

## Usage

```
bgnbd_nocov_LL_ind(vLogparams, vX, vT_x, vT_cal)

bgnbd_nocov_LL_sum(vLogparams, vX, vT_x, vT_cal)

bgnbd_staticcov_LL_ind(vParams, vX, vT_x, vT_cal, mCov_life, mCov_trans)

bgnbd_staticcov_LL_sum(vParams, vX, vT_x, vT_cal, mCov_life, mCov_trans)
```

## Arguments

| | |
|---|---|
| `vLogparams` | vector with the BG/NBD model parameters at log scale. See Details. |
| `vX` | Frequency vector of length n counting the numbers of purchases. |
| `vT_x` | Recency vector of length n. |
| `vT_cal` | Vector of length n indicating the total number of periods of observation. |

| | |
|---|---|
| vParams | vector with the parameters for the BG/NBD model at log scale and the static covariates at original scale. See Details. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |

## Details

vLogparams is a vector with model parameters r,alpha_0,a,b at log-scale, in this order.

vParams is vector with the BG/NBD model parameters at log scale, followed by the parameters for the lifetime covariates at original scale and then followed by the parameters for the transaction covariates at original scale

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vLogparams at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vLogparams at the respective position.

## Value

Returns the respective Log-Likelihood value(s) for the BG/NBD model with or without covariates.

## References

Fader PS, Hardie BGS, Lee, KL (2005). ""Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model" Marketing Science, 24(2), 275–284.

Fader PS, Hardie BGS (2013). "Overcoming the BG/NBD Model's #NUM! Error Problem" URL http://brucehardie.com/notes/027/bgnbd_num_error.pdf.

Fader PS, Hardie BGS (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

---

bgnbd_PAlive *BG/NBD: Probability of Being Alive*

---

## Description

Calculates the probability of a customer being alive at the end of the calibration period, based on a customer's past transaction behavior and the BG/NBD model parameters.

- bgnbd_nocov_PAlive P(alive) for the BG/NBD model without covariates
- bgnbd_staticcov_PAlive P(alive) for the BG/NBD model with static covariates

## Usage

```
bgnbd_nocov_PAlive(r, alpha, a, b, vX, vT_x, vT_cal)

bgnbd_staticcov_PAlive(
  r,
  alpha,
  a,
  b,
  vX,
  vT_x,
  vT_cal,
  vCovParams_trans,
  vCovParams_life,
  mCov_trans,
  mCov_life
)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process |
| alpha | scale parameter of the Gamma distribution of the purchase process |
| a | shape parameter of the Beta distribution of the lifetime process |
| b | shape parameter of the Beta distribution of the lifetime process |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| vCovParams_trans | Vector of estimated parameters for the transaction covariates. |
| vCovParams_life | Vector of estimated parameters for the lifetime covariates. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |

## Details

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_trans at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_life at the respective position.

## Value

Returns a vector with the PAlive for each customer.

## References

Fader PS, Hardie BGS, Lee, KL (2005). ""Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model" Marketing Science, 24(2), 275–284.

Fader PS, Hardie BGS (2013). "Overcoming the BG/NBD Model's #NUM! Error Problem" URL http://brucehardie.com/notes/027/bgnbd_num_error.pdf.

Fader PS, Hardie BGS (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

---

# cdnow *CDNOW dataset*

---

## Description

A dataset containing the entire purchase history up to the end of June 1998 of the cohort of 23,570 individuals who made their first-ever purchase at CDNOW in the first quarter of 1997.

## Usage

```
data("cdnow")
```

## Format

A `data.table` with 6696 rows and 4 variables:

`Id` Customer Id

`Date` Date of purchase

`CDs` Amount of CDs purchased

`Price` Price of purchase

## References

Fader, Peter S. and Bruce G.,S. Hardie, (2001), "Forecasting Repeat Sales at CDNOW: A Case Study," Interfaces, 31 (May-June), Part 2 of 2, p94-107.

clvdata *Create an object for transactional data required to estimate CLV*

## Description

Creates a data object that contains the prepared transaction data and that is used as input for model fitting. The transaction data may be split in an estimation and holdout sample if desired. The model then will only be fit on the estimation sample.

If covariates should be used when fitting a model, covariate data can be added to an object returned from this function.

## Usage

```
clvdata(
  data.transactions,
  date.format,
  time.unit,
  estimation.split = NULL,
  name.id = "Id",
  name.date = "Date",
  name.price = "Price"
)
```

## Arguments

| | |
|---|---|
| `data.transactions` | Transaction data as `data.frame` or `data.table`. See details. |
| `date.format` | Character string that indicates the format of the date variable in the data used. See details. |
| `time.unit` | What time unit defines a period. May be abbreviated, capitalization is ignored. See details. |
| `estimation.split` | Indicates the length of the estimation period. See details. |
| `name.id` | Column name of the customer id in `data.transaction`. |
| `name.date` | Column name of the transaction date in `data.transaction`. |
| `name.price` | Column name of price in `data.transaction`. NULL if no spending data is present. |

## Details

`data.transactions` A `data.frame` or `data.table` with customers' purchase history. Every transaction record consists of a purchase date and a customer id. Optionally, the price of the transaction may be included to also allow for prediction of future customer spending.

`time.unit` The definition of a single period. Currently available are `"hours"`, `"days"`, `"weeks"`, and `"years"`. May be abbreviated.

`date.format` A single format to use when parsing any date that is given as character input. This includes the dates given in `data.transaction`, `estimation.split`, or as an input to any other function at a later point, such as `prediction.end` in `predict`. The function `parse_date_time` of package `lubridate` is used to parse inputs and hence all formats it accepts in argument `orders` can be used. For example, a date of format "year-month-day" (i.e., "2010-06-17") is indicated with `"ymd"`. Other combinations such as `"dmy"`, `"dym"`, `"ymd HMS"`, or `"HMS dmy"` are possible as well.

`estimation.split` May be specified as either the number of periods since the first transaction or the timepoint (either as character, Date, or POSIXct) at which the estimation period ends. The indicated timepoint itself will be part of the estimation sample. If no value is provided or set to `NULL`, the whole dataset will used for fitting the model (no holdout sample).

**Aggregation of Transactions:**

Multiple transactions by the same customer that occur on the minimally representable temporal resolution are aggregated to a single transaction with their spending summed. For time units `days` and any other coarser `Date`-based time units (i.e. `weeks`, `years`), this means that transactions on the same day are combined. When using finer time units such as `hours` which are based on `POSIXct`, transactions on the same second are aggregated.

For the definition of repeat-purchases, combined transactions are viewed as a single transaction. Hence, repeat-transactions are determined from the aggregated transactions.

## Value

An object of class `clv.data`. See the class definition clv.data for more details about the returned object.

The function `summary` can be used to obtain and print a summary of the data. The generic accessor function `nobs` is available to read out the number of customers.

## See Also

`SetStaticCovariates` to add static covariates

`SetDynamicCovariates` for how to add dynamic covariates

`plot` to plot the repeat transactions

`summary` to summarize the transaction data

`pnbd` to fit Pareto/NBD models on a `clv.data` object

## Examples

```
data("cdnow")

# create clv data object with weekly periods
#    and no splitting
clv.data.cdnow <- clvdata(data.transactions = cdnow,
                          date.format="ymd",
                          time.unit = "weeks")

# same but split after 37 periods
clv.data.cdnow <- clvdata(data.transactions = cdnow,
```

```
                          date.format="ymd",
                          time.unit = "w",
                          estimation.split = 37)

# same but estimation end on the 15th Oct 1997
clv.data.cdnow <- clvdata(data.transactions = cdnow,
                          date.format="ymd",
                          time.unit = "w",
                          estimation.split = "1997-10-15")



# summary of the transaction data
summary(clv.data.cdnow)

# plot the transaction data
plot(clv.data.cdnow)

# create data with the weekly periods defined to
#   start on Mondays

## Not run:
# set start of week to Monday
oldopts <- options("lubridate.week.start"=1)

# create clv.data while Monday is the beginning of the week
clv.data.cdnow <- clvdata(data.transactions = cdnow,
                          date.format="ymd",
                          time.unit = "weeks")

# Dynamic covariates now have to be supplied for every Monday

# set week start to what it was before
options(oldopts)

## End(Not run)
```

---

`fitted.clv.fitted` *Extract Unconditional Expectation*

---

## Description

Extract the unconditional expectation (future transactions unconditional on being "alive") from a fitted clv model. This is the unconditional expectation data that is used when plotting the fitted model.

## Usage

```
## S3 method for class 'clv.fitted'
fitted(object, prediction.end = NULL, verbose = FALSE, ...)
```

## Arguments

| | |
|---|---|
| object | A fitted clv model for which the unconditional expectation is desired. |
| prediction.end | Until what point in time to predict. This can be the number of periods (numeric) or a form of date/time object. See details. |
| verbose | Show details about the running of the function. |
| ... | Ignored |

## Details

prediction.end indicates until when to predict or plot and can be given as either a point in time (of class Date, POSIXct, or character) or the number of periods. If prediction.end is of class character, the date/time format set when creating the data object is used for parsing. If prediction.end is the number of periods, the end of the fitting period serves as the reference point from which periods are counted. Only full periods may be specified. If prediction.end is omitted or NULL, it defaults to the end of the holdout period if present and to the end of the estimation period otherwise.

The first prediction period is defined to start right after the end of the estimation period. If for example weekly time units are used and the estimation period ends on Sunday 2019-01-01, then the first day of the first prediction period is Monday 2019-01-02. Each prediction period includes a total of 7 days and the first prediction period therefore will end on, and include, Sunday 2019-01-08. Subsequent prediction periods again start on Mondays and end on Sundays. If prediction.end indicates a timepoint on which to end, this timepoint is included in the prediction period.

## Value

A data.table which contains the following columns:

| | |
|---|---|
| period.until | The timepoint that marks the end (up until and including) of the period to which the data in this row refers. |
| period.num | The number of this period. |
| expectation | The value of the unconditional expectation for the period that ends on period.until. |

## See Also

plot to plot the unconditional expectation

gg *Gamma/Gamma Spending model*

## Description

Fits the Gamma-Gamma model on a given object of class `clv.data` to predict customers' mean spending per transaction.

## Usage

```
## S4 method for signature 'clv.data'
gg(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
  remove.first.transaction = TRUE,
  verbose = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `clv.data` | The data object on which the model is fitted. |
| `start.params.model` | Named start parameters containing the optimization start parameters for the model without covariates. |
| `optimx.args` | Additional arguments to control the optimization which are forwarded to `optimx::optimx`. If multiple optimization methods are specified, only the result of the last method is further processed. |
| `remove.first.transaction` | Whether customer's first transaction are removed. If `TRUE` all zero-repeaters are excluded from model fitting. |
| `verbose` | Show details about the running of the function. |
| `...` | Ignored |

## Details

Model parameters for the G/G model are `p`, `q`, and `gamma`.
`p`: shape parameter of the Gamma distribution of the spending process.
`q`: shape parameter of the Gamma distribution to account for customer heterogeneity.
`gamma`: scale parameter of the Gamma distribution to account for customer heterogeneity.
If no start parameters are given, 1.0 is used for all model parameters. All parameters are required to be > 0.

The Gamma-Gamma model cannot be estimated for data that contains negative prices. Customers with a mean spending of zero or a transaction count of zero are ignored during model fitting.

**The G/G model:** The G/G model allows to predict a value for future customer transactions. Usually, the G/G model is used in combination with a probabilistic model predicting customer transaction such as the Pareto/NBD or the BG/NBD model.

## Value

An object of class clv.gg is returned.

The function summary can be used to obtain and print a summary of the results. The generic accessor functions coefficients, vcov, fitted, logLik, AIC, BIC, and nobs are available.

## References

Colombo R, Jiang W (1999). "A stochastic RFM model." Journal of Interactive Marketing, 13(3), 2–12.

Fader PS, Hardie BG, Lee K (2005). "RFM and CLV: Using Iso-Value Curves for Customer Base Analysis." Journal of Marketing Research, 42(4), 415–430.

Fader PS, Hardie BG (2013). "The Gamma-Gamma Model of Monetary Value." URL http://www.brucehardie.com/notes/025/gamma_gamma.pdf.

## See Also

clvdata to create a clv data object.

predict to predict expected mean spending for every customer.

plot to plot the density of customer's mean transaction value compared to the model's prediction.

## Examples

```
data("apparelTrans")
clv.data.apparel <- clvdata(apparelTrans, date.format = "ymd",
                            time.unit = "w", estimation.split = 40)

# Fit the gg model
gg(clv.data.apparel)

# Give initial guesses for the model parameters
gg(clv.data.apparel,
     start.params.model = c(p=0.5, q=15, gamma=2))


# pass additional parameters to the optimizer (optimx)
#    Use Nelder-Mead as optimization method and print
#    detailed information about the optimization process
apparel.gg <- gg(clv.data.apparel,
                    optimx.args = list(method="Nelder-Mead",
                                       control=list(trace=6)))

# estimated coefs
coef(apparel.gg)
```

```
# summary of the fitted model
summary(apparel.gg)

# Plot model vs empirical distribution
plot(apparel.gg)

# predict mean spending and compare against
#    actuals in the holdout period
predict(apparel.gg)
```

---

ggomnbd *Gamma-Gompertz/NBD model*

---

## Description

Fits Gamma-Gompertz/NBD models on transactional data with static and without covariates.

## Usage

```
## S4 method for signature 'clv.data'
ggomnbd(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
  verbose = TRUE,
  ...
)

## S4 method for signature 'clv.data.static.covariates'
ggomnbd(
  clv.data,
  start.params.model = c(),
  optimx.args = list(),
  verbose = TRUE,
  names.cov.life = c(),
  names.cov.trans = c(),
  start.params.life = c(),
  start.params.trans = c(),
  names.cov.constr = c(),
  start.params.constr = c(),
  reg.lambdas = c(),
  ...
)
```

## Arguments

| | |
|---|---|
| clv.data | The data object on which the model is fitted. |
| start.params.model | Named start parameters containing the optimization start parameters for the model without covariates. |
| optimx.args | Additional arguments to control the optimization which are forwarded to optimx::optimx. If multiple optimization methods are specified, only the result of the last method is further processed. |
| verbose | Show details about the running of the function. |
| ... | Ignored |
| names.cov.life | Which of the set Lifetime covariates should be used. Missing parameter indicates all covariates shall be used. |
| names.cov.trans | Which of the set Transaction covariates should be used. Missing parameter indicates all covariates shall be used. |
| start.params.life | Named start parameters containing the optimization start parameters for all lifetime covariates. |
| start.params.trans | Named start parameters containing the optimization start parameters for all transaction covariates. |
| names.cov.constr | Which covariates should be forced to use the same parameters for the lifetime and transaction process. The covariates need to be present as both, lifetime and transaction covariates. |
| start.params.constr | Named start parameters containing the optimization start parameters for the constraint covariates. |
| reg.lambdas | Named lambda parameters used for the L2 regularization of the lifetime and the transaction covariate parameters. Lambdas have to be >= 0. |

## Details

Model parameters for the GGompertz /NBD model are r,alpha,beta,b and s.
r: shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process.
alpha: scale parameter of the Gamma distribution of the purchase process.
beta: scale parameter for the Gamma distribution for the lifetime process.
b: scale parameter of the Gompertz distribution (constant across customers).
s: shape parameter of the Gamma distribution for the lifetime process. The smaller s, the stronger the heterogeneity of customer lifetimes.

If no start parameters are given, r = 1, alpha = 1, beta = 1, b = 1, s = 1 is used. All model start parameters are required to be > 0. If no start values are given for the covariate parameters, 0.1 is used.

Note that the DERT expression has not been derived (yet) and it consequently is not possible to calculated values for DERT and CLV.

**The Gamma-Gompertz/NBD model:** There are two key differences of the gamma/Gompertz/NBD (GGom/NBD) model compared to the relative to the well-known Pareto/NBD model: (i) its probability density function can exhibit a mode at zero or an interior mode, and (ii) it can be skewed to the right or to the left. Therefore, the GGom/NBD model is more flexible than the Pareto/NBD model. According to Bemmaor and Glady (2012) can indicate substantial differences in expected residual lifetimes compared to the Pareto/NBD. The GGom/NBD tends to be appropriate when firms are reputed and their offerings are differentiated.

## Value

Depending on the data object on which the model was fit, ggomnbd returns either an object of class clv.ggomnbd or clv.ggomnbd.static.cov.

The function summary can be used to obtain and print a summary of the results. The generic accessor functions coefficients, vcov, fitted, logLik, AIC, BIC, and nobs are available.

## References

Bemmaor AC, Glady N (2012). "Modeling Purchasing Behavior with Sudden "Death": A Flexible Customer Lifetime Model" Management Science, 58(5), 1012-1021.

## See Also

clvdata to create a clv data object, SetStaticCovariates to add static covariates to an existing clv data object.

predict to predict expected transactions, probability of being alive, and customer lifetime value for every customer

plot to plot the unconditional expectation as predicted by the fitted model

The generic functions vcov, summary, fitted.

## Examples

```
data("apparelTrans")
clv.data.apparel <- clvdata(apparelTrans, date.format = "ymd",
                            time.unit = "w", estimation.split = 40)

# Fit standard ggomnbd model
ggomnbd(clv.data.apparel)

# Give initial guesses for the model parameters
ggomnbd(clv.data.apparel,
     start.params.model = c(r=0.5, alpha=15, b=5, beta=10, s=0.5))


# pass additional parameters to the optimizer (optimx)
#    Use Nelder-Mead as optimization method and print
#    detailed information about the optimization process
apparel.ggomnbd <- ggomnbd(clv.data.apparel,
                      optimx.args = list(method="Nelder-Mead",
                                         control=list(trace=6)))
```

```
# estimated coefs
coef(apparel.ggomnbd)

# summary of the fitted model
summary(apparel.ggomnbd)

# predict CLV etc for holdout period
predict(apparel.ggomnbd)

# predict CLV etc for the next 15 periods
predict(apparel.ggomnbd, prediction.end = 15)


# To estimate the ggomnbd model with static covariates,
#   add static covariates to the data
data("apparelStaticCov")
clv.data.static.cov <-
 SetStaticCovariates(clv.data.apparel,
                     data.cov.life = apparelStaticCov,
                     names.cov.life = c("Gender", "Channel"),
                     data.cov.trans = apparelStaticCov,
                     names.cov.trans = c("Gender", "Channel"))

# Fit ggomnbd with static covariates
ggomnbd(clv.data.static.cov)

# Give initial guesses for both covariate parameters
ggomnbd(clv.data.static.cov, start.params.trans = c(Gender=0.75, Channel=0.7),
                   start.params.life  = c(Gender=0.5, Channel=0.5))

# Use regularization
ggomnbd(clv.data.static.cov, reg.lambdas = c(trans = 5, life=5))

# Force the same coefficient to be used for both covariates
ggomnbd(clv.data.static.cov, names.cov.constr = "Gender",
                   start.params.constr = c(Gender=0.5))

# Fit model only with the Channel covariate for life but
# keep all trans covariates as is
ggomnbd(clv.data.static.cov, names.cov.life = c("Channel"))
```

---

ggomnbd_CET *GGompertz/NBD: Conditional Expected Transactions*

---

## Description

Calculates the expected number of transactions in a given time period based on a customer's past transaction behavior and the GGompertz/NBD model parameters.

- ggomnbd_nocov_CET Conditional Expected Transactions without covariates
- ggomnbd_staticcov_CET Conditional Expected Transactions with static covariates

## Usage

```
ggomnbd_nocov_CET(r, alpha_0, b, s, beta_0, dPeriods, vX, vT_x, vT_cal)

ggomnbd_staticcov_CET(
  r,
  alpha_0,
  b,
  s,
  beta_0,
  dPeriods,
  vX,
  vT_x,
  vT_cal,
  vCovParams_trans,
  vCovParams_life,
  mCov_life,
  mCov_trans
)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process. |
| alpha_0 | scale parameter of the Gamma distribution of the purchase process. |
| b | scale parameter of the Gompertz distribution (constant across customers) |
| s | shape parameter of the Gamma distribution for the lifetime process The smaller s, the stronger the heterogeneity of customer lifetimes. |
| beta_0 | scale parameter for the Gamma distribution for the lifetime process |
| dPeriods | number of periods to predict |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| vCovParams_trans | Vector of estimated parameters for the transaction covariates. |
| vCovParams_life | Vector of estimated parameters for the lifetime covariates. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |

## Details

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_trans at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_life at the respective position.

## Value

Returns a vector containing the conditional expected transactions for the existing customers in the GGompertz/NBD model.

## References

Bemmaor AC, Glady N (2012). "Modeling Purchasing Behavior with Sudden "Death": A Flexible Customer Lifetime Model" Management Science, 58(5), 1012-1021.

---

# ggomnbd_expectation *GGompertz/NBD: Unconditional Expectation*

---

## Description

Computes the expected number of repeat transactions in the interval (0, vT_i] for a randomly selected customer, where 0 is defined as the point when the customer came alive.

## Usage

```
ggomnbd_nocov_expectation(r, alpha_0, b, s, beta_0, vT_i)

ggomnbd_staticcov_expectation(r, b, s, vAlpha_i, vBeta_i, vT_i)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process. |
| alpha_0 | scale parameter of the Gamma distribution of the purchase process. |
| b | scale parameter of the Gompertz distribution (constant across customers) |
| s | shape parameter of the Gamma distribution for the lifetime process The smaller s, the stronger the heterogeneity of customer lifetimes. |
| beta_0 | scale parameter for the Gamma distribution for the lifetime process |
| vT_i | Number of periods since the customer came alive |
| vAlpha_i | Vector of individual parameters alpha |
| vBeta_i | Vector of individual parameters beta |

## Value

Returns the expected transaction values according to the chosen model.

## References

Bemmaor AC, Glady N (2012). "Modeling Purchasing Behavior with Sudden "Death": A Flexible Customer Lifetime Model" Management Science, 58(5), 1012-1021.

---

# ggomnbd_LL *GGompertz/NBD: Log-Likelihood functions*

---

## Description

Calculates the Log-Likelihood values for the GGompertz/NBD model with and without covariates.

The function ggomnbd_nocov_LL_ind calculates the individual LogLikelihood values for each customer for the given parameters.

The function ggomnbd_nocov_LL_sum calculates the LogLikelihood value summed across customers for the given parameters.

The function ggomnbd_staticcov_LL_ind calculates the individual LogLikelihood values for each customer for the given parameters and covariates.

The function ggomnbd_staticcov_LL_sum calculates the individual LogLikelihood values summed across customers.

## Usage

```
ggomnbd_nocov_LL_ind(vLogparams, vX, vT_x, vT_cal)

ggomnbd_nocov_LL_sum(vLogparams, vX, vT_x, vT_cal)

ggomnbd_staticcov_LL_ind(vParams, vX, vT_x, vT_cal, mCov_life, mCov_trans)

ggomnbd_staticcov_LL_sum(vParams, vX, vT_x, vT_cal, mCov_life, mCov_trans)
```

## Arguments

| | |
|---|---|
| vLogparams | vector with the GGompertz/NBD model parameters at log scale. See Details. |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| vParams | vector with the parameters for the GGompertz/NBD model at log scale and the static covariates at original scale. See Details. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |

## Details

vLogparams is a vector with model parameters r,alpha_0,b,s,beta_0 at log-scale, in this order.

vParams is vector with the GGompertz/NBD model parameters at log scale, followed by the parameters for the lifetime covariates at original scale and then followed by the parameters for the transaction covariates at original scale

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vParams at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vParams at the respective position.

## Value

Returns the respective Log-Likelihood value(s) for the GGompertz/NBD model with or without covariates.

## References

Bemmaor AC, Glady N (2012). "Modeling Purchasing Behavior with Sudden "Death": A Flexible Customer Lifetime Model" Management Science, 58(5), 1012-1021.

---

# ggomnbd_PAlive *GGompertz/NBD: Probability of Being Alive*

---

## Description

Calculates the probability of a customer being alive at the end of the calibration period, based on a customer's past transaction behavior and the GGompertz/NBD model parameters.

- ggomnbd_nocov_PAlive P(alive) for the GGompertz/NBD model without covariates
- ggomnbd_staticcov_PAlive P(alive) for the GGompertz/NBD model with static covariates

## Usage

```
ggomnbd_staticcov_PAlive(
  r,
  alpha_0,
  b,
  s,
  beta_0,
  vX,
  vT_x,
  vT_cal,
  vCovParams_trans,
  vCovParams_life,
```

```
  mCov_life,
  mCov_trans
)

ggomnbd_nocov_PAlive(r, alpha_0, b, s, beta_0, vX, vT_x, vT_cal)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process. |
| alpha_0 | scale parameter of the Gamma distribution of the purchase process. |
| b | scale parameter of the Gompertz distribution (constant across customers) |
| s | shape parameter of the Gamma distribution for the lifetime process The smaller s, the stronger the heterogeneity of customer lifetimes. |
| beta_0 | scale parameter for the Gamma distribution for the lifetime process |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| vCovParams_trans | Vector of estimated parameters for the transaction covariates. |
| vCovParams_life | Vector of estimated parameters for the lifetime covariates. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |

## Details

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_trans at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_life at the respective position.

## Value

Returns a vector with the PAlive for each customer.

## References

Bemmaor AC, Glady N (2012). "Modeling Purchasing Behavior with Sudden "Death": A Flexible Customer Lifetime Model" Management Science, 58(5), 1012-1021.

gg_LL *Gamma-Gamma: Log-Likelihood Function*

## Description

Calculates the Log-Likelihood value for the Gamma-Gamma model.

## Usage

```
gg_LL(vLogparams, vX, vM_x)
```

## Arguments

| | |
|---|---|
| vLogparams | a vector containing the log of the parameters p, q, gamma |
| vX | frequency vector of length n counting the numbers of purchases |
| vM_x | the observed average spending for every customer during the calibration time. |

## Details

vLogparams is a vector with the parameters for the Gamma-Gamma model. It has three parameters (p, q, gamma). The scale parameter for each transaction is distributed across customers according to a gamma distribution with parameters q (shape) and gamma (scale).

## Value

Returns the Log-Likelihood value for the Gamma-Gamma model.

## References

Colombo R, Jiang W (1999). "A stochastic RFM model." Journal of Interactive Marketing, 13(3), 2–12.

Fader PS, Hardie BG, Lee K (2005). "RFM and CLV: Using Iso-Value Curves for Customer Base Analysis." Journal of Marketing Research, 42(4), 415–430.

Fader PS, Hardie BG (2013). "The Gamma-Gamma Model of Monetary Value." URL http://www.brucehardie.com/notes/025/gamma_gamma.pdf.

## nobs.clv.data *Number of observations*

### Description

The number of observations is defined as the number of unique customers in the transaction data.

### Usage

```
## S3 method for class 'clv.data'
nobs(object, ...)
```

### Arguments

| | |
|---|---|
| object | An object of class clv.data. |
| ... | Ignored |

### Value

The number of customers.

## nobs.clv.fitted *Number of observations*

### Description

The number of observations is defined as the number of unique customers for which the model was fit.

### Usage

```
## S3 method for class 'clv.fitted'
nobs(object, ...)
```

### Arguments

| | |
|---|---|
| object | An object of class clv.fitted. |
| ... | Ignored |

### Value

The number of customers.

`plot.clv.data` *Plot actual repeat transactions*

## Description

Plots the actual repeat transactions for the given CLV data object.

## Usage

```
## S3 method for class 'clv.data'
plot(
  x,
  prediction.end = NULL,
  cumulative = FALSE,
  plot = TRUE,
  verbose = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | The clv data object to plot |
| `prediction.end` | Until what point in time to predict. This can be the number of periods (numeric) or a form of date/time object. See details. |
| `cumulative` | Whether the cumulative actual repeat transactions should be plotted. |
| `plot` | Whether a plot should be created or only the assembled data returned. |
| `verbose` | Show details about the running of the function. |
| `...` | Ignored |

## Details

`prediction.end` indicates until when to predict or plot and can be given as either a point in time (of class `Date`, `POSIXct`, or `character`) or the number of periods. If `prediction.end` is of class character, the date/time format set when creating the data object is used for parsing. If `prediction.end` is the number of periods, the end of the fitting period serves as the reference point from which periods are counted. Only full periods may be specified. If `prediction.end` is omitted or NULL, it defaults to the end of the holdout period if present and to the end of the estimation period otherwise.

The first prediction period is defined to start right after the end of the estimation period. If for example weekly time units are used and the estimation period ends on Sunday 2019-01-01, then the first day of the first prediction period is Monday 2019-01-02. Each prediction period includes a total of 7 days and the first prediction period therefore will end on, and include, Sunday 2019-01-08. Subsequent prediction periods again start on Mondays and end on Sundays. If `prediction.end` indicates a timepoint on which to end, this timepoint is included in the prediction period.

If there are no repeat transactions until `prediction.end`, only the time for which there is data is plotted. If the data is returned (i.e. with argument `plot=FALSE`), the respective rows contain `NA` in column `Number of Repeat Transactions`.

## Value

An object of class ggplot from package ggplot2 is returned by default. If the parameter plot is FALSE, the data that would have been melted and used to create the plot is returned. It is a data.table which contains the following columns:

period.until The timepoint that marks the end (up until and including) of the period to which the data in this row refers.

Number of Repeat Transactions
The number of actual repeat transactions in the period that ends at period.until.

## Examples

```
data("cdnow")
clv.data.cdnow <- clvdata(cdnow, time.unit="w",
                          estimation.split=37,
                          date.format="ymd")

# Plot the actual repeat transactions
plot(clv.data.cdnow)

# plot cumulative repeat transactions
plot(clv.data.cdnow, cumulative=TRUE)

# Dont automatically plot but tweak further
gg.cdnow <- plot(clv.data.cdnow)

# change Title
library(ggplot2)
gg.cdnow + ggtitle("CDnow repeat transactions")

# Dont return a plot but only the data from
#   which it would have been created
dt.plot.data <- plot(clv.data.cdnow, plot=FALSE)
```

---

# plot.clv.fitted.spending

*Plot expected and actual mean spending per transaction*

---

## Description

Compares the density of the observed average spending per transaction (empirical distribution) to the model's distribution of mean transaction spending (weighted by the actual number of transactions).

## Usage

```
## S3 method for class 'clv.fitted.spending'
plot(x, n = 256, verbose = TRUE, ...)

## S4 method for signature 'clv.fitted.spending'
plot(x, n = 256, verbose = TRUE, ...)
```

## Arguments

| | |
|---|---|
| x | The fitted spending model to plot |
| n | Number of points at which the empirical and model density are calculated. Should be a power of two. |
| verbose | Show details about the running of the function. |
| ... | Ignored |

## Value

An object of class ggplot from package ggplot2 is returned by default.

## References

Colombo R, Jiang W (1999). "A stochastic RFM model." Journal of Interactive Marketing, 13(3), 2–12.

Fader PS, Hardie BG, Lee K (2005). "RFM and CLV: Using Iso-Value Curves for Customer Base Analysis." Journal of Marketing Research, 42(4), 415–430.

Fader PS, Hardie BG (2013). "The Gamma-Gamma Model of Monetary Value." URL http://www.brucehardie.com/notes/025/gamma_gamma.pdf.

## See Also

plot for transaction models

## Examples

```
data("cdnow")

clv.cdnow <- clvdata(cdnow,
  date.format="ymd",
  time.unit = "week",
  estimation.split = "1997-09-30")

est.gg <- gg(clv.data = clv.cdnow)

# Compare empirical to theoretical distribution
plot(est.gg)

## Not run:
# Modify the created plot further
```

```
library(ggplot2)
gg.cdnow <- plot(est.gg)
gg.cdnow + ggtitle("CDnow Spending Distribution")

## End(Not run)
```

---

`plot.clv.fitted.transactions` *Plot expected and actual repeat transactions*

---

## Description

Plot the actual repeat transactions and overlay it with the repeat transaction as predicted by the fitted model. Currently, following previous literature, the in-sample unconditional expectation is plotted in the holdout period. In the future, we might add the option to also plot the summed CET for the holdout period as an alternative evaluation metric.

## Usage

```
## S3 method for class 'clv.fitted.transactions'
plot(
  x,
  prediction.end = NULL,
  newdata = NULL,
  cumulative = FALSE,
  transactions = TRUE,
  label = NULL,
  plot = TRUE,
  verbose = TRUE,
  ...
)

## S4 method for signature 'clv.fitted.transactions'
plot(
  x,
  prediction.end = NULL,
  newdata = NULL,
  cumulative = FALSE,
  transactions = TRUE,
  label = NULL,
  plot = TRUE,
  verbose = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | The fitted clv model to plot |
| prediction.end | Until what point in time to predict. This can be the number of periods (numeric) or a form of date/time object. See details. |
| newdata | An object of class clv.data for which the plotting should be made with the fitted model. If none or NULL is given, the plot is made for the data on which the model was fit. |
| cumulative | Whether the cumulative expected (and actual) transactions should be plotted. |
| transactions | Whether the actual observed repeat transactions should be plotted. |
| label | Character string to label the model in the legend |
| plot | Whether a plot should be created or only the assembled data is returned. |
| verbose | Show details about the running of the function. |
| ... | Ignored |

## Details

prediction.end indicates until when to predict or plot and can be given as either a point in time (of class Date, POSIXct, or character) or the number of periods. If prediction.end is of class character, the date/time format set when creating the data object is used for parsing. If prediction.end is the number of periods, the end of the fitting period serves as the reference point from which periods are counted. Only full periods may be specified. If prediction.end is omitted or NULL, it defaults to the end of the holdout period if present and to the end of the estimation period otherwise.

The first prediction period is defined to start right after the end of the estimation period. If for example weekly time units are used and the estimation period ends on Sunday 2019-01-01, then the first day of the first prediction period is Monday 2019-01-02. Each prediction period includes a total of 7 days and the first prediction period therefore will end on, and include, Sunday 2019-01-08. Subsequent prediction periods again start on Mondays and end on Sundays. If prediction.end indicates a timepoint on which to end, this timepoint is included in the prediction period.

Note that only whole periods can be plotted and that the prediction end might not exactly match prediction.end. See the Note section for more details.

The newdata argument has to be a clv data object of the exact same class as the data object on which the model was fit. In case the model was fit with covariates, newdata needs to contain identically named covariate data.

The use case for newdata is mainly two-fold: First, to estimate model parameters only on a sample of the data and then use the fitted model object to predict or plot for the full data set provided through newdata. Second, for models with dynamic covariates, to provide a clv data object with longer covariates than contained in the data on which the model was estimated what allows to predict or plot further. When providing newdata, some models might require additional steps that can significantly increase runtime.

## Value

An object of class ggplot from package ggplot2 is returned by default. If the parameter plot is FALSE, the data that would have been melted and used to create the plot is returned. It is a data.table which contains the following columns:

`period.until` The timepoint that marks the end (up until and including) of the period to which the data in this row refers.

`Number of Repeat Transactions`
The number of actual repeat transactions in the period that ends at `period.until`. Only if `transactions` is `TRUE`.

`"Name of Model" or "label"`
The value of the unconditional expectation for the period that ends on `period.until`.

## Note

Because the unconditional expectation for a period is derived as the difference of the cumulative expectations calculated at the beginning and at end of the period, all timepoints for which the expectation is calculated are required to be spaced exactly 1 time unit apart.

If `prediction.end` does not coincide with the start of a time unit, the last timepoint for which the expectation is calculated and plotted therefore is not `prediction.end` but the start of the first time unit after `prediction.end`.

## See Also

`plot` for spending models

## Examples

```
data("cdnow")

# Fit ParetoNBD model on the CDnow data
pnbd.cdnow <- pnbd(clvdata(cdnow, time.unit="w",
                           estimation.split=37,
                           date.format="ymd"))

# Plot actual repeat transaction, overlayed with the
#  expected repeat transactions as by the fitted model
plot(pnbd.cdnow)

# Plot cumulative expected transactions of only the model
plot(pnbd.cdnow, cumulative=TRUE, transactions=FALSE)

# Plot forecast until 2001-10-21
plot(pnbd.cdnow, prediction.end = "2001-10-21")

# Plot until 2001-10-21, as date
plot(pnbd.cdnow,
     prediction.end = lubridate::dym("21-2001-10"))

# Plot 15 time units after end of estimation period
plot(pnbd.cdnow, prediction.end = 15)

# Save the data generated for plotting
#   (period, actual transactions, expected transactions)
```

```
plot.out <- plot(pnbd.cdnow, prediction.end = 15)

# A ggplot object is returned that can be further tweaked
library("ggplot2")
gg.pnbd.cdnow <- plot(pnbd.cdnow)
gg.pnbd.cdnow + ggtitle("PNBD on CDnow")
```

---

pnbd *Pareto/NBD models*

---

## Description

Fits Pareto/NBD models on transactional data with and without covariates.

## Usage

```
## S4 method for signature 'clv.data'
pnbd(
  clv.data,
  start.params.model = c(),
  use.cor = FALSE,
  start.param.cor = c(),
  optimx.args = list(),
  verbose = TRUE,
  ...
)

## S4 method for signature 'clv.data.static.covariates'
pnbd(
  clv.data,
  start.params.model = c(),
  use.cor = FALSE,
  start.param.cor = c(),
  optimx.args = list(),
  verbose = TRUE,
  names.cov.life = c(),
  names.cov.trans = c(),
  start.params.life = c(),
  start.params.trans = c(),
  names.cov.constr = c(),
  start.params.constr = c(),
  reg.lambdas = c(),
  ...
)
```

```
## S4 method for signature 'clv.data.dynamic.covariates'
pnbd(
  clv.data,
  start.params.model = c(),
  use.cor = FALSE,
  start.param.cor = c(),
  optimx.args = list(),
  verbose = TRUE,
  names.cov.life = c(),
  names.cov.trans = c(),
  start.params.life = c(),
  start.params.trans = c(),
  names.cov.constr = c(),
  start.params.constr = c(),
  reg.lambdas = c(),
  ...
)
```

## Arguments

| | |
|---|---|
| clv.data | The data object on which the model is fitted. |
| start.params.model | Named start parameters containing the optimization start parameters for the model without covariates. |
| use.cor | Whether the correlation between the transaction and lifetime process should be estimated. |
| start.param.cor | Start parameter for the optimization of the correlation. |
| optimx.args | Additional arguments to control the optimization which are forwarded to optimx::optimx. If multiple optimization methods are specified, only the result of the last method is further processed. |
| verbose | Show details about the running of the function. |
| ... | Ignored |
| names.cov.life | Which of the set Lifetime covariates should be used. Missing parameter indicates all covariates shall be used. |
| names.cov.trans | Which of the set Transaction covariates should be used. Missing parameter indicates all covariates shall be used. |
| start.params.life | Named start parameters containing the optimization start parameters for all lifetime covariates. |
| start.params.trans | Named start parameters containing the optimization start parameters for all transaction covariates. |
| names.cov.constr | Which covariates should be forced to use the same parameters for the lifetime and transaction process. The covariates need to be present as both, lifetime and transaction covariates. |

`start.params.constr`
: Named start parameters containing the optimization start parameters for the constraint covariates.

`reg.lambdas`
: Named lambda parameters used for the L2 regularization of the lifetime and the transaction covariate parameters. Lambdas have to be >= 0.

## Details

Model parameters for the Pareto/NBD model are `alpha`, `r`, `beta`, and `s`.
`s`: shape parameter of the Gamma distribution for the lifetime process. The smaller s, the stronger the heterogeneity of customer lifetimes.
`beta`: rate parameter for the Gamma distribution for the lifetime process.
`r`: shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process.
`alpha`: rate parameter of the Gamma distribution of the purchase process.

Based on these parameters, the average purchase rate while customers are active is r/alpha and the average dropout rate is s/beta.

Ideally, the starting parameters for r and s represent your best guess concerning the heterogeneity of customers in their buy and die rate. If covariates are included into the model additionally parameters for the covariates affecting the attrition and the purchase process are part of the model.

If no start parameters are given, 1.0 is used for all model parameters and 0.1 for covariate parameters. The model start parameters are required to be > 0.

**The Pareto/NBD model:** The Pareto/NBD is the first model addressing the issue of modeling customer purchases and attrition simultaneously for non-contractual settings. The model uses a Pareto distribution, a combination of an Exponential and a Gamma distribution, to explicitly model customers' (unobserved) attrition behavior in addition to customers' purchase process.
In general, the Pareto/NBD model consist of two parts. A first process models the purchase behavior of customers as long as the customers are active. A second process models customers' attrition. Customers live (and buy) for a certain unknown time until they become inactive and "die". Customer attrition is unobserved. Inactive customers may not be reactivated. For technical details we refer to the original paper by Schmittlein, Morrison and Colombo (1987) and the detailed technical note of Fader and Hardie (2005).

**Pareto/NBD model with static covariates:** The standard Pareto/NBD model captures heterogeneity was solely using Gamma distributions. However, often exogenous knowledge, such as for example customer demographics, is available. The supplementary knowledge may explain part of the heterogeneity among the customers and therefore increase the predictive accuracy of the model. In addition, we can rely on these parameter estimates for inference, i.e. identify and quantify effects of contextual factors on the two underlying purchase and attrition processes. For technical details we refer to the technical note by Fader and Hardie (2007).

**Pareto/NBD model with dynamic covariates:** In many real-world applications customer purchase and attrition behavior may be influenced by covariates that vary over time. In consequence, the timing of a purchase and the corresponding value of at covariate a that time becomes relevant. Time-varying covariates can affect customer on aggregated level as well as on an individual level: In the first case, all customers are affected simultaneously, in the latter case a covariate is only relevant for a particular customer. For technical details we refer to the paper by Bachmann, Meierer and Näf (2020).

## Value

Depending on the data object on which the model was fit, pnbd returns either an object of class clv.pnbd, clv.pnbd.static.cov, or clv.pnbd.dynamic.cov.

The function summary can be used to obtain and print a summary of the results. The generic accessor functions coefficients, vcov, fitted, logLik, AIC, BIC, and nobs are available.

## Note

The Pareto/NBD model with dynamic covariates can currently not be fit with data that has a temporal resolution of less than one day (data that was built with time unit hours).

## References

Schmittlein DC, Morrison DG, Colombo R (1987). "Counting Your Customers: Who-Are They and What Will They Do Next?" Management Science, 33(1), 1–24.

Fader PS, Hardie BGS (2005). "A Note on Deriving the Pareto/NBD Model and Related Expressions." URL http://www.brucehardie.com/notes/009/pareto_nbd_derivations_2005-11-05.pdf.

Fader PS, Hardie BG (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

## See Also

clvdata to create a clv data object, SetStaticCovariates to add static covariates to an existing clv data object.

predict to predict expected transactions, probability of being alive, and customer lifetime value for every customer

plot to plot the unconditional expectation as predicted by the fitted model

The generic functions vcov, summary, fitted.

SetDynamicCovariates to add dynamic covariates on which the pnbd model can be fit.

## Examples

```
data("apparelTrans")
clv.data.apparel <- clvdata(apparelTrans, date.format = "ymd",
                            time.unit = "w", estimation.split = 40)

# Fit standard pnbd model
pnbd(clv.data.apparel)

# Give initial guesses for the model parameters
pnbd(clv.data.apparel,
     start.params.model = c(r=0.5, alpha=15, s=0.5, beta=10))


# pass additional parameters to the optimizer (optimx)
```

```
#    Use Nelder-Mead as optimization method and print
#    detailed information about the optimization process
apparel.pnbd <- pnbd(clv.data.apparel,
                     optimx.args = list(method="Nelder-Mead",
                                        control=list(trace=6)))

# estimated coefs
coef(apparel.pnbd)

# summary of the fitted model
summary(apparel.pnbd)

# predict CLV etc for holdout period
predict(apparel.pnbd)

# predict CLV etc for the next 15 periods
predict(apparel.pnbd, prediction.end = 15)


# Estimate correlation as well
pnbd(clv.data.apparel, use.cor = TRUE)


# To estimate the pnbd model with static covariates,
#   add static covariates to the data
data("apparelStaticCov")
clv.data.static.cov <-
 SetStaticCovariates(clv.data.apparel,
                     data.cov.life = apparelStaticCov,
                     names.cov.life = c("Gender", "Channel"),
                     data.cov.trans = apparelStaticCov,
                     names.cov.trans = c("Gender", "Channel"))

# Fit pnbd with static covariates
pnbd(clv.data.static.cov)

# Give initial guesses for both covariate parameters
pnbd(clv.data.static.cov, start.params.trans = c(Gender=0.75, Channel=0.7),
                   start.params.life  = c(Gender=0.5, Channel=0.5))

# Use regularization
pnbd(clv.data.static.cov, reg.lambdas = c(trans = 5, life=5))

# Force the same coefficient to be used for both covariates
pnbd(clv.data.static.cov, names.cov.constr = "Gender",
                   start.params.constr = c(Gender=0.5))

# Fit model only with the Channel covariate for life but
# keep all trans covariates as is
pnbd(clv.data.static.cov, names.cov.life = c("Channel"))

# Add dynamic covariates data to the data object
#   add dynamic covariates to the data
```

```
## Not run:
data("apparelDynCov")
clv.data.dyn.cov <-
  SetDynamicCovariates(clv.data = clv.data.apparel,
                       data.cov.life = apparelDynCov,
                       data.cov.trans = apparelDynCov,
                       names.cov.life = c("Marketing", "Gender", "Channel"),
                       names.cov.trans = c("Marketing", "Gender", "Channel"),
                       name.date = "Cov.Date")


# Fit PNBD with dynamic covariates
pnbd(clv.data.dyn.cov)

# The same fitting options as for the
#  static covariate are available
pnbd(clv.data.dyn.cov, reg.lambdas = c(trans=10, life=2))

## End(Not run)
```

---

pnbd_CET *Pareto/NBD: Conditional Expected Transactions*

---

## Description

Calculates the expected number of transactions in a given time period based on a customer's past transaction behavior and the Pareto/NBD model parameters.

- pnbd_nocov_CET Conditional Expected Transactions without covariates
- pnbd_staticcov_CET Conditional Expected Transactions with static covariates

## Usage

```
pnbd_nocov_CET(r, alpha_0, s, beta_0, dPeriods, vX, vT_x, vT_cal)

pnbd_staticcov_CET(
  r,
  alpha_0,
  s,
  beta_0,
  dPeriods,
  vX,
  vT_x,
  vT_cal,
  vCovParams_trans,
  vCovParams_life,
```

```
  mCov_trans,
  mCov_life
)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process |
| alpha_0 | rate parameter of the Gamma distribution of the purchase process |
| s | shape parameter of the Gamma distribution for the lifetime process. The smaller s, the stronger the heterogeneity of customer lifetimes |
| beta_0 | rate parameter for the Gamma distribution for the lifetime process. |
| dPeriods | number of periods to predict |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| vCovParams_trans | Vector of estimated parameters for the transaction covariates. |
| vCovParams_life | Vector of estimated parameters for the lifetime covariates. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |

## Details

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_trans at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_life at the respective position.

## Value

Returns a vector containing the conditional expected transactions for the existing customers in the Pareto/NBD model.

## References

Schmittlein DC, Morrison DG, Colombo R (1987). "Counting Your Customers: Who-Are They and What Will They Do Next?" Management Science, 33(1), 1–24.

Fader PS, Hardie BGS (2005). "A Note on Deriving the Pareto/NBD Model and Related Expressions." URL http://www.brucehardie.com/notes/009/pareto_nbd_derivations_2005-11-05.pdf.

Fader PS, Hardie BG (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

---

pnbd_DERT *Pareto/NBD: Discounted Expected Residual Transactions*

---

## Description

Calculates the discounted expected residual transactions.

- pnbd_nocov_DERT Discounted expected residual transactions for the Pareto/NBD model without covariates
- pnbd_staticcov_DERT Discounted expected residual transactions for the Pareto/NBD model with static covariates

## Usage

```
pnbd_nocov_DERT(
  r,
  alpha_0,
  s,
  beta_0,
  continuous_discount_factor,
  vX,
  vT_x,
  vT_cal
)

pnbd_staticcov_DERT(
  r,
  alpha_0,
  s,
  beta_0,
  continuous_discount_factor,
  vX,
  vT_x,
  vT_cal,
  mCov_life,
  mCov_trans,
  vCovParams_life,
  vCovParams_trans
)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process |
| alpha_0 | rate parameter of the Gamma distribution of the purchase process |
| s | shape parameter of the Gamma distribution for the lifetime process. The smaller s, the stronger the heterogeneity of customer lifetimes |
| beta_0 | rate parameter for the Gamma distribution for the lifetime process. |
| continuous_discount_factor | continuous discount factor to use |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |
| vCovParams_life | Vector of estimated parameters for the lifetime covariates. |
| vCovParams_trans | Vector of estimated parameters for the transaction covariates. |

## Details

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_trans at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vCovParams_life at the respective position.

## Value

Returns a vector with the DERT for each customer.

## References

Schmittlein DC, Morrison DG, Colombo R (1987). "Counting Your Customers: Who-Are They and What Will They Do Next?" Management Science, 33(1), 1–24.

Fader PS, Hardie BGS (2005). "A Note on Deriving the Pareto/NBD Model and Related Expressions." URL http://www.brucehardie.com/notes/009/pareto_nbd_derivations_2005-11-05.pdf.

Fader PS, Hardie BG (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

pnbd_expectation *Pareto/NBD: Unconditional Expectation*

## Description

Computes the expected number of repeat transactions in the interval (0, vT_i] for a randomly selected customer, where 0 is defined as the point when the customer came alive.

## Usage

```
pnbd_nocov_expectation(r, s, alpha_0, beta_0, vT_i)

pnbd_staticcov_expectation(r, s, vAlpha_i, vBeta_i, vT_i)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process |
| s | shape parameter of the Gamma distribution for the lifetime process. The smaller s, the stronger the heterogeneity of customer lifetimes |
| alpha_0 | rate parameter of the Gamma distribution of the purchase process |
| beta_0 | rate parameter for the Gamma distribution for the lifetime process. |
| vT_i | Number of periods since the customer came alive |
| vAlpha_i | Vector of individual parameters alpha |
| vBeta_i | Vector of individual parameters beta |

## Value

Returns the expected transaction values according to the chosen model.

## References

Schmittlein DC, Morrison DG, Colombo R (1987). "Counting Your Customers: Who-Are They and What Will They Do Next?" Management Science, 33(1), 1–24.

Fader PS, Hardie BGS (2005). "A Note on Deriving the Pareto/NBD Model and Related Expressions." URL http://www.brucehardie.com/notes/009/pareto_nbd_derivations_2005-11-05.pdf.

Fader PS, Hardie BG (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

pnbd_LL *Pareto/NBD: Log-Likelihood functions*

## Description

Calculates the Log-Likelihood values for the Pareto/NBD model with and without covariates.

The function pnbd_nocov_LL_ind calculates the individual LogLikelihood values for each customer for the given parameters.

The function pnbd_nocov_LL_sum calculates the LogLikelihood value summed across customers for the given parameters.

The function pnbd_staticcov_LL_ind calculates the individual LogLikelihood values for each customer for the given parameters and covariates.

The function pnbd_staticcov_LL_sum calculates the individual LogLikelihood values summed across customers.

## Usage

```
pnbd_nocov_LL_ind(vLogparams, vX, vT_x, vT_cal)

pnbd_nocov_LL_sum(vLogparams, vX, vT_x, vT_cal)

pnbd_staticcov_LL_ind(vParams, vX, vT_x, vT_cal, mCov_life, mCov_trans)

pnbd_staticcov_LL_sum(vParams, vX, vT_x, vT_cal, mCov_life, mCov_trans)
```

## Arguments

| | |
|---|---|
| vLogparams | vector with the Pareto/NBD model parameters at log scale. See Details. |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| vParams | vector with the parameters for the Pareto/NBD model at log scale and the static covariates at original scale. See Details. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |

## Details

vLogparams is a vector with model parameters r,alpha_0,s,beta_0 at log-scale, in this order.

vParams is vector with the Pareto/NBD model parameters at log scale, followed by the parameters for the lifetime covariates at original scale and then followed by the parameters for the transaction covariates at original scale

mCov_trans is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to vParams at the respective position.

mCov_life is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to vParams at the respective position.

## Value

Returns the respective Log-Likelihood value(s) for the Pareto/NBD model with or without covariates.

## References

Schmittlein DC, Morrison DG, Colombo R (1987). "Counting Your Customers: Who-Are They and What Will They Do Next?" Management Science, 33(1), 1–24.

Fader PS, Hardie BGS (2005). "A Note on Deriving the Pareto/NBD Model and Related Expressions." URL http://www.brucehardie.com/notes/009/pareto_nbd_derivations_2005-11-05.pdf.

Fader PS, Hardie BG (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

---

# pnbd_PAlive *Pareto/NBD: Probability of Being Alive*

---

## Description

Calculates the probability of a customer being alive at the end of the calibration period, based on a customer's past transaction behavior and the Pareto/NBD model parameters.

- pnbd_nocov_PAlive P(alive) for the Pareto/NBD model without covariates
- pnbd_staticcov_PAlive P(alive) for the Pareto/NBD model with static covariates

## Usage

```
pnbd_nocov_PAlive(r, alpha_0, s, beta_0, vX, vT_x, vT_cal)

pnbd_staticcov_PAlive(
  r,
  alpha_0,
  s,
  beta_0,
  vX,
  vT_x,
  vT_cal,
```

```
  vCovParams_trans,
  vCovParams_life,
  mCov_trans,
  mCov_life
)
```

## Arguments

| | |
|---|---|
| r | shape parameter of the Gamma distribution of the purchase process. The smaller r, the stronger the heterogeneity of the purchase process |
| alpha_0 | rate parameter of the Gamma distribution of the purchase process |
| s | shape parameter of the Gamma distribution for the lifetime process. The smaller s, the stronger the heterogeneity of customer lifetimes |
| beta_0 | rate parameter for the Gamma distribution for the lifetime process. |
| vX | Frequency vector of length n counting the numbers of purchases. |
| vT_x | Recency vector of length n. |
| vT_cal | Vector of length n indicating the total number of periods of observation. |
| vCovParams_trans | Vector of estimated parameters for the transaction covariates. |
| vCovParams_life | Vector of estimated parameters for the lifetime covariates. |
| mCov_trans | Matrix containing the covariates data affecting the transaction process. One column for each covariate. |
| mCov_life | Matrix containing the covariates data affecting the lifetime process. One column for each covariate. |

## Details

`mCov_trans` is a matrix containing the covariates data of the time-invariant covariates that affect the transaction process. Each column represents a different covariate. For every column a gamma parameter needs to added to `vCovParams_trans` at the respective position.

`mCov_life` is a matrix containing the covariates data of the time-invariant covariates that affect the lifetime process. Each column represents a different covariate. For every column a gamma parameter needs to added to `vCovParams_life` at the respective position.

## Value

Returns a vector with the PAlive for each customer.

## References

Schmittlein DC, Morrison DG, Colombo R (1987). "Counting Your Customers: Who-Are They and What Will They Do Next?" Management Science, 33(1), 1–24.

Fader PS, Hardie BGS (2005). "A Note on Deriving the Pareto/NBD Model and Related Expressions." URL http://www.brucehardie.com/notes/009/pareto_nbd_derivations_2005-11-05.pdf.

Fader PS, Hardie BG (2007). "Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models." URL http://www.brucehardie.com/notes/019/time_invariant_covariates.pdf.

---

`predict.clv.fitted.spending`

*Predict customers' future spending*

---

## Description

Predict customer's future mean spending per transaction and compare it to the actual mean spending in the holdout period.

## Usage

```
## S3 method for class 'clv.fitted.spending'
predict(object, newdata = NULL, verbose = TRUE, ...)

## S4 method for signature 'clv.fitted.spending'
predict(object, newdata = NULL, verbose = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `object` | A fitted spending model for which prediction is desired. |
| `newdata` | A clv data object for which predictions should be made with the fitted model. If none or NULL is given, predictions are made for the data on which the model was fit. |
| `verbose` | Show details about the running of the function. |
| `...` | Ignored |

## Details

If `newdata` is provided, the individual customer statistics underlying the model are calculated the same way as when the model was fit initially. Hence, if `remove.first.transaction` was `TRUE`, this will be applied to `newdata` as well.

## Value

An object of class `data.table` with columns:

| | |
|---|---|
| `Id` | The respective customer identifier |
| `actual.mean.spending` | Actual mean spending per transaction in the holdout period. Only if there is a holdout period otherwise it is not reported. |
| `predicted.mean.spending` | The mean spending per transaction as predicted by the fitted spending model. |

## See Also

models to predict spending: gg.

models to predict transactions: pnbd, bgnbd, ggomnbd.

predict for transaction models

## Examples

```
data("apparelTrans")

# Fit gg model on data
apparel.holdout <- clvdata(apparelTrans, time.unit="w",
                           estimation.split=37, date.format="ymd")
apparel.gg <- gg(apparel.holdout)

# Predict customers' future mean spending per transaction
predict(apparel.gg)
```

---

# predict.clv.fitted.transactions

*Predict CLV from a fitted transaction model*

---

## Description

Probabilistic customer attrition models predict in general three expected characteristics for every customer:

- "conditional expected transactions" (CET), which is the number of transactions to expect from a customer during the prediction period,
- "probability of a customer being alive" (PAlive) at the end of the estimation period and
- "discounted expected residual transactions" (DERT) for every customer, which is the total number of transactions for the residual lifetime of a customer discounted to the end of the estimation period. In the case of time-varying covariates, instead of DERT, "discounted expected conditional transactions" (DECT) is predicted. DECT does only cover a finite time horizon in contrast to DERT. For continuous.discount.factor=0, DECT corresponds to CET.

In order to derive a monetary value such as CLV, customer spending has to be considered. If the clv.data object contains spending information, customer spending can be predicted using a Gamma/Gamma spending model for parameter predict.spending and the predicted CLV is be calculated (if the transaction model supports DERT/DECT). In this case, the prediction additionally contains the following two columns:

- "predicted.mean.spending", the mean spending per transactions as predicted by the spending model.
- "CLV", the customer lifetime value. CLV is the product of DERT/DECT and predicted spending.

## Usage

```
## S3 method for class 'clv.fitted.transactions'
predict(
  object,
  newdata = NULL,
  prediction.end = NULL,
  predict.spending = gg,
  continuous.discount.factor = 0.1,
  verbose = TRUE,
  ...
)

## S4 method for signature 'clv.fitted.transactions'
predict(
  object,
  newdata = NULL,
  prediction.end = NULL,
  predict.spending = gg,
  continuous.discount.factor = 0.1,
  verbose = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| object | A fitted clv transaction model for which prediction is desired. |
| newdata | A clv data object for which predictions should be made with the fitted model. If none or NULL is given, predictions are made for the data on which the model was fit. |
| prediction.end | Until what point in time to predict. This can be the number of periods (numeric) or a form of date/time object. See details. |
| predict.spending | Whether and how to predict spending and based on it also CLV, if possible. See details. |
| continuous.discount.factor | continuous discount factor to use to calculate DERT/DECT |
| verbose | Show details about the running of the function. |
| ... | Ignored |

## Details

predict.spending indicates whether to predict customers' spending and if so, the spending model to use. Accepted inputs are either a logical (TRUE/FALSE), a method to fit a spending model (i.e. gg), or an already fitted spending model. If provided TRUE, a Gamma-Gamma model is fit with default options. If argument newdata is provided, the spending model is fit on newdata. Predicting spending is only possible if the transaction data contains spending information. See examples for illustrations of valid inputs.

The newdata argument has to be a clv data object of the exact same class as the data object on which the model was fit. In case the model was fit with covariates, newdata needs to contain identically named covariate data.

The use case for newdata is mainly two-fold: First, to estimate model parameters only on a sample of the data and then use the fitted model object to predict or plot for the full data set provided through newdata. Second, for models with dynamic covariates, to provide a clv data object with longer covariates than contained in the data on which the model was estimated what allows to predict or plot further. When providing newdata, some models might require additional steps that can significantly increase runtime.

prediction.end indicates until when to predict or plot and can be given as either a point in time (of class Date, POSIXct, or character) or the number of periods. If prediction.end is of class character, the date/time format set when creating the data object is used for parsing. If prediction.end is the number of periods, the end of the fitting period serves as the reference point from which periods are counted. Only full periods may be specified. If prediction.end is omitted or NULL, it defaults to the end of the holdout period if present and to the end of the estimation period otherwise.

The first prediction period is defined to start right after the end of the estimation period. If for example weekly time units are used and the estimation period ends on Sunday 2019-01-01, then the first day of the first prediction period is Monday 2019-01-02. Each prediction period includes a total of 7 days and the first prediction period therefore will end on, and include, Sunday 2019-01-08. Subsequent prediction periods again start on Mondays and end on Sundays. If prediction.end indicates a timepoint on which to end, this timepoint is included in the prediction period.

continuous.discount.factor allows to adjust the discount rate used to estimated the discounted expected transactions (DERT/DECT). The default value is 0.1 (=10%). Note that a continuous rate needs to be provided.

## Value

An object of class data.table with columns:

| | |
|---|---|
| Id | The respective customer identifier |
| period.first | First timepoint of prediction period |
| period.last | Last timepoint of prediction period |
| period.length | Number of time units covered by the period indicated by period.first and period.last (including both ends). |
| PAlive | Probability to be alive at the end of the estimation period |
| CET | The Conditional Expected Transactions |
| DERT or DECT | Discounted Expected Residual Transactions or Discounted Expected Conditional Transactions for dynamic covariates models |
| actual.x | Actual number of transactions until prediction.end. Only if there is a holdout period and the prediction ends in it, otherwise it is not reported. |
| actual.total.spending | Actual total spending until prediction.end. Only if there is a holdout period and the prediction ends in it, otherwise it is not reported. |
| predicted.mean.spending | The mean spending per transactions as predicted by the spending model. |
| predicted.CLV | Customer Lifetime Value based on DERT/DECT and predicted.mean.spending. |

## See Also

models to predict transactions: pnbd, bgnbd, ggomnbd.

models to predict spending: gg.

predict for spending models

## Examples

```
data("apparelTrans")
# Fit pnbd standard model on data, WITH holdout
apparel.holdout <- clvdata(apparelTrans, time.unit="w",
                           estimation.split=37, date.format="ymd")
apparel.pnbd <- pnbd(apparel.holdout)

# Predict until the end of the holdout period
predict(apparel.pnbd)

# Predict until 10 periods (weeks in this case) after
#   the end of the 37 weeks fitting period
predict(apparel.pnbd, prediction.end = 10) # ends on 2010-11-28

# Predict until 31th Dec 2016 with the timepoint as a character
predict(apparel.pnbd, prediction.end = "2016-12-31")

# Predict until 31th Dec 2016 with the timepoint as a Date
predict(apparel.pnbd, prediction.end = lubridate::ymd("2016-12-31"))


# Predict future transactions but not spending and CLV
predict(apparel.pnbd, predict.spending = FALSE)

# Predict spending by fitting a Gamma-Gamma model
predict(apparel.pnbd, predict.spending = gg)

# Fit a spending model separately and use it to predict spending
apparel.gg <- gg(apparel.holdout, remove.first.transaction = FALSE)
predict(apparel.pnbd, predict.spending = apparel.gg)


# Fit pnbd standard model WITHOUT holdout
pnc <- pnbd(clvdata(apparelTrans, time.unit="w", date.format="ymd"))

# This fails, because without holdout, a prediction.end is required
## Not run:
predict(pnc)

## End(Not run)

# But it works if providing a prediction.end
```

```
predict(pnc, prediction.end = 10) # ends on 2016-12-17
```

---

`SetDynamicCovariates` *Add Dynamic Covariates to a CLV data object*

---

## Description

Add dynamic covariate data to an existing data object of class `clv.data`. The returned object can be used to fit models with dynamic covariates.

No covariate data can be added to a clv data object which already has any covariate set.

At least 1 covariate is needed for both processes and no categorical covariate may be of only a single category.

## Usage

```
SetDynamicCovariates(
  clv.data,
  data.cov.life,
  data.cov.trans,
  names.cov.life,
  names.cov.trans,
  name.id = "Id",
  name.date = "Date"
)
```

## Arguments

| | |
|---|---|
| `clv.data` | CLV data object to add the covariates data to. |
| `data.cov.life` | Dynamic covariate data as `data.frame` or `data.table` for the lifetime process. |
| `data.cov.trans` | Dynamic covariate data as `data.frame` or `data.table` for the transaction process. |
| `names.cov.life` | Vector with names of the columns in `data.cov.life` that contain the covariates. |
| `names.cov.trans` | Vector with names of the columns in `data.cov.trans` that contain the covariates. |
| `name.id` | Name of the column to find the Id data for both, `data.cov.life` and `data.cov.trans`. |
| `name.date` | Name of the column to find the Date data for both, `data.cov.life` and `data.cov.trans`. |

### Details

`data.cov.life` and `data.cov.trans` are `data.frames` or `data.tables` that each contain exactly 1 row for every combination of timepoint and customer. For each customer appearing in the transaction data there needs to be covariate data at every timepoint that marks the start of a period as defined by `time.unit`. It has to range from the start of the estimation sample (`timepoint.estimation.start`) until the end of the period in which the end of the holdout sample (`timepoint.holdout.end`) falls. See the the provided data `apparelDynCov` for illustration. Covariates of class `character` or `factor` are converted to k-1 numeric dummies.

`Date` as `character` If the Date column in the covariate data is of type `character`, the `date.format` given when creating the the `clv.data` object is used for parsing.

### Value

An object of class `clv.data.dynamic.covariates`. See the class definition clv.data.dynamic.covariates for more details about the returned object.

### Examples

```
## Not run:
data("apparelTrans")
data("apparelDynCov")

# Create a clv data object without covariates
clv.data.apparel <- clvdata(apparelTrans, time.unit="w",
                            date.format="ymd")

# Add static covariate data
clv.data.dyn.cov  <-
   SetDynamicCovariates(clv.data.apparel,
                        data.cov.life  = apparelDynCov,
                        names.cov.life = c("Marketing", "Gender", "Channel"),
                        data.cov.trans = apparelDynCov,
                        names.cov.trans = c("Marketing", "Gender", "Channel"),
                        name.id = "Id",
                        name.date = "Cov.Date")

# summary output about covariates data
summary(clv.data.dyn.cov)

# fit pnbd model with dynamic covariates
pnbd(clv.data.dyn.cov)

## End(Not run)
```

---

## SetStaticCovariates *Add Static Covariates to a CLV data object*

---

## Description

Add static covariate data to an existing data object of class `clv.data`. The returned object then can be used to fit models with static covariates.

No covariate data can be added to a clv data object which already has any covariate set.

At least 1 covariate is needed for both processes and no categorical covariate may be of only a single category.

## Usage

```
SetStaticCovariates(
  clv.data,
  data.cov.life,
  data.cov.trans,
  names.cov.life,
  names.cov.trans,
  name.id = "Id"
)
```

## Arguments

| | |
|---|---|
| `clv.data` | CLV data object to add the covariates data to. |
| `data.cov.life` | Static covariate data as `data.frame` or `data.table` for the lifetime process. |
| `data.cov.trans` | Static covariate data as `data.frame` or `data.table` for the transaction process. |
| `names.cov.life` | Vector with names of the columns in `data.cov.life` that contain the covariates. |
| `names.cov.trans` | Vector with names of the columns in `data.cov.trans` that contain the covariates. |
| `name.id` | Name of the column to find the Id data for both, `data.cov.life` and `data.cov.trans`. |

## Details

`data.cov.life` and `data.cov.trans` are `data.frames` or `data.tables` that each contain exactly one single row of covariate data for every customer appearing in the transaction data. Covariates of class `character` or `factor` are converted to k-1 numeric dummy variables.

## Value

An object of class `clv.data.static.covariates`. See the class definition clv.data.static.covariates for more details about the returned object.

## Examples

```
data("apparelTrans")
data("apparelStaticCov")
```

```
# Create a clv data object without covariates
clv.data.apparel <- clvdata(apparelTrans, time.unit="w",
                            date.format="ymd")

# Add static covariate data
clv.data.apparel.cov  <-
   SetStaticCovariates(clv.data.apparel,
                       data.cov.life  = apparelStaticCov,
                       names.cov.life = "Gender",
                       data.cov.trans = apparelStaticCov,
                       names.cov.trans = "Gender",
                       name.id = "Id")

# more summary output
summary(clv.data.apparel.cov)

# fit model with static covariates
pnbd(clv.data.apparel.cov)
```

---

`summary.clv.fitted` *Summarizing a fitted CLV model*

---

## Description

Summary method for fitted CLV models that provides statistics about the estimated parameters and information about the optimization process. If multiple optimization methods were used (for example if specified in parameter `optimx.args`), all information here refers to the last method/row of the resulting `optimx` object.

## Usage

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

## S3 method for class 'clv.fitted.transactions.static.cov'
summary(object, ...)

## S3 method for class 'summary.clv.fitted'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

## Arguments

| | |
|---|---|
| object | A fitted CLV model |
| ... | Ignored for summary, forwarded to printCoefmat for print. |
| x | an object of class "summary.clv.no.covariates", usually, a result of a call to summary.clv.no.covariates. |
| digits | the number of significant digits to use when printing. |
| signif.stars | logical. If TRUE, 'significance stars' are printed for each coefficient. |

## Value

This function computes and returns a list of summary information of the fitted model given in object. It returns a list of class summary.clv.no.covariates that contains the following components:

name.model — the name of the fitted model.

call — The call used to fit the model.

tp.estimation.start — Date or POSIXct indicating when the fitting period started.

tp.estimation.end — Date or POSIXct indicating when the fitting period ended.

estimation.period.in.tu — Length of fitting period in time.units.

time.unit — Time unit that defines a single period.

coefficients — a px4 matrix with columns for the estimated coefficients, its standard error, the t-statistic and corresponding (two-sided) p-value.

estimated.LL — the value of the log-likelihood function at the found solution.

AIC — Akaike's An Information Criterion for the fitted model.

BIC — Schwarz's Bayesian Information Criterion for the fitted model.

KKT1 — Karush-Kuhn-Tucker optimality conditions of the first order, as returned by optimx.

KKT2 — Karush-Kuhn-Tucker optimality conditions of the second order, as returned by optimx.

fevals — The number of calls to the log-likelihood function during optimization.

method — The last method used to obtain the final solution.

additional.options — A list of additional options used for model fitting.

- **Correlation** Whether the correlation between the purchase and the attrition process was estimated.
- **estimated.param.cor** Correlation coefficient measuring the correlation between the two processes, if used.

For models fits with static covariates, the list additionally is of class summary.clv.static.covariates and the list in additional.options contains the following elements:

additional.options

**Regularization** Whether L2 regularization for parameters of contextual factors was used.

**lambda.life** The regularization lambda used for the parameters of the Lifetime process, if used.

**lambda.trans** The regularization lambda used for the parameters of the Transaction process, if used.

**Constraint covs** Whether any covariate parameters were forced to be the same for both processes.

**Constraint params** Name of the covariate parameters which were constraint, if used.

## See Also

The model fitting functions pnbd.

Function coef will extract the coefficients matrix including summary statistics and function vcov will extract the vcov from the returned summary object.

## Examples

```
data("apparelTrans")

# Fit pnbd standard model, no covariates
clv.data.apparel <- clvdata(apparelTrans, time.unit="w",
                            estimation.split=40, date.format="ymd")
pnbd.apparel <- pnbd(clv.data.apparel)

# summary about model fit
summary(pnbd.apparel)

# Add static covariate data
data("apparelStaticCov")
data.apparel.cov  <-
   SetStaticCovariates(clv.data.apparel,
                       data.cov.life  = apparelStaticCov,
                       names.cov.life = "Gender",
                       data.cov.trans = apparelStaticCov,
                       names.cov.trans = "Gender",
                       name.id = "Id")

# fit model with covariates and regualization
pnbd.apparel.cov <- pnbd(data.apparel.cov,
                         reg.lambdas = c(life=2, trans=4))

# additional summary about covariate parameters
#   and used regularization
summary(pnbd.apparel.cov)
```

`vcov.clv.fitted` *Calculate Variance-Covariance Matrix for CLV Models fitted with Maximum Likelihood Estimation*

## Description

Returns the variance-covariance matrix of the parameters of the fitted model object. The variance-covariance matrix is derived from the Hessian that results from the optimization procedure. First, the Moore-Penrose generalized inverse of the Hessian is used to obtain an estimate of the variance-covariance matrix. Next, because some parameters may be transformed for the purpose of restricting their value during the log-likelihood estimation, the variance estimates are adapted to be comparable to the reported coefficient estimates. If the result is not positive definite, Matrix::nearPD is used with standard settings to find the nearest positive definite matrix.

If multiple estimation methods were used, the Hessian of the last method is used.

## Usage

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

## Arguments

| | |
|---|---|
| `object` | a fitted clv model object |
| `...` | Ignored |

## Value

A matrix of the estimated covariances between the parameters of the model. The row and column names correspond to the parameter names given by the `coef` method.

## See Also

MASS::ginv, Matrix::nearPD

# Index