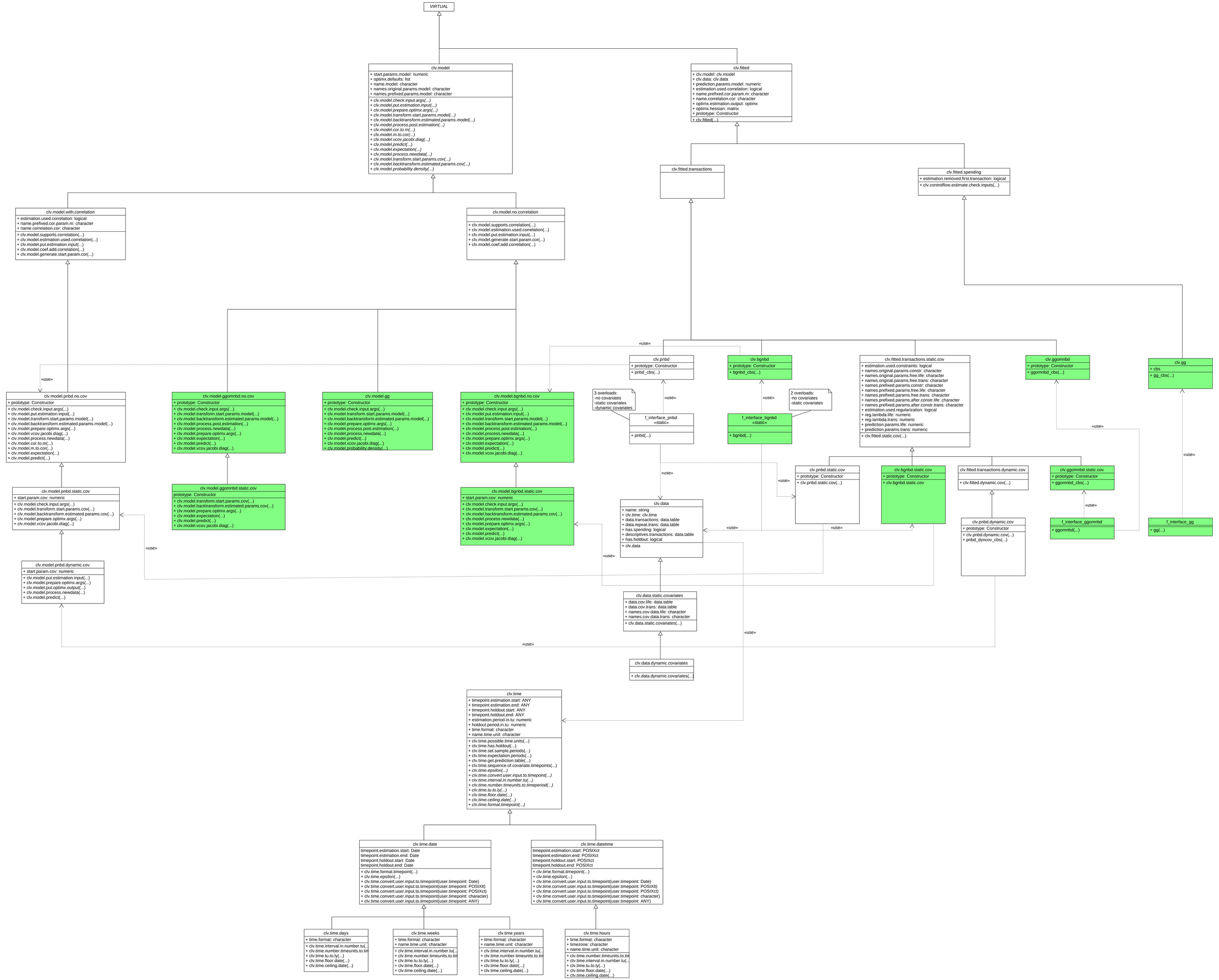
VIRTUAL
clv.model
+ start.params.model: numeric
+ optimx.defaults: list
+ name.model: character
+ names.original.params.model: character
+ names.prefixed.params.model: character
+ clv.model.check.input.args(...)
+ clv.model.put.estimation.input(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.transform.start.params.model(...)
+ clv.model.backtransform.estimated.params.model(...)
+ clv.model.process.post.estimation(...)
+ clv.model.cor.to.m(...)
+ clv.model.m.to.cor(...)
+ clv.model.vcov.jacobi.diag(...)
+ clv.model.predict(...)
+ clv.model.expectation(...)
+ clv.model.process.newdata(...)
+ clv.model.transform.start.params.cov(...)
+ clv.model.backtransform.estimated.params.cov(...)
+ clv.model.probability.density(...)
clv.fitted
+ clv.model: clv.model
+ clv.data: clv.data
+ prediction.params.model: numeric
+ estimation.used.correlation: logical
+ name.prefixed.cor.param.m: character
+ name.correlation.cor: character
+ optimx.estimation.output: optimx
+ optimx.hessian: matrix
+ prototype: Constructor
+ clv.fitted(...)
clv.model.with.correlation
+ estimation.used.correlation: logical
+ name.prefixed.cor.param.m: character
+ name.correlation.cor: character
+ clv.model.supports.correlation(...)
+ clv.model.estimation.used.correlation(...)
+ clv.model.put.estimation.input(...)
+ clv.model.coef.add.correlation(...)
+ clv.model.generate.start.param.cor(...)
clv.model.no.correlation
+ clv.model.supports.correlation(...)
+ clv.model.estimation.used.correlation(...)
+ clv.model.put.estimation.input(...)
+ clv.model.generate.start.param.cor(...)
+ clv.model.coef.add.correlation(...)
clv.fitted.transactions
clv.fitted.spending
+ estimation.removed.first.transaction: logical
+ clv.controlflow.estimate.check.inputs(...)
«use»
clv.pnbd
+ prototype: Constructor
+ pnbd_cbs(...)
clv.bgnbd
+ prototype: Constructor
+ bgnbd_cbs(...)
clv.fitted.transactions.static.cov
+ estimation.used.constraints: logical
+ names.original.params.constr: character
+ names.original.params.free.life: character
+ names.original.params.free.trans: character
+ names.prefixed.params.constr: character
+ names.prefixed.params.free.life: character
+ names.prefixed.params.free.trans: character
+ names.prefixed.params.after.constr.life: character
+ names.prefixed.params.after.constr.trans: character
+ estimation.used.regularization: logical
+ reg.lambda.life: numeric
+ reg.lambda.trans: numeric
+ prediction.params.life: numeric
+ prediction.params.trans: numeric
+ clv.fitted.static.cov(...)
clv.ggomnbd
+ prototype: Constructor
+ ggomnbd_cbs(...)
clv.gg
+ cbs
+ gg_cbs(...)
clv.model.pnbd.no.cov
+ prototype: Constructor
+ clv.model.check.input.args(...)
+ clv.model.put.estimation.input(...)
+ clv.model.transform.start.params.model(...)
+ clv.model.backtransform.estimated.params.model(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.vcov.jacobi.diag(...)
+ clv.model.process.newdata(...)
+ clv.model.expectation(...)
+ clv.model.cor.to.m(...)
+ clv.model.m.to.cor(...)
+ clv.model.expectation(...)
+ clv.model.predict(...)
clv.model.ggomnbd.no.cov
+ prototype: Constructor
+ clv.model.check.input.args(...)
+ clv.model.transform.start.params.model(...)
+ clv.model.backtransform.estimated.params.model(...)
+ clv.model.process.post.estimation(...)
+ clv.model.process.newdata(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.expectation(...)
+ clv.model.predict(...)
+ clv.model.vcov.jacobi.diag(...)
clv.model.gg
+ prototype: Constructor
+ clv.model.check.input.args(...)
+ clv.model.transform.start.params.model(...)
+ clv.model.backtransform.estimated.params.model(...)
+ clv.model.process.post.estimation(...)
+ clv.model.process.newdata(...)
+ clv.model.predict(...)
+ clv.model.vcov.jacobi.diag(...)
+ clv.model.probability.density(...)
clv.model.bgnbd.no.cov
+ prototype: Constructor
+ clv.model.check.input.args(...)
+ clv.model.put.estimation.input(...)
+ clv.model.transform.start.params.model(...)
+ clv.model.backtransform.estimated.params.model(...)
+ clv.model.process.post.estimation(...)
+ clv.model.process.newdata(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.expectation(...)
+ clv.model.predict(...)
+ clv.model.vcov.jacobi.diag(...)
3 overloads:
-no covariates
-static covariates
-dynamic covariates
f_interface_pnbd
«static»
+ pnbd(...)
2 overloads:
-no covariates
-static covariates
f_interface_bgnbd
«static»
+ bgnbd(...)
clv.pnbd.static.cov
+ prototype: Constructor
+ clv.pnbd.static.cov(...)
clv.bgnbd.static.cov
+ prototype: Constructor
+ clv.bgnbd.static.cov
clv.fitted.transactions.dynamic.cov
+ clv.fitted.dynamic.cov(...)
clv.ggomnbd.static.cov
+ prototype: Constructor
+ ggomnbd_cbs(...)
clv.pnbd.dynamic.cov
+ prototype: Constructor
+ clv.pnbd.dynamic.cov(...)
+ pnbd_dyncov_cbs(...)
f_interface_ggomnbd
+ ggomnbd(...)
f_interface_gg
+ gg(...)
clv.model.pnbd.static.cov
+ start.param.cov: numeric
+ clv.model.check.input.args(...)
+ clv.model.transform.start.params.cov(...)
+ clv.model.backtransform.estimated.params.cov(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.vcov.jacobi.diag(...)
clv.model.ggomnbd.static.cov
prototype: Constructor
+ clv.model.transform.start.params.cov(...)
+ clv.model.backtransform.estimated.params.cov(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.expectation(...)
+ clv.model.predict(...)
+ clv.model.vcov.jacobi.diag(...)
clv.model.bgnbd.static.cov
+ start.param.cov: numeric
+ clv.model.check.input.args(...)
+ clv.model.transform.start.params.cov(...)
+ clv.model.backtransform.estimated.params.cov(...)
+ clv.model.process.newdata(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.expectation(...)
+ clv.model.predict(...)
+ clv.model.vcov.jacobi.diag(...)
clv.data
+ name: string
+ clv.time: clv.time
+ data.transactions: data.table
+ data.repeat.trans: data.table
+ has.spending: logical
+ descriptives.transactions: data.table
+ has.holdout: logical
+ clv.data
clv.model.pnbd.dynamic.cov
+ start.param.cov: numeric
+ clv.model.put.estimation.input(...)
+ clv.model.prepare.optimx.args(...)
+ clv.model.put.optimx.output(...)
+ clv.model.process.newdata(...)
+ clv.model.predict(...)
clv.data.static.covariates
+ data.cov.life: data.table
+ data.cov.trans: data.table
+ names.cov.data.life: character
+ names.cov.data.trans: character
+ clv.data.static.covariates(...)
clv.data.dynamic.covariates
+ clv.data.dynamic.covariates(...)
clv.time
+ timepoint.estimation.start: ANY
+ timepoint.estimation.end: ANY
+ timepoint.holdout.start: ANY
+ timepoint.holdout.end: ANY
+ estimation.period.in.tu: numeric
+ holdout.period.in.tu: numeric
+ time.format: character
+ name.time.unit: character
+ clv.time.possible.time.units(...)
+ clv.time.has.holdout(...)
+ clv.time.set.sample.periods(...)
+ clv.time.expectation.periods(...)
+ clv.time.get.prediction.table(...)
+ clv.time.sequence.of.covariate.timepoints(...)
+ clv.time.epsilon(...)
+ clv.time.convert.user.input.to.timepoint(...)
+ clv.time.interval.in.number.tu(...)
+ clv.time.number.timeunits.to.timeperiod(...)
+ clv.time.tu.to.ly(...)
+ clv.time.floor.date(...)
+ clv.time.ceiling.date(...)
+ clv.time.format.timepoint(...)
clv.time.date
timepoint.estimation.start: Date
timepoint.estimation.end: Date
timepoint.holdout.start: Date
timepoint.holdout.end: Date
+ clv.time.format.timepoint(...)
+ clv.time.epsilon(...)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: Date)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: POSIXlt)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: POSIXct)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: character)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: ANY)
clv.time.datetime
timepoint.estimation.start: POSIXct
timepoint.estimation.end: POSIXct
timepoint.holdout.start: POSIXct
timepoint.holdout.end: POSIXct
+ clv.time.format.timepoint(...)
+ clv.time.epsilon(...)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: Date)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: POSIXlt)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: POSIXct)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: character)
+ clv.time.convert.user.input.to.timepoint(user.timepoint: ANY)
clv.time.days
+ time.format: character
+ clv.time.interval.in.number.tu(...
+ clv.time.number.timeunits.to.tim
+ clv.time.tu.to.ly(...)
+ clv.time.floor.date(...)
+ clv.time.ceiling.date(...)
clv.time.weeks
+ time.format: character
+ name.time.unit: character
+ clv.time.interval.in.number.tu(...
+ clv.time.number.timeunits.to.tim
+ clv.time.tu.to.ly(...)
+ clv.time.floor.date(...)
+ clv.time.ceiling.date(...)
clv.time.years
+ time.format: character
+ name.time.unit: character
+ clv.time.interval.in.number.tu(...
+ clv.time.number.timeunits.to.tim
+ clv.time.tu.to.ly(...)
+ clv.time.floor.date(...)
+ clv.time.ceiling.date(...)
clv.time.hours
+ time.format: character
+ timezone: character
+ name.time.unit: character
+ clv.time.number.timeunits.to.tim
+ clv.time.interval.in.number.tu(...
+ clv.time.tu.to.ly(...)
+ clv.time.floor.date(...)
+ clv.time.ceiling.date(...)