

ISoP StudyGroup mrgsolve demo

Kyle Baron

13 May 2016

Load: mrgsolve, modmrg, magrittr, dplyr

```
library(mrgsolve) #
library(modmrg)   #
library(magrittr) #
library(dplyr)    #
```

Source: functions.R, data.R

```
source("functions.R") #
make <- function() make_worksheet("demo.R")
```

Load a model from modmrg 1-compartment PK model

```
mod <- pk1cmt()
```

Basics:

- show
- param/init
- Update
- Simulate / plot
- Check class

```
mod

##
##
## ----- mrgsolve model object (unix) -----
## Project: /Users/kyleb/Rlibs/lib/modmrg/project
## source:      pk1cmt.cpp
## shared object: modmrg (loaded)
##
## compile date:
## Time:        start: 0 end: 24 delta: 1
## >            add: <none>
## >            tscale: 1
##
## Compartments: EV1 CENT EV2 [3]
## Parameters:   CL VC KA1 KA2 VMAX KM [6]
## Omega:        0x0
## Sigma:        0x0
##
## Solver:       atol: 1e-08 rtol: 1e-08
## >             maxsteps: 2000 hmin: 0 hmax: 0
```

```
param(mod)
```

```
##
## Model parameters (N=6):
## name value . name value
## CL 1 | KM 2
## KA1 1 | VC 10
## KA2 1 | VMAX 0
```

```
init(mod)
```

```
##
## Model initial conditions (N=3):
## name value . name value
## CENT (2) 0 | EV2 (3) 0
## EV1 (1) 0 | . ... .
```

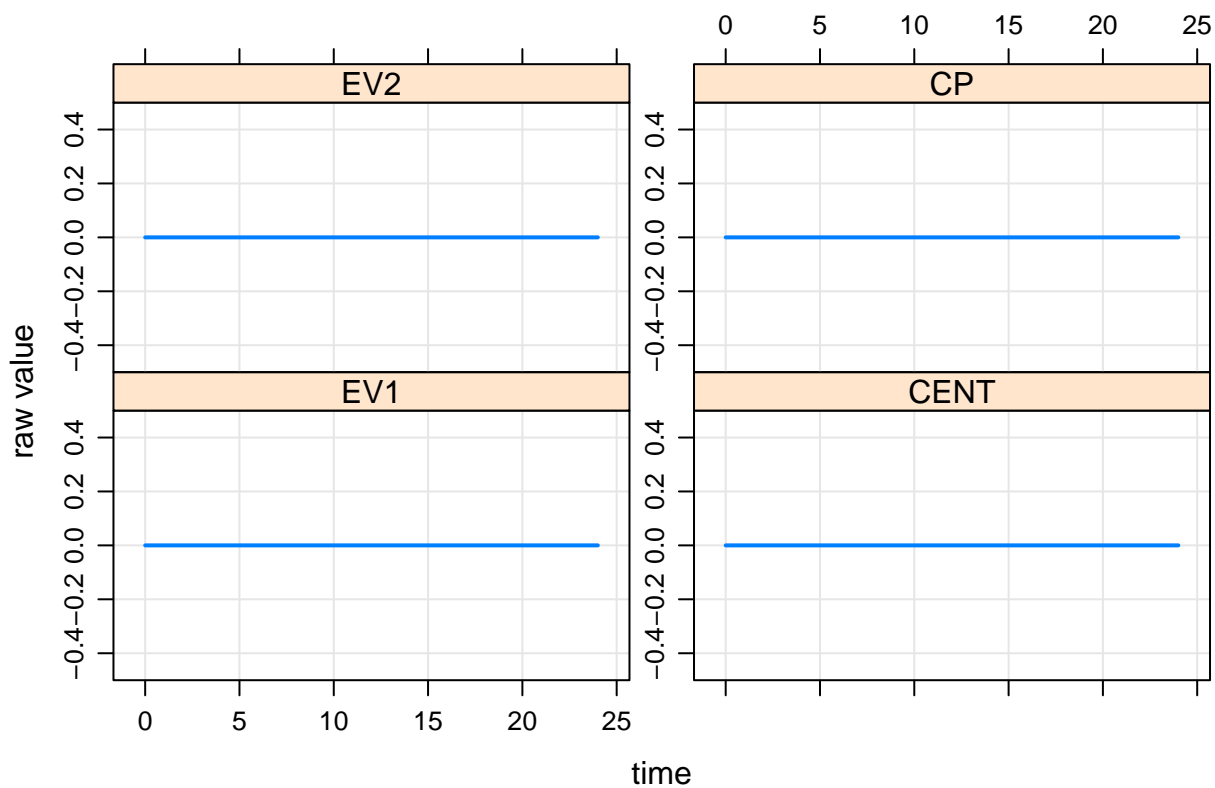
```
mod %>% mrgsim
```

```
## Model: pk1cmt.cpp
## Dim: 25 x 6
## Time: 0 to 24
## ID: 1
## ID time EV1 CENT EV2 CP
## [1,] 1 0 0 0 0 0
## [2,] 1 1 0 0 0 0
## [3,] 1 2 0 0 0 0
## [4,] 1 3 0 0 0 0
## [5,] 1 4 0 0 0 0
## [6,] 1 5 0 0 0 0
## [7,] 1 6 0 0 0 0
## [8,] 1 7 0 0 0 0
```

```
mod %>% mrgsim %>% class
```

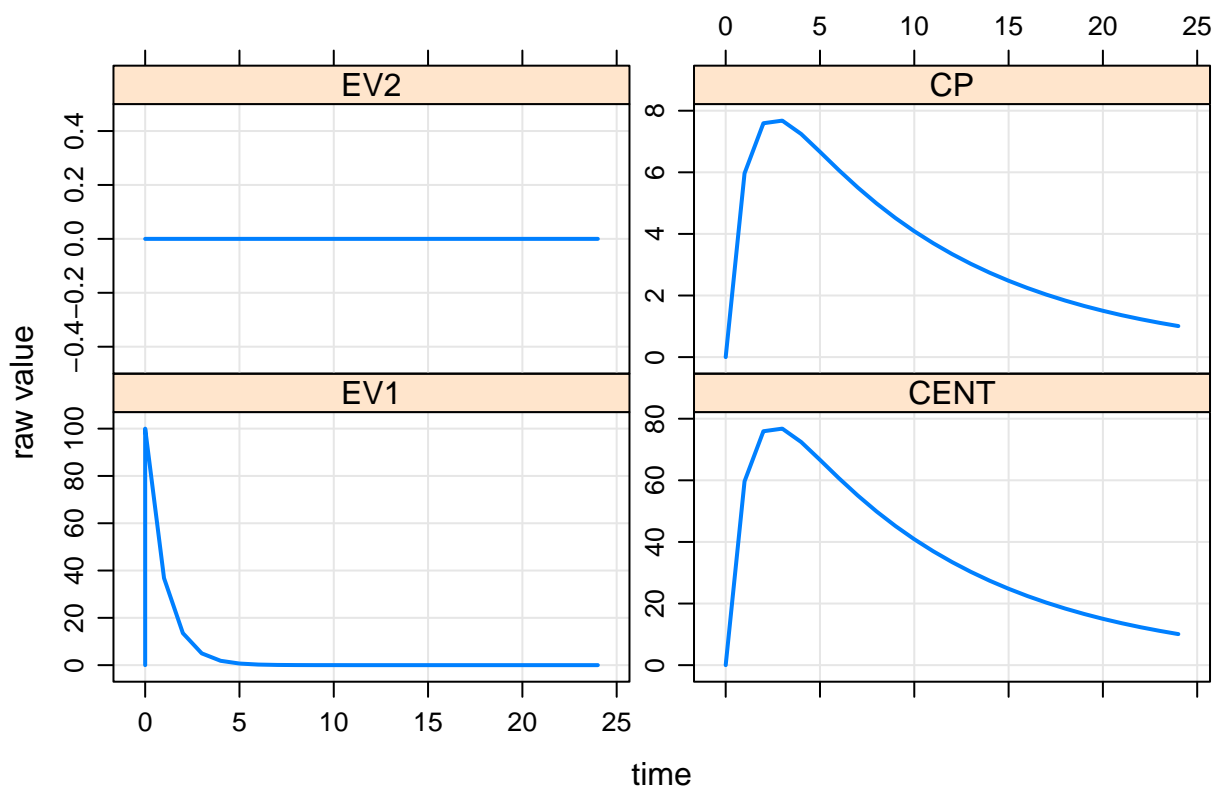
```
## [1] "mrgsims"
## attr(,"package")
## [1] "mrgsolve"
```

```
mod %>% mrgsim %>% plot
```



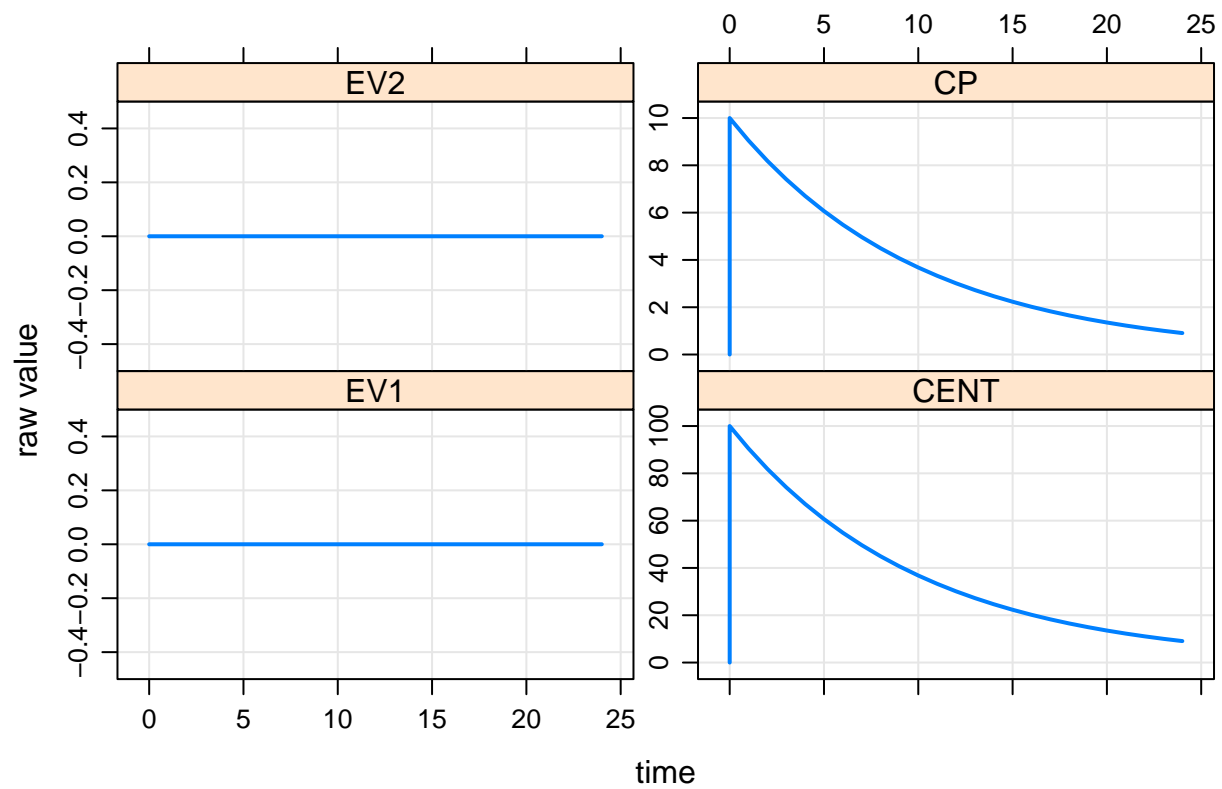
Add dosing event: 100 mg PO x1

```
mod %>% ev(amt=100) %>% mrgsim %>% plot
```

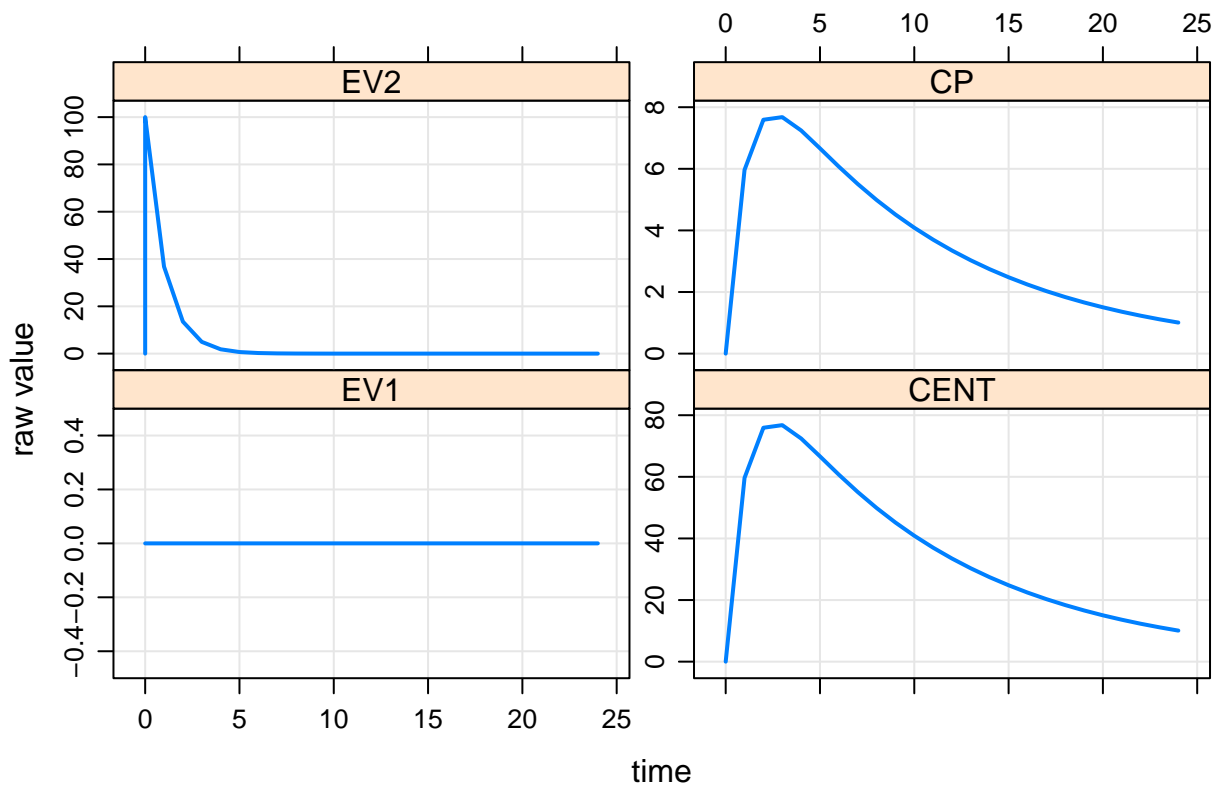


Dose to cmt=2, dose to “EV2”

```
mod %>% ev(amt=100, cmt=2) %>% mrgsim %>% plot
```



```
mod %>% ev(amt=100, cmt="EV2") %>% mrgsim %>% plot
```



Items you can have in `ev`

- time
- amt
- rate
- cmt
- evid
- ii / addl
- ss

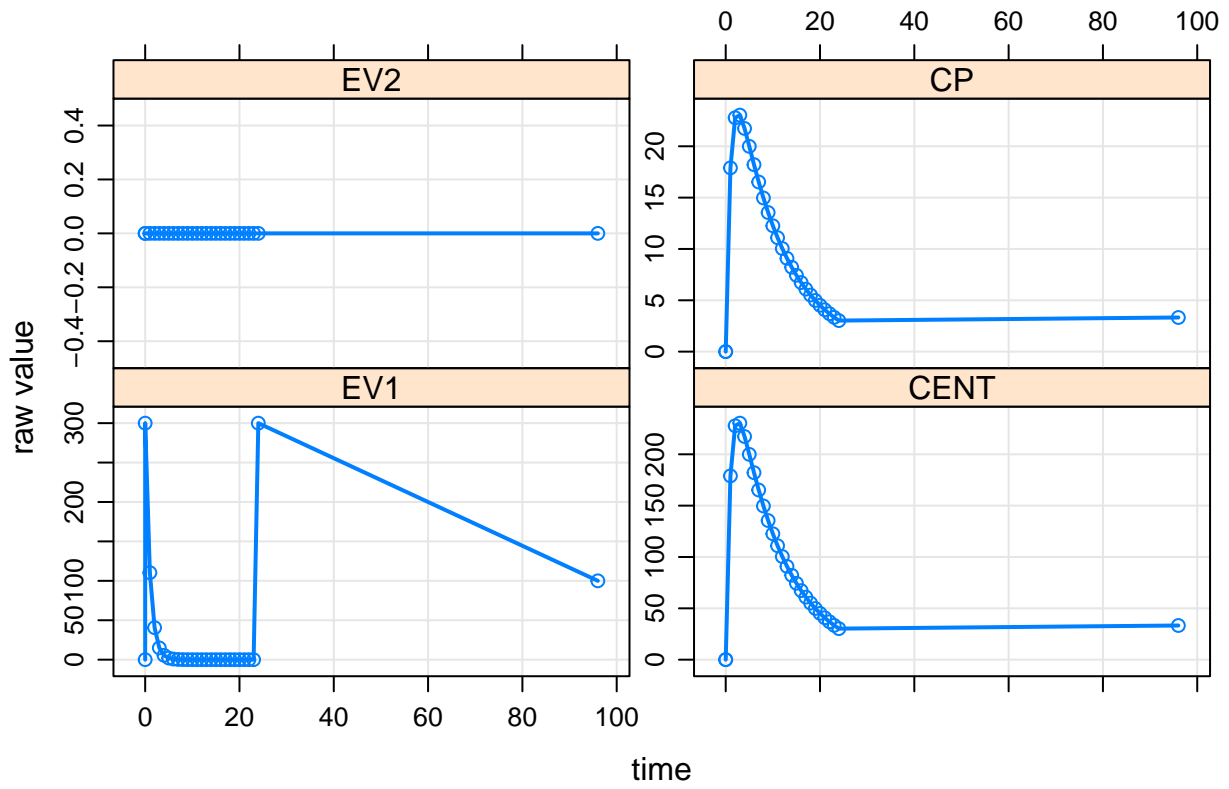
Events `ev`

- 300 mg PO Q24H x 4, then 100 mg Q8H x14

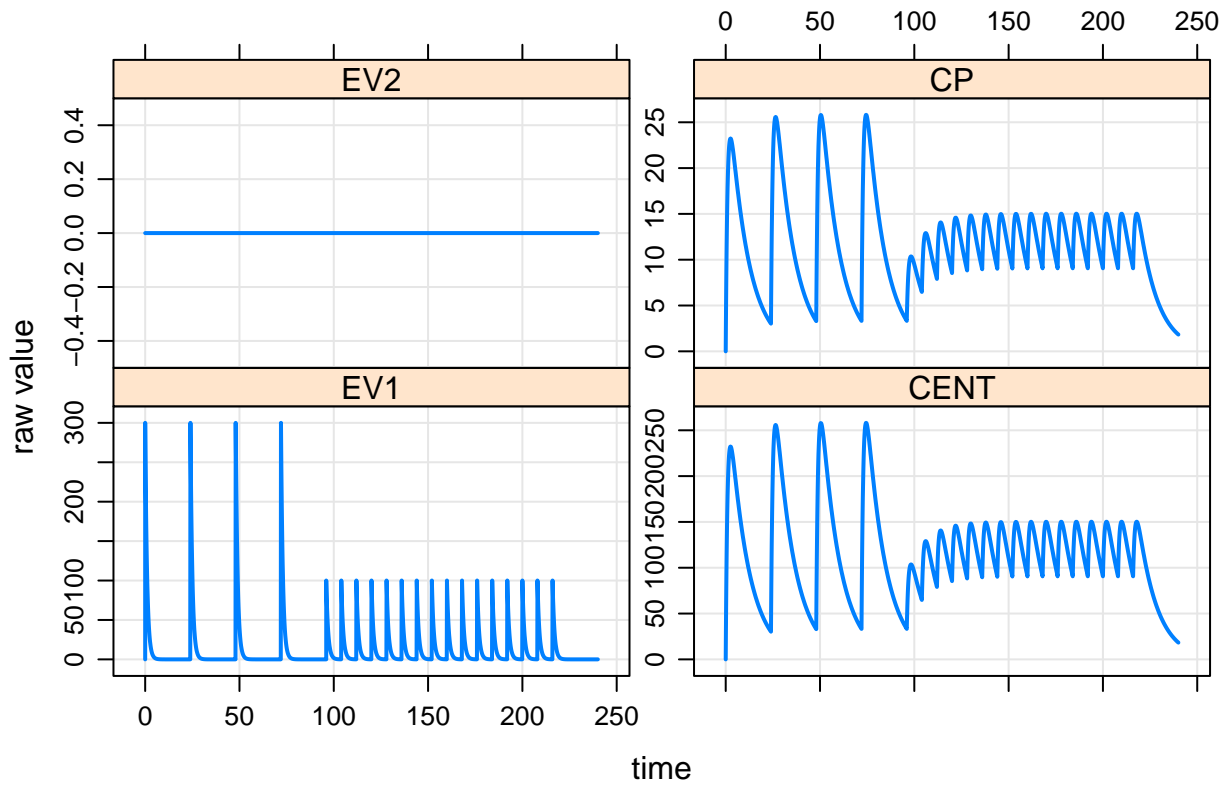
```
e1 <- ev(amt=300, ii=24, addl=3)
e2 <- ev(amt=100, ii=8, addl=15)
e <- e1 %then% e2
```

Simulate from `e`, plot/both

```
mod %>% ev(e) %>% mrgsim %>% plot(type='b')
```



```
mod %>% ev(e) %>% mrgsim(end=240,delta=0.1) %>% plot
```

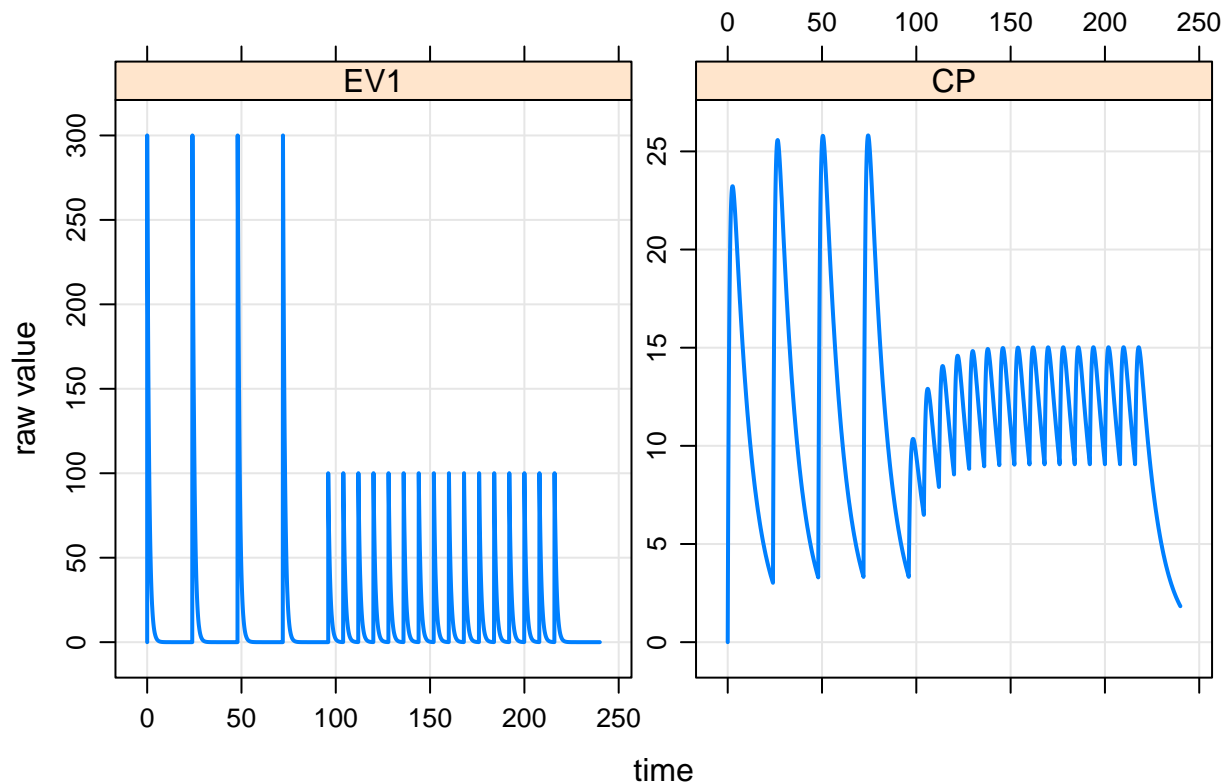


Persistent update: end/delta

```
mod %<>% update(end=240,delta=0.1)
```

Request certain outputs

```
mod %>% Req(EV1,CP) %>% ev(e) %>% mrgsim %>% plot
```



Request CP, end -> 96

```
mod %<>% Req(CP) %>% update(end=96)
```

data_set

data:

- 100/300/1000 over 10H Q24H x3

```
data <- expand.ev(amt=c(100,300,1000),
                 ii=24, addl=2, cmt=2)
```

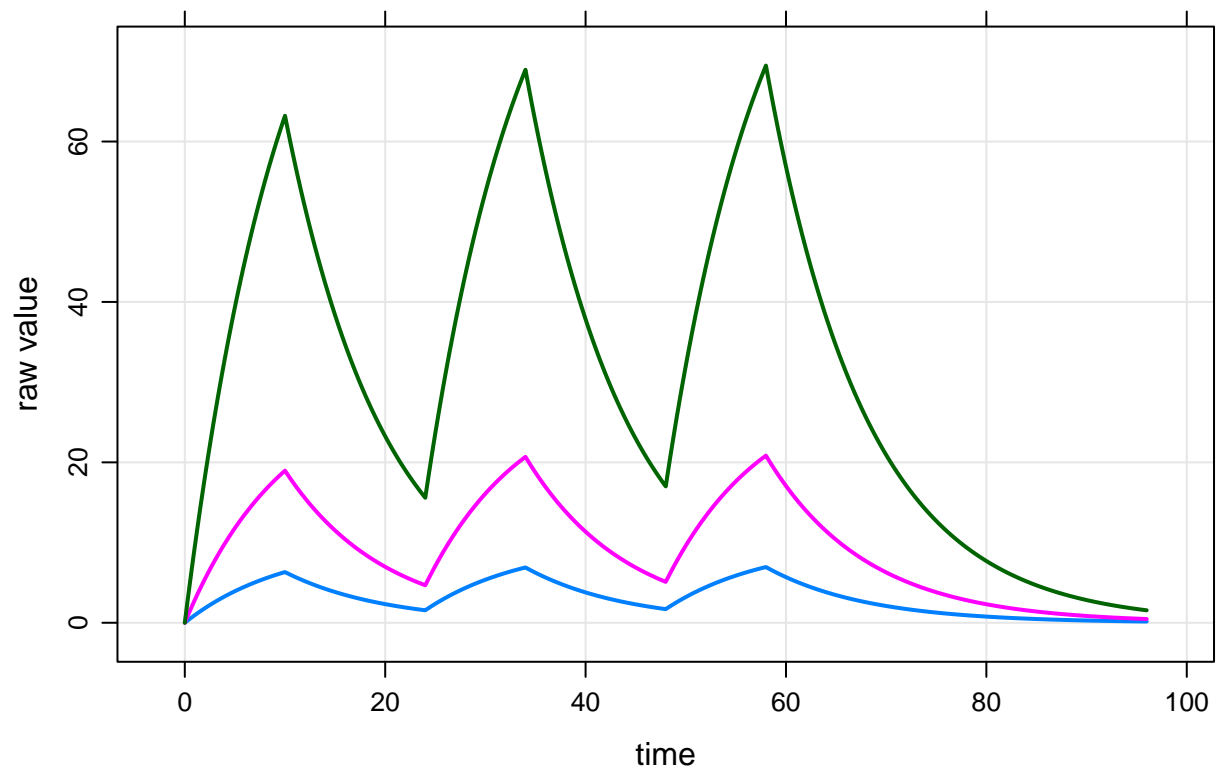
```
data %<>% mutate(rate = amt/10)
```

data

```
##   ID  amt ii addl cmt evid time rate
## 1  1  100 24   2   2    1    0   10
## 2  2  300 24   2   2    1    0   30
## 3  3 1000 24   2   2    1    0  100
```

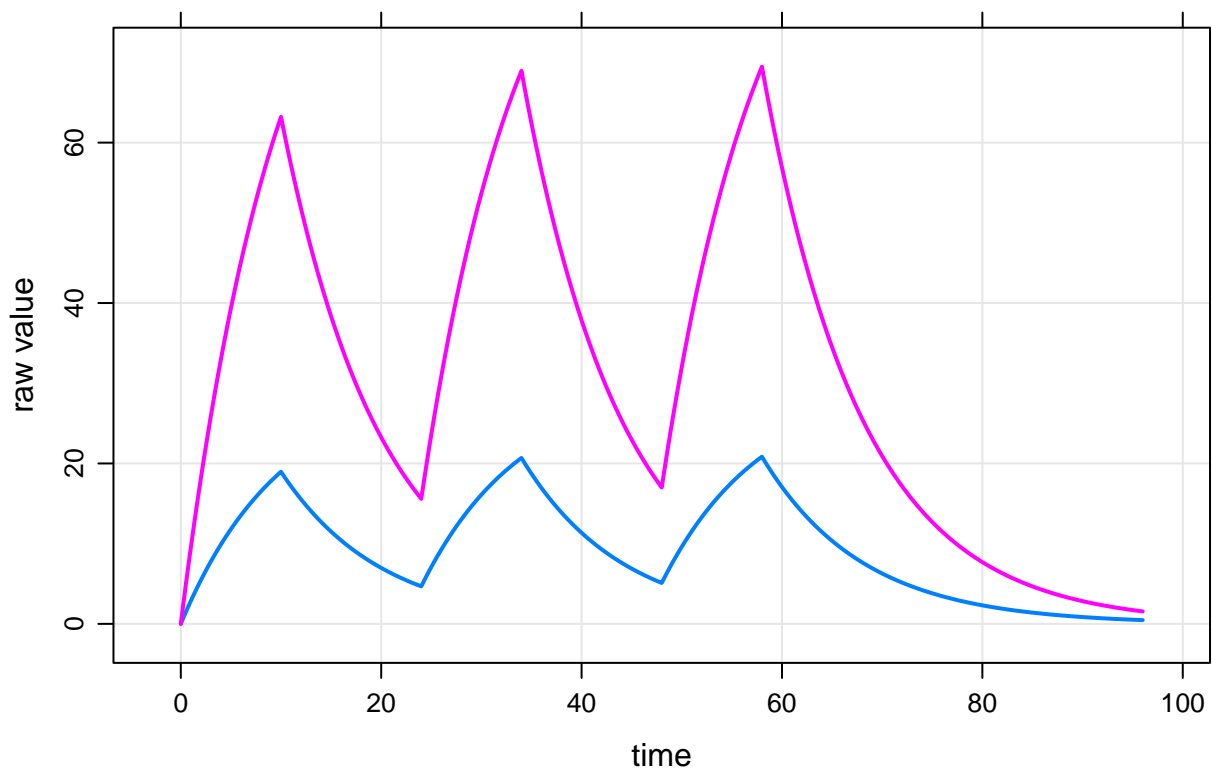
Simulate

```
mod %>% data_set(data) %>% mrgsim %>% plot
```



Filter and simulate

```
mod %>% data_set(data, ID > 1) %>% mrgsim %>% plot
```

data: extran1

plot: CP~time|ID

```
data(extran1)
```

```
extran1
```

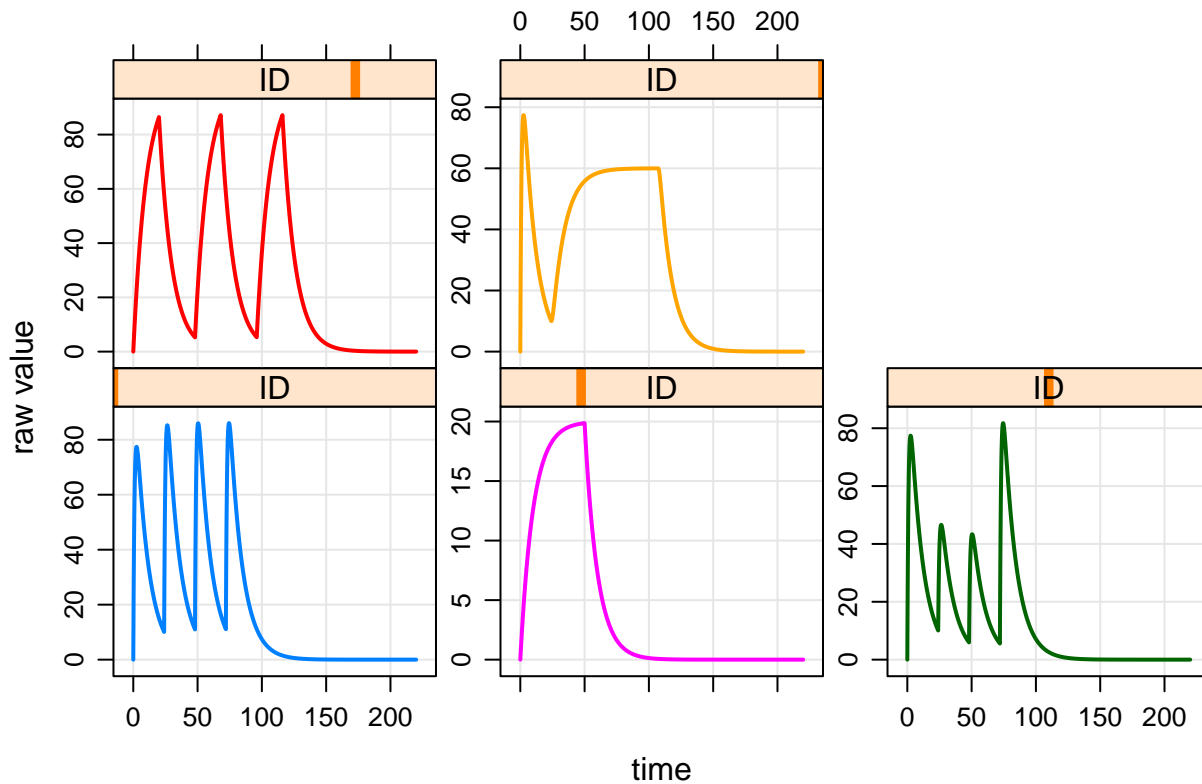
```
##   ID  amt cmt time addl ii rate evid
## 1  1 1000  1   0   3 24   0   1
## 2  2 1000  2   0   0 0  20   1
## 3  3 1000  1   0   0 0   0   1
## 4  3  500  1  24   0 0   0   1
## 5  3  500  1  48   0 0   0   1
## 6  3 1000  1  72   0 0   0   1
## 7  4 2000  2   0   2 48  100  1
## 8  5 1000  1   0   0 0   0   1
## 9  5 5000  1  24   0 0   60  1
```

```
mod %>%
```

```
  data_set(extran1) %>%
```

```
  mrgsim(end=220) %>%
```

```
  plot(CP~time|ID)
```



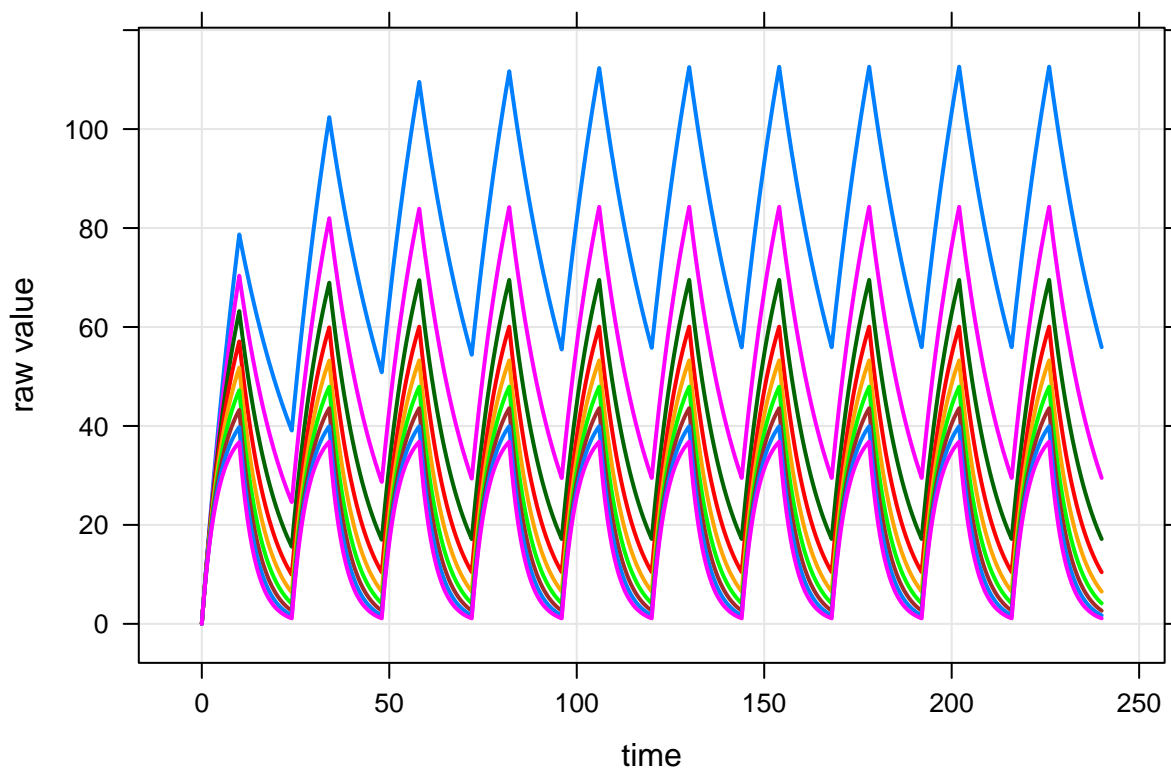
data:

- 1000 mg doses IV over 10H
- CL ~ 0.5 → 2.5

```
data <- expand.ev(amt=1000, ii=24, addl=100,
  rate = 100, cmt=2,
  CL = seq(0.5,2.5,0.25))
```

Simulate

```
mod %>%
  data_set(data) %>%
  mrgsim(end=240) %>%
  plot(CP~time,scales="same")
```



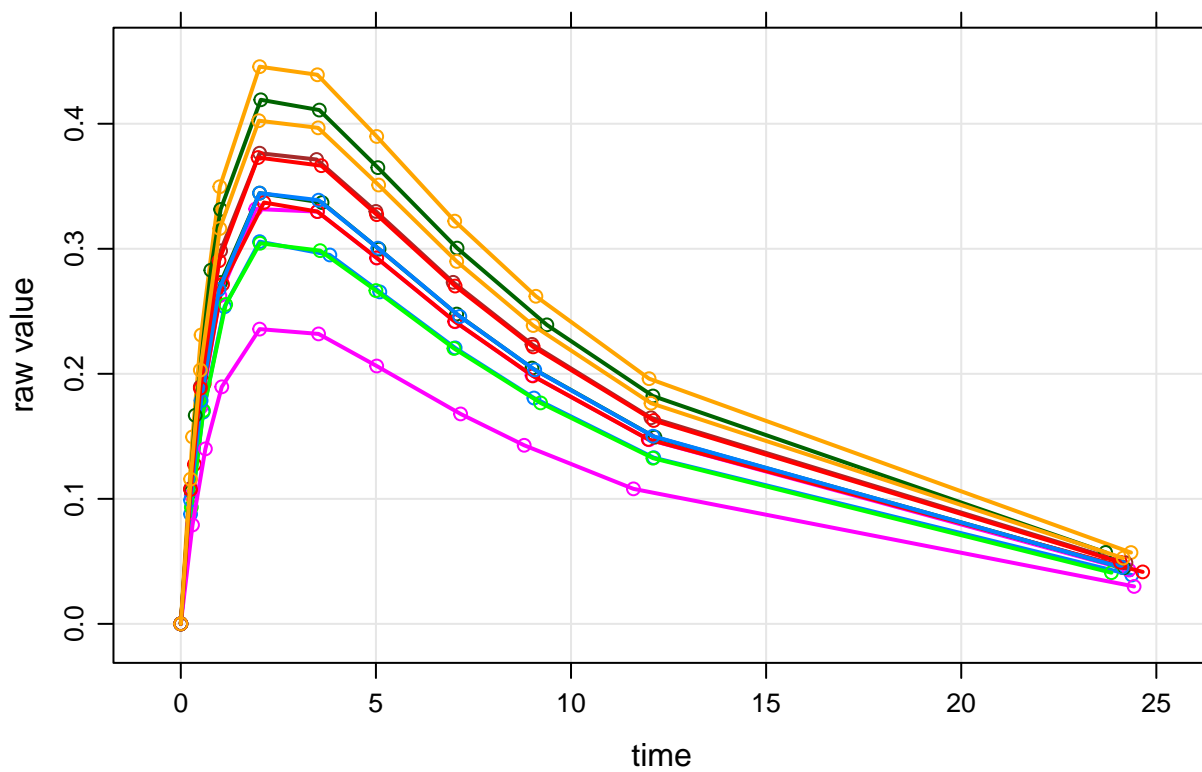
- data:
- exTheoph

```
data(exTheoph)
head(exTheoph)
```

```
##   ID   WT Dose time  conc  cmt  amt evid
## 1   1 79.6 4.02 0.00  0.00   1 4.02    1
## 2   1 79.6 4.02 0.25  2.84   0 0.00    0
## 3   1 79.6 4.02 0.57  6.57   0 0.00    0
## 4   1 79.6 4.02 1.12 10.50   0 0.00    0
## 5   1 79.6 4.02 2.02  9.66   0 0.00    0
## 6   1 79.6 4.02 3.82  8.58   0 0.00    0
```

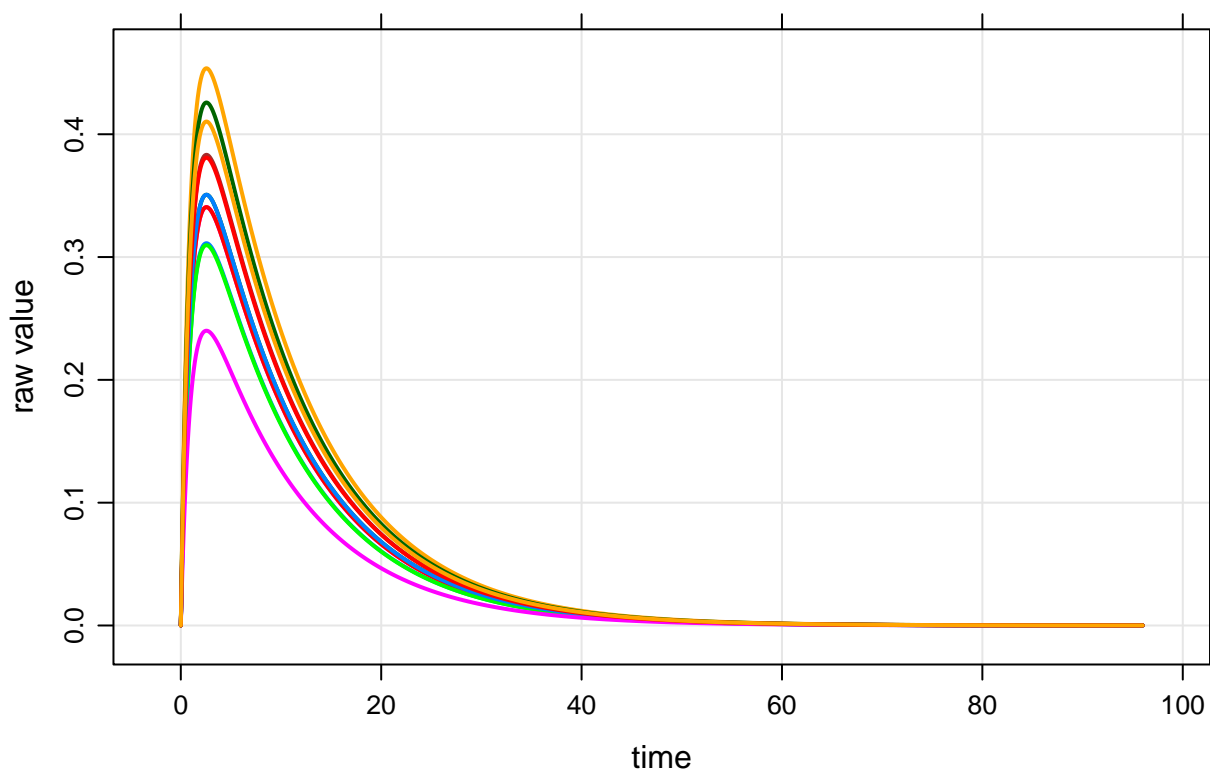
Simulate from exTheoph

```
mod %>% data_set(exTheoph) %>% mrgsim(delta=0.1) %>% plot(type='b')
```



Filter doses, then simulate

```
mod %>% data_set(exTheoph,subset=evid==1) %>% mrgsim(delta=0.1) %>% plot
```



Switch back to demo.R?

Model specification

- Parameters and compartments
- Set initial conditions
- Covariates and random effects
- Bioavailability / Lag time / Infusion D/R
- ODEs
- Output variables
- 1-cmt model, first-order absorption
- Parameters: TVCL, TVVC, KA, WT, WTCL
- Compartments: GUT CENT
- Covariate model: CL~WT, VC~WT
- Output: CP = CENT/V, KA

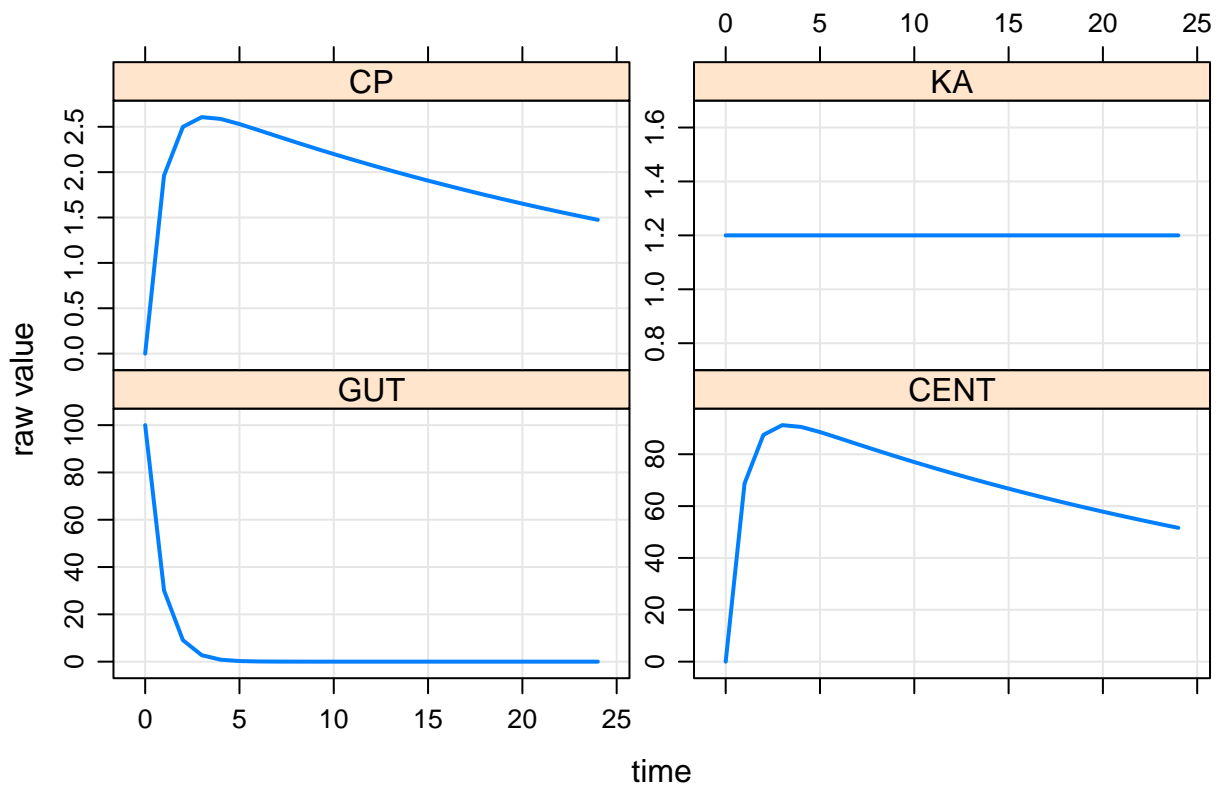
```
code <- '  
$PARAM TVCL = 1, TVVC = 35, KA = 1.2  
WT = 70, WTCL = 0.75  
  
$CMT GUT CENT  
  
$MAIN  
double CL = TVCL*pow(WT/70,WTCL);  
double V = TVVC*pow(WT/70,1);  
  
$ODE  
dxdt_GUT = -KA*GUT;  
dxdt_CENT = KA*GUT - (CL/V)*CENT;  
  
$TABLE  
table(CP) = CENT/V;  
  
$CAPTURE KA  
'
```

Parse, compile and load

```
mod <- mcode("spec", code)
```

Simulate / init()

```
mod %>% init(GUT=100) %>% mrgsim %>% plot
```



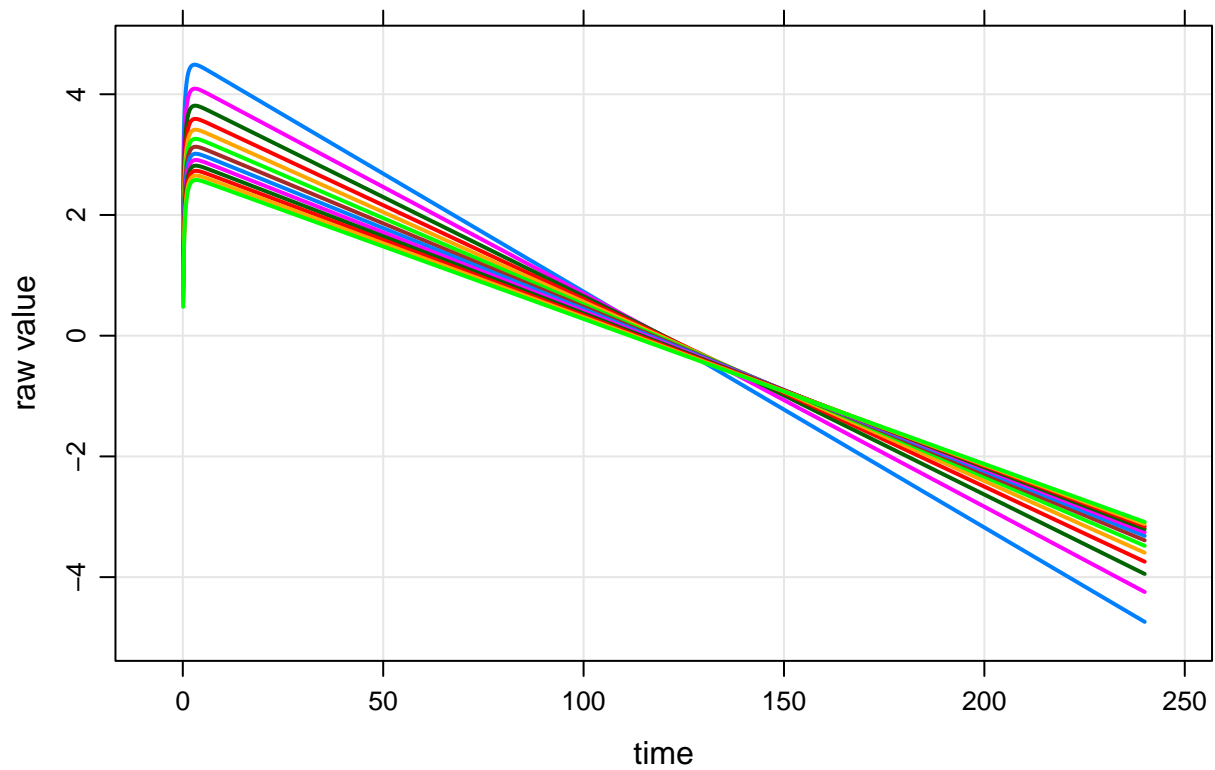
data:

- 1000 mg po x1
- WT from 20 to 140 by 10 kg

```
data <- expand.ev(WT = seq(20,140,10), amt=1000)
```

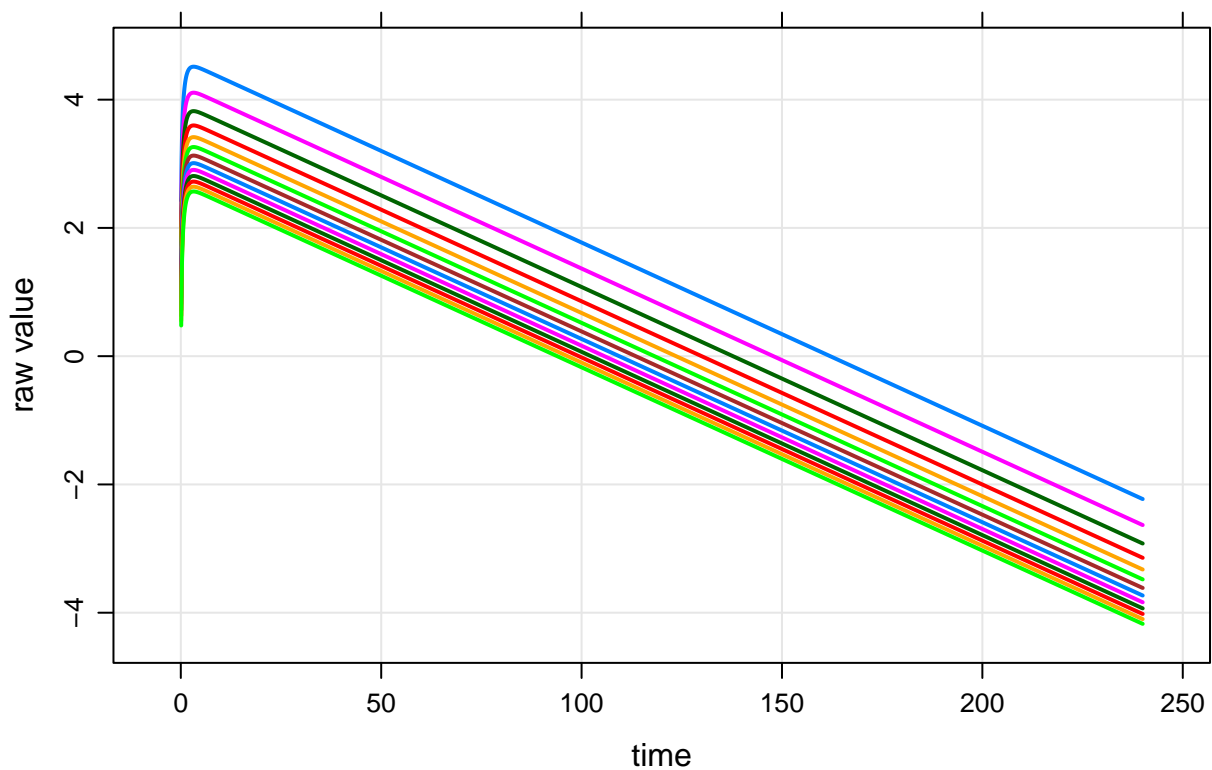
Simulate / plot logCP ~ time by ID

```
mod %>%
  data_set(data) %>%
  mrgsim(delta=0.1, end=240) %>%
  plot(log(CP) ~time)
```



What happens to half-life when WTCL=1?

```
mod %>%
  data_set(data) %>%
  param(WTCL = 1) %>%
  mrgsim(delta=0.1, end=240) %>%
  plot(log(CP)~time)
```



Add KIN, KOUT, IC50, FBIO

```
code <- '
$PARAM TVCL = 1, TVVC = 35, KA = 1.2
WT = 70, WTCL = 0.75
KIN = 100, KOUT = 2, IC50 = 4, FBIO = 0.6

$CMT GUT CENT RESP

$MAIN
double CL = TVCL*pow(WT/70,WTCL);
double V = TVVC*pow(WT/70,1);

RESP_0 = KIN/KOUT;

F_CENT = FBIO;

$ODE
double CP = CENT/V;
double INH = CP/(IC50+CP);

dxdt_GUT = -KA*GUT;
dxdt_CENT = KA*GUT - (CL/V)*CENT;
dxdt_RESP = KIN*(1-INH) - KOUT*RESP;

$TABLE
table(CP) = CENT/V;

$CAPTURE CL
'
```



```
mod <- mcode("specpd", code)
```

Check initial conditions

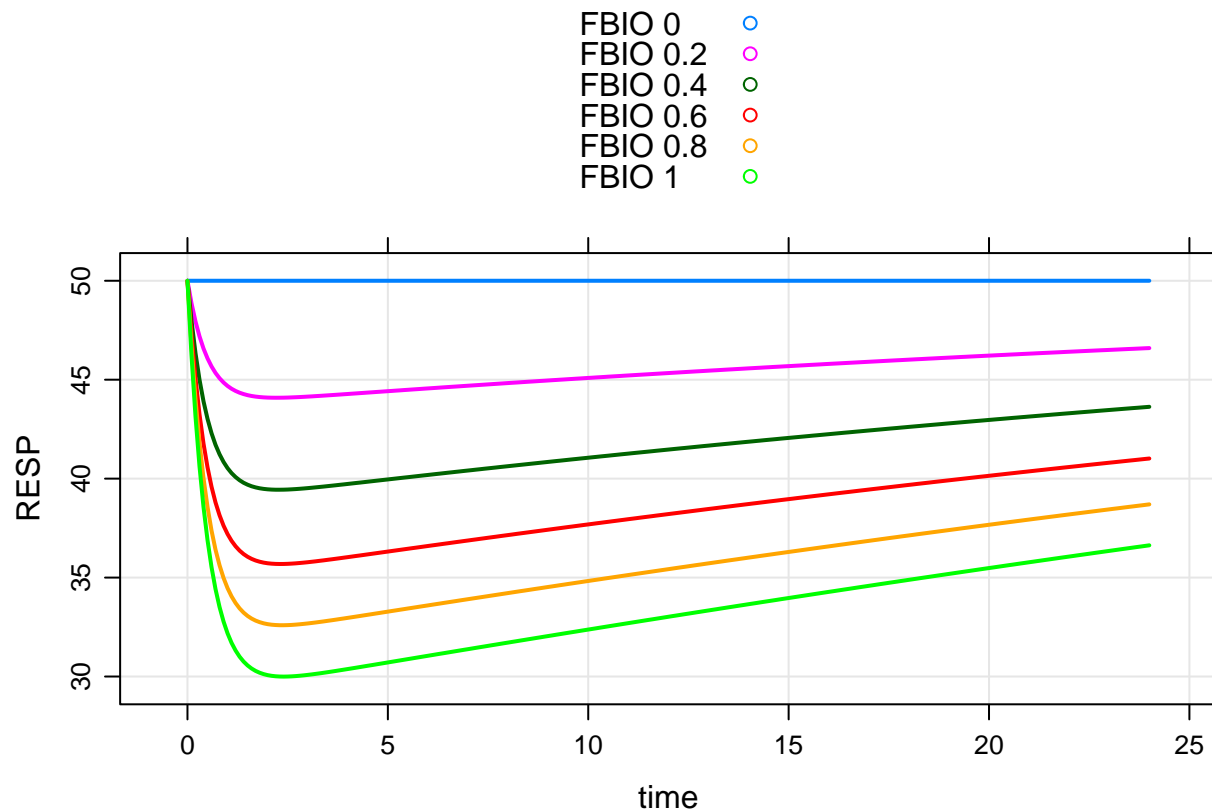
```
init(mod)
```

```
##
## Model initial conditions (N=3):
## name      value . name      value
## CENT (2)   0    | RESP (3)   50
## GUT (1)    0    | . ...      .
```

Simulate:

- IV bolus 100 mg x1
- Look at FBIO from 0 to 1 by 0.1 / knobs
- Plot response ~ time grouped by FBIO

```
mod %>%
  ev(amt=100, cmt=2) %>%
  update(delta=0.1) %>%
  Req(RESP) %>%
  knobs(FBIO = seq(0,1,0.2)) %>%
  plot()
```



Add random effects and \$OMEGA

```

code <- '
$PARAM TVCL = 1, TVVC = 35, KA = 1.2
KIN = 100, KOUT=2, IC50 = 2

$CMT GUT CENT RESP

$OMEGA 0.1 0.5 0.9

$MAIN
double CL = TVCL*exp(ETA(1));
double V = TVVC*exp(ETA(2));

RESP_0 = KIN/KOUT;

$ODE
double CP = CENT/V;
double INH = CP/(IC50+CP);

dxdt_GUT = -KA*GUT;
dxdt_CENT = KA*GUT - (CL/V)*CENT;
dxdt_RESP = KIN*(1-INH) - KOUT*RESP;

$CAPTURE CL V
'

```

Compile with mcode

```
mod <- mcode("specpop", code)
```

- Simulate 50 patients at 1000 mg dose, 100 kg
- end -> 120, delta -> 1

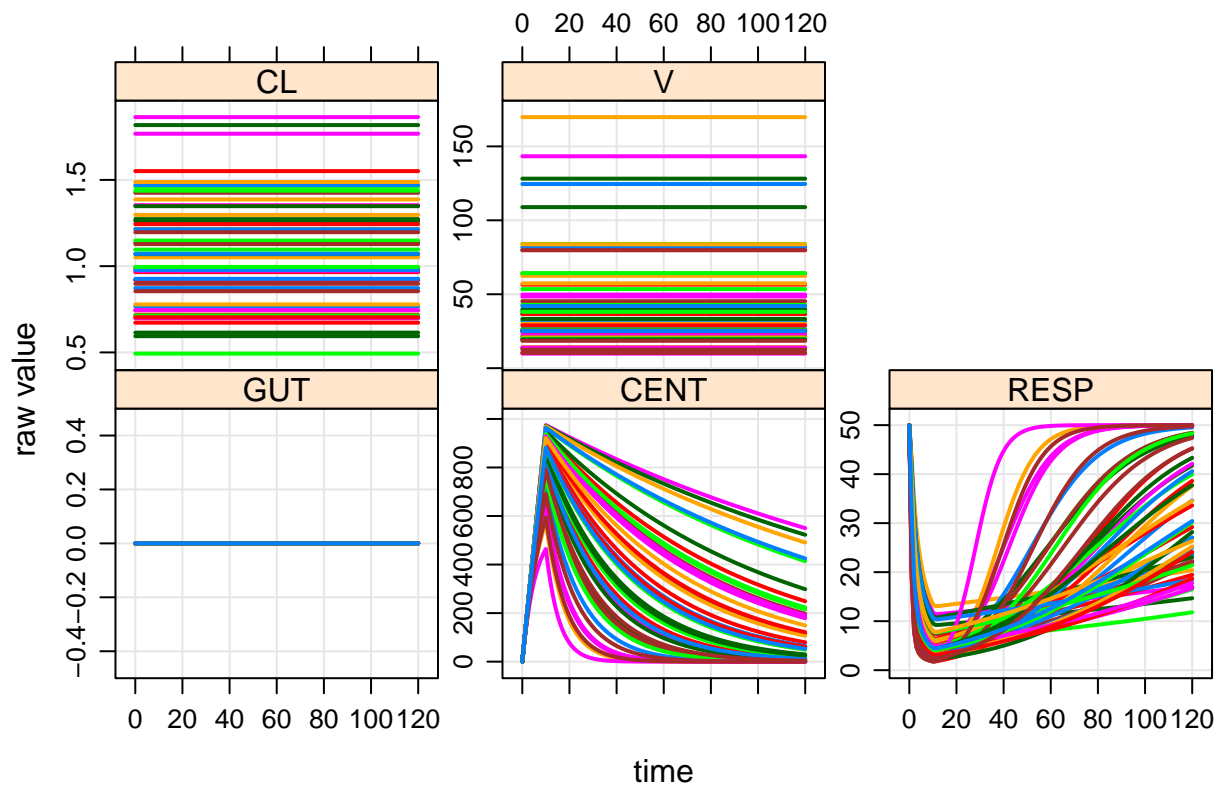
```

data <- expand.ev(ID=1:50, amt=1000, cmt=2, rate=100)

out <-
  mod %>%
  data_set(data) %>%
  mrgsim(delta=1, end=120)

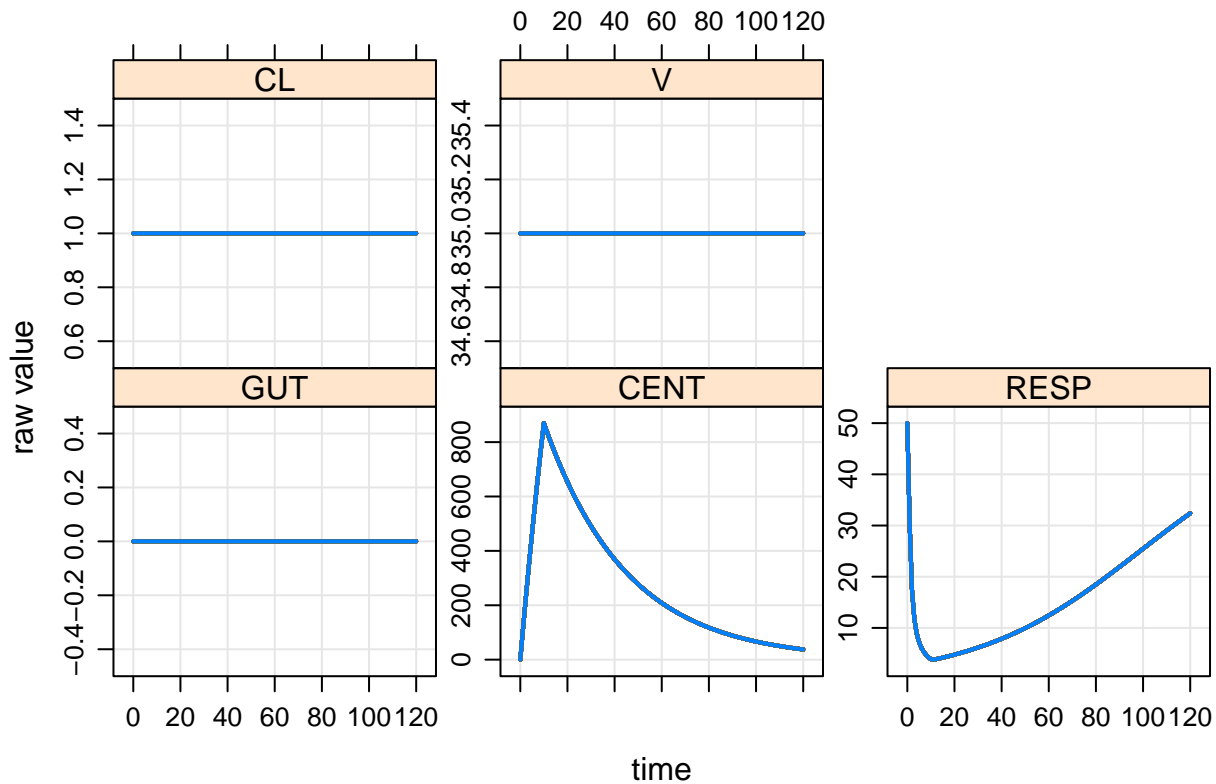
plot(out)

```



drop.re

```
mod %>%
  data_set(data) %>%
  zero.re %>%
  mrgsim(delta=1,end=120) %>%
  plot()
```



```
devtools::session_info()
```

```
## Session info -----

## setting value
## version R version 3.3.0 (2016-05-03)
## system x86_64, darwin13.4.0
## ui X11
## language (EN)
## collate en_US.UTF-8
## tz America/Chicago
## date 2016-05-13

## Packages -----

## package * version date source
## assertthat 0.1 2013-11-08 local
## DBI 0.4-1 2016-05-08 CRAN (R 3.3.0)
## devtools 1.10.0 2016-01-23 CRAN (R 3.2.1)
## digest 0.6.9 2016-01-08 CRAN (R 3.2.1)
## dplyr * 0.4.3 2015-09-01 CRAN (R 3.2.1)
## evaluate 0.8.3 2016-03-05 CRAN (R 3.2.3)
## formatR 1.3 2016-03-05 CRAN (R 3.2.3)
## htmltools 0.3.5 2016-03-21 CRAN (R 3.2.3)
## knitr 1.12.27 2016-04-30 Github (yihui/knitr@77de0a4)
## lattice 0.20-33 2015-07-14 CRAN (R 3.2.3)
## lazyeval 0.1.10 2015-01-02 CRAN (R 3.1.2)
```

##	magrittr	* 1.5	2014-11-22 CRAN (R 3.1.2)
##	memoise	1.0.0	2016-01-29 CRAN (R 3.2.1)
##	modmgr	* 0.0.1	2016-05-11 local
##	mrgsolve	* 0.6.1.9000	2016-05-11 local
##	R6	2.1.2	2016-01-26 CRAN (R 3.2.3)
##	Rcpp	0.12.4	2016-03-26 CRAN (R 3.2.3)
##	rmarkdown	0.9.6	2016-04-30 Github (rstudio/rmarkdown@e07c5f6)
##	stringi	1.0-1	2015-10-22 CRAN (R 3.2.1)
##	stringr	1.0.0	2015-04-30 CRAN (R 3.1.3)
##	yaml	2.1.13	2014-06-12 CRAN (R 3.0.2)