APS2025

Kyra Evers

2025-05-13

```
library(sdbuildR)
```

```
##
## Attaching package: 'sdbuildR'

## The following objects are masked from 'package:stats':
##
##     simulate, step

## The following object is masked from 'package:utils':
##
##     debugger
```

```
sdbuildR_setup()
```

```
## Julia version 1.11.5 at location C:\Users\kevers1\AppData\Roaming\R\data\R\JULIAC~1\julia\111~1.5\JU

## Loading setup script for JuliaCall...

## Finish loading setup script for JuliaCall.

## Setting up Julia environment for sdbuildR...
```

APS 2025 Example Model

```
# workload
# sleep
# colors

all_colors = c("#5855a2", "#f79556", "#fccc44","#d6d0a9",
               # "#bdb1e5",
               "#1e1735")
# scales::show_col(all_colors)
```

```
sfm = xmile()
```

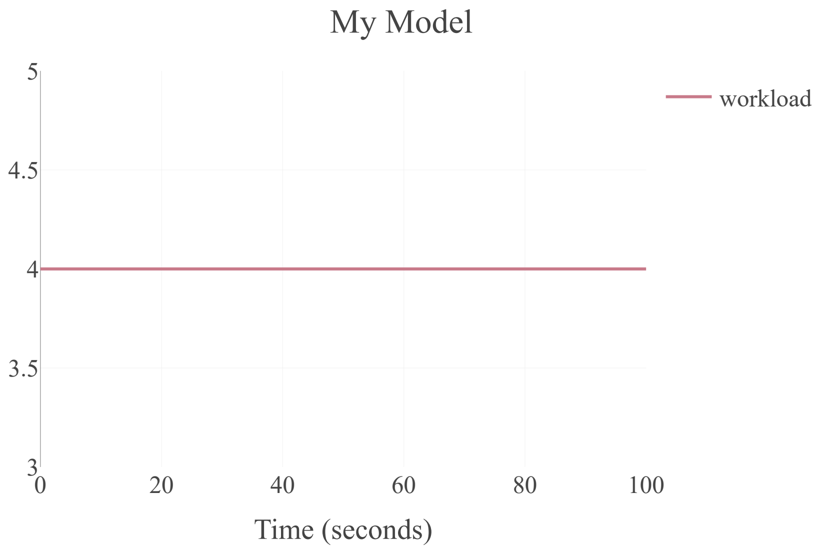
```r
sfm = xmile()
summary(sfm)
```

```
## Your model contains:
## * 0 Stocks
## * 0 Flows
## * 0 Constants
## * 0 Auxiliaries
## * 0 Graphical Functions
## * 0 Custom model units
## * 0 Macros
##
## Simulation time: 0.0 to 100.0 seconds (dt = 0.01)
## Simulation settings: solver euler in Julia
```

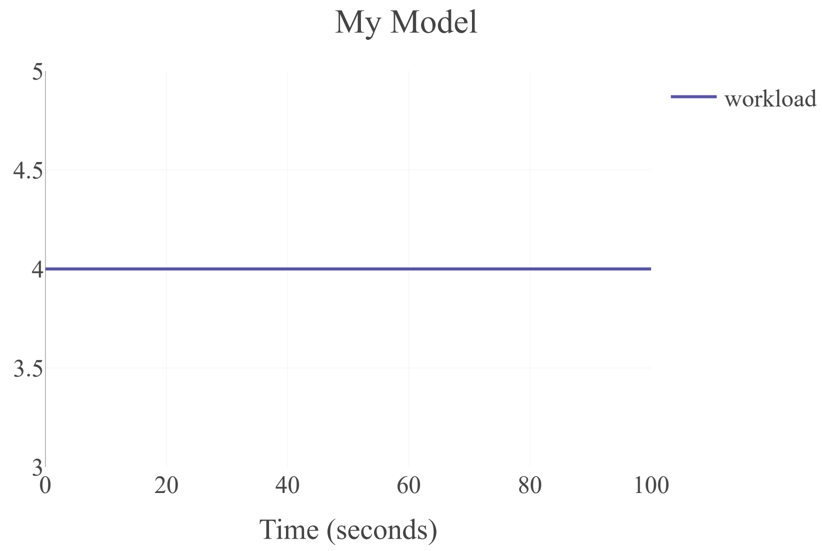
```r
sfm = sfm %>%
  build("workload", "stock",
        eqn = 4, units = "hours/day")

plot(sfm)
```

3

```
sfm = sfm %>%
  build("workload", "stock",
        eqn = 4, units = "hours/day")

sim = simulate(sfm)
plot(sim)
```

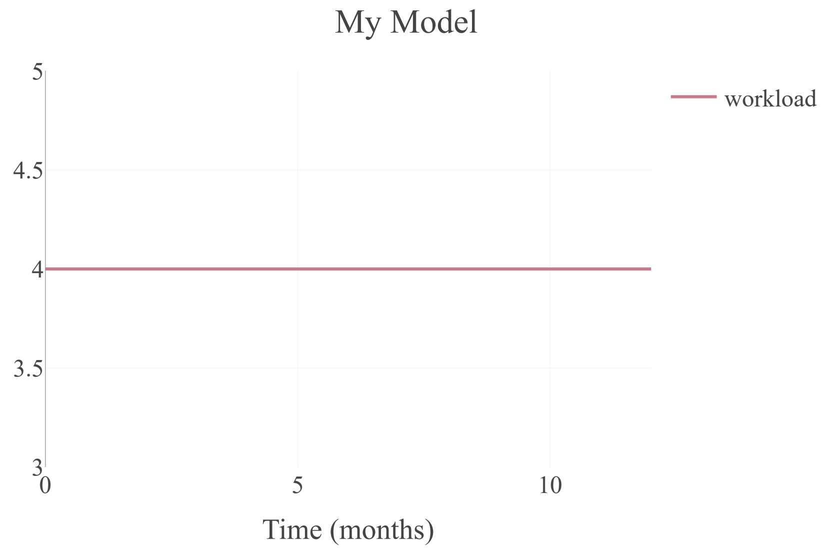


```
colors = all_colors
plot(sim, colors=colors)
```

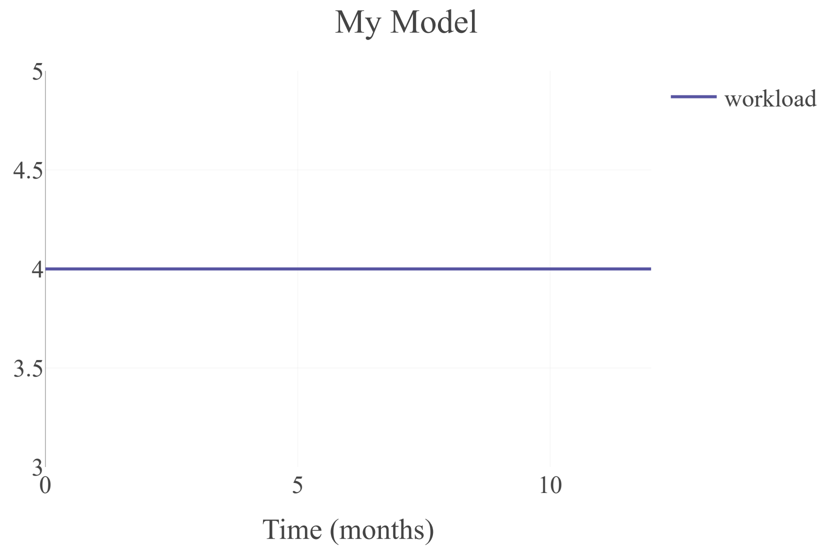


```r
sfm = sfm %>%
  sim_specs(stop = 12, time_units = "month")
```

```
sim = simulate(sfm)
plot(sim)
```

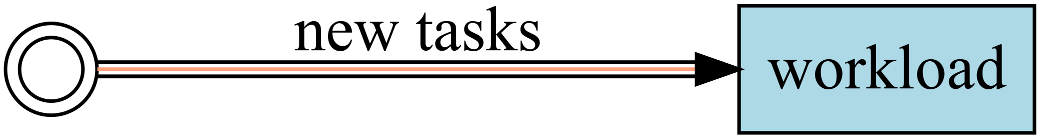


```
colors = all_colors
plot(sim, colors=colors)
```

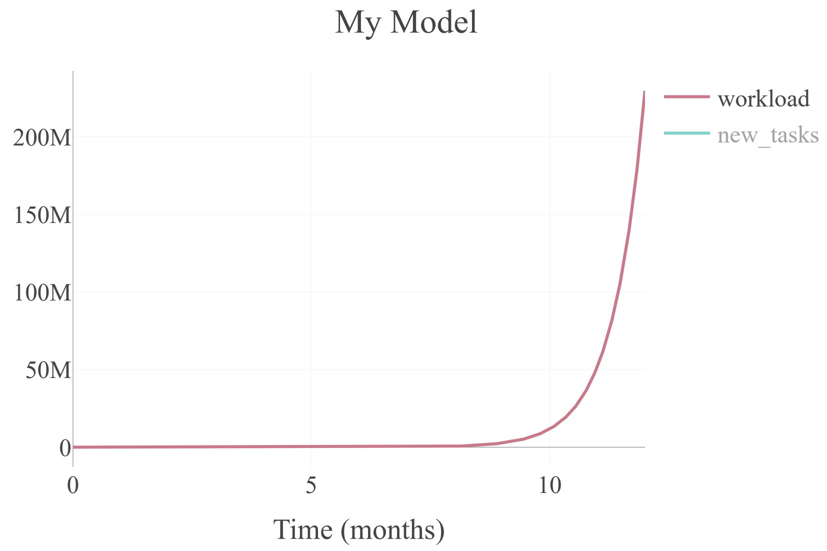


```
sfm = sfm %>%
  build("new_tasks", "flow",
        eqn = "workload * work_growth",
        to = "workload", units = "hours/day/month")
```

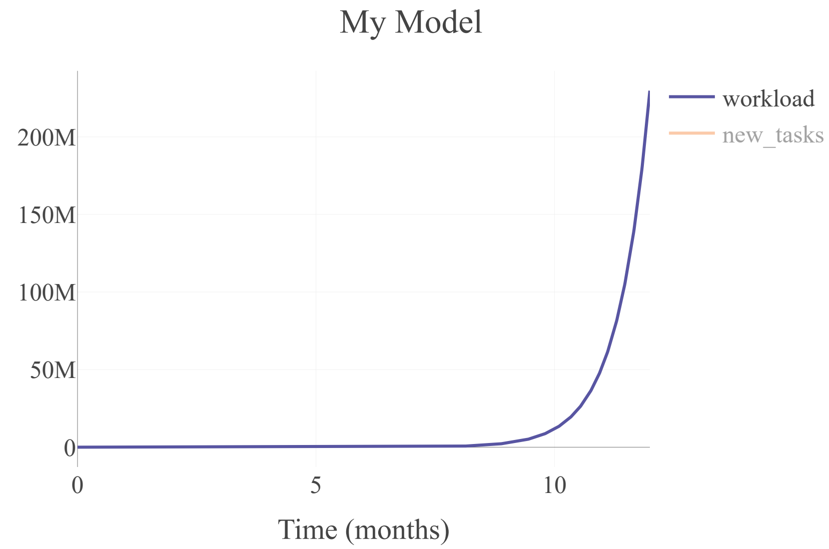
```
sfm = sfm %>%
  build("new_tasks", "flow",
        eqn = "workload * work_growth",
        to = "workload", units = "hours/day/month") %>%
  build("work_growth", "constant",
        eqn = 1.5, units = "1/month")

plot(sfm)
```


new tasks
workload

```r
sfm = sfm %>%
  build("new_tasks", "flow",
        eqn = "workload * work_growth",
        to = "workload", units = "hours/day/month") %>%
  build("work_growth", "constant",
        eqn = 1.5, units = "1/month")

sim = simulate(sfm)
plot(sim)
```

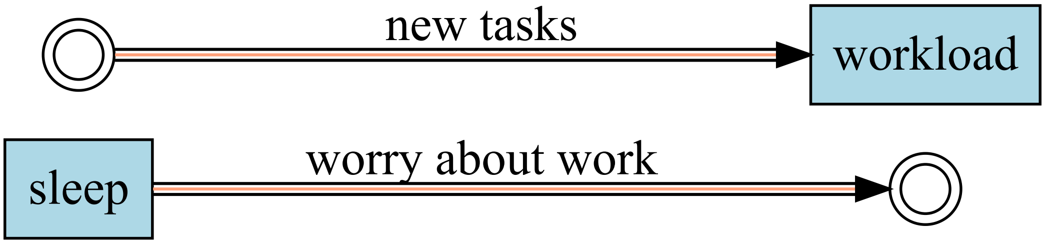


```
colors = all_colors
plot(sim, colors=colors)
```

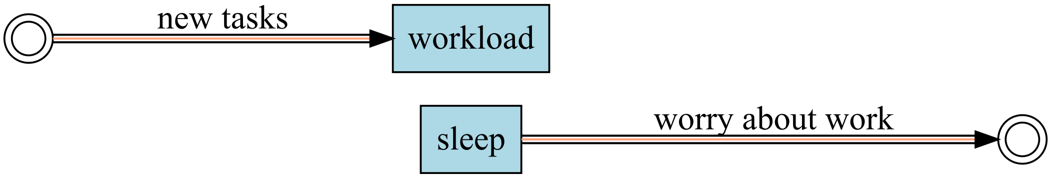

My Model
workload
new_tasks
200M
150M
100M
50M
0
0
5
10
Time (months)

```
sfm = sfm %>%
  build(c("sleep", "necessary_sleep", "worry_factor"),
        c("stock", "constant", "constant"),
        eqn = c("necessary_sleep", 8, .1),
        units = c("hours/day", "hours/day", "1/month"))
```

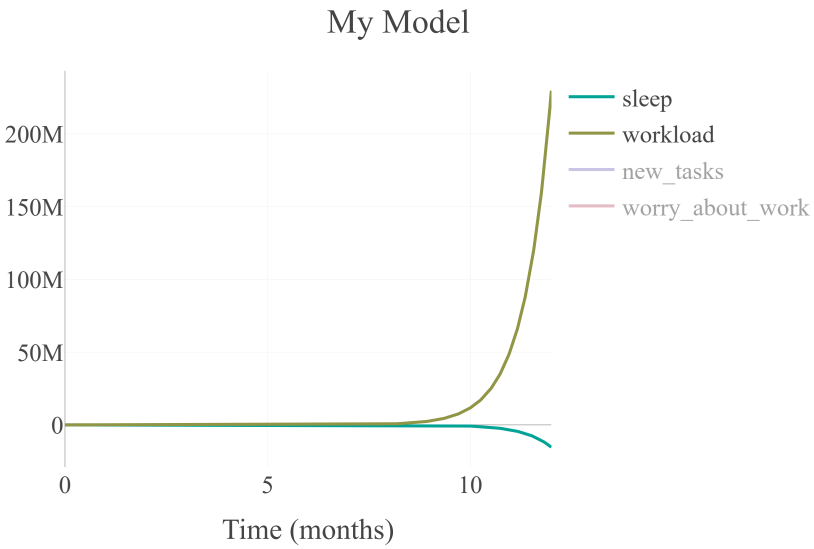
```r
sfm = sfm %>%
  build("worry_about_work", "flow",
        eqn = "workload * worry_factor",
        from = "sleep", units = "hours/day/month")

plot(sfm)
```


new tasks
workload
sleep
worry about work

```
plot(sfm, center_stocks = TRUE)
```

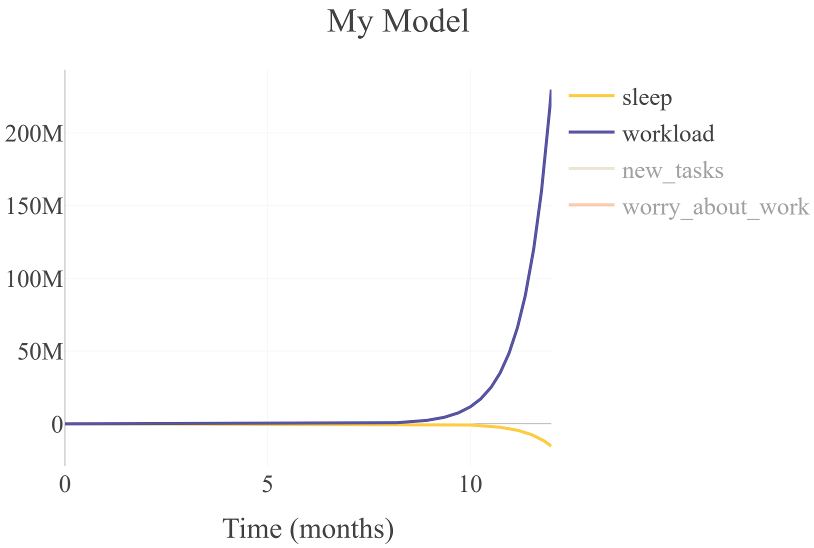


```r
sfm = sfm %>%
  build(c("sleep", "necessary_sleep", "worry_factor"),
        c("stock", "constant", "constant"),
        eqn = c("necessary_sleep", 8, .1),
        units = c("hours/day", "hours/day", "1/month")) %>%

  build("worry_about_work", "flow",
        eqn = "workload * worry_factor",
        from = "sleep", units = "hours/day/month")

sim = simulate(sfm)
plot(sim)
```

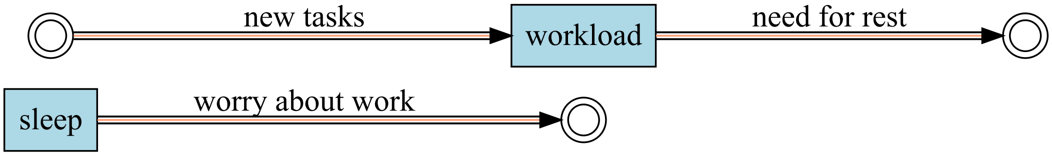


```
colors = all_colors
colors[1] = all_colors[2] # switch 1 and 2
```

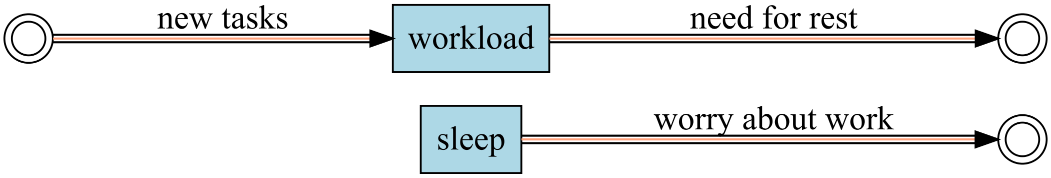
```
colors[2] = all_colors[1] # switch 1 and 2
plot(sim, colors=colors)
```



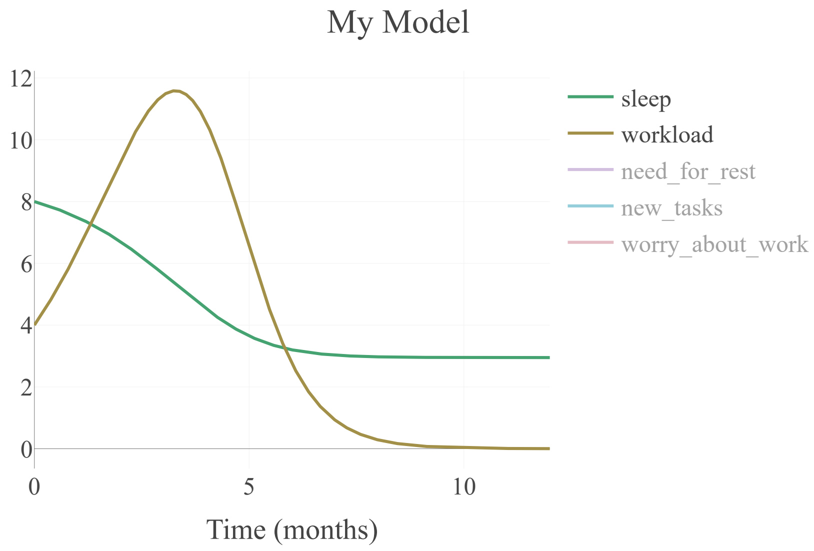
```r
sfm = sfm %>%
  build("need_for_rest",  "flow",
        eqn = "workload * necessary_sleep / sleep / u('1month')",
        from = "workload",
        units = "hours/day/month")

plot(sfm)
```


new tasks
workload
need for rest
sleep
worry about work

```r
plot(sfm, center_stocks = TRUE)
```

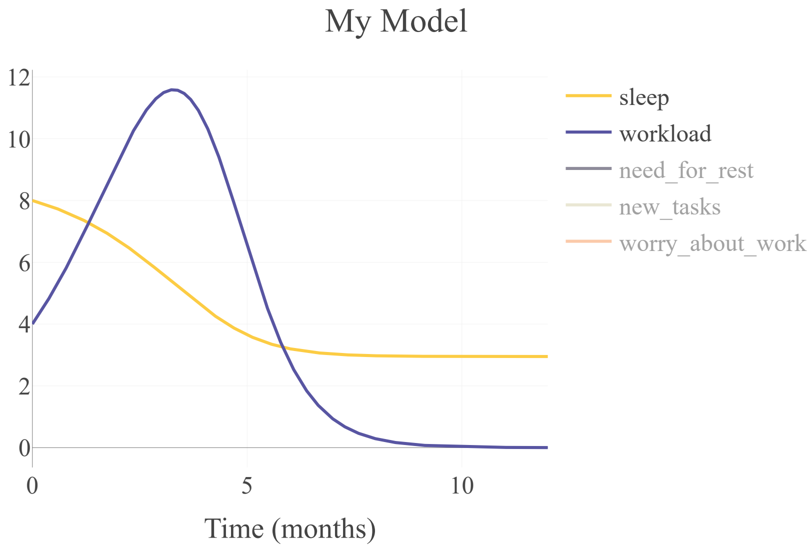


```r
sfm = sfm %>%
  build("need_for_rest",  "flow",
        eqn = "workload * necessary_sleep / sleep / u('1month')",
        from = "workload",
        units = "hours/day/month")

sim = simulate(sfm)
plot(sim)
```



```
colors = all_colors
colors[1] = all_colors[2] # switch 1 and 2
```

```
colors[2] = all_colors[1] # switch 1 and 2
plot(sim, colors=colors)
```



Model iteration

```r
sfm = sfm %>% build("workload", eqn = 8)
```

```r
sfm = sfm %>% build("sleep", change_name = "sleep_hours")
```

```r
sfm = sfm %>% build("necessary_sleep", change_type = "aux")
```

```r
sfm = sfm %>% build("workload", erase = TRUE)
```

```r
sfm = xmile() %>%
  macro("sig", eqn = "function(x, slope = 1, midpoint = .5){
        1 / (1 + exp(-slope*(x-midpoint)))
        }")
```

Debugger

```
sfm = xmile() %>%
  build("new_tasks", "flow", to = "workload")

debugger(sfm)
```

```
## Problems:

## * Your model has no stocks.
##
## * These flows are connected to a stock that does not exist:
##   - new_tasks
##
## * The properties below contain references to undefined variables.
## Please define the missing variables or correct any spelling mistakes.
## - new_tasks$to: workload

##
## Potentially problematic:

## * These variables have an equation of 0:
## - new_tasks
```

Units

regex

custom

```
sfm = sfm %>%
  model_units("quality", doc = "Quality of life") %>%
  model_units("QALY", eqn = "years*quality", doc = "Quality-adjusted life year")
```

```
sfm = xmile() %>%
  model_units("BMI", eqn = "kilograms per meters squared",
              doc = "Body Mass Index")
```

```
sfm = xmile() %>%
  model_units("BMI", eqn = "kilograms per meters squared",
              doc = "Body Mass Index")
as.data.frame(sfm)
```

```
##          type name    eqn             doc
## 1 model_units  BMI kg/m^2 Body Mass Index
```

in equations

```
sfm = xmile() %>%
 build("weight_gain", "flow", eqn = "u('2 BMI / year')",
       units = "BMI/year")
```

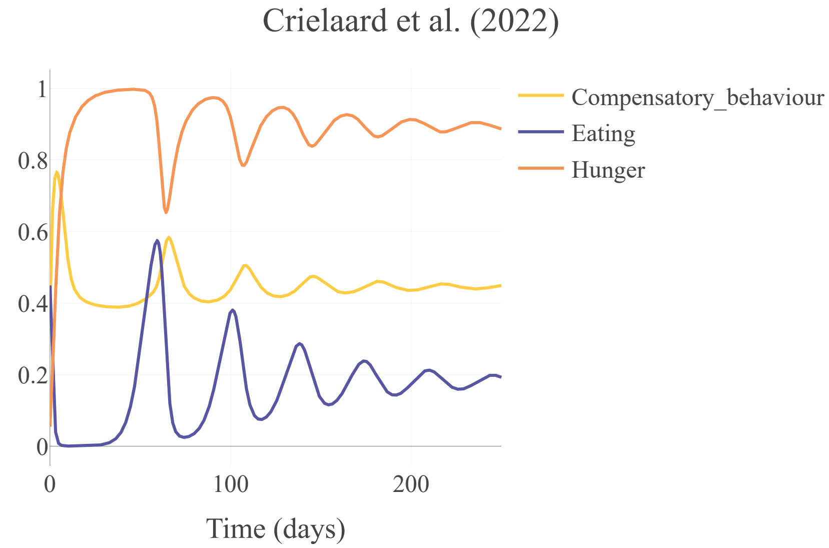
not from scratch

```
URL = "https://insightmaker.com/insight/5LxQr0waZGgBcPJcNTC029/Crielaard-et-al-2022"
```

```
sfm = insightmaker_to_sfm(URL)
```

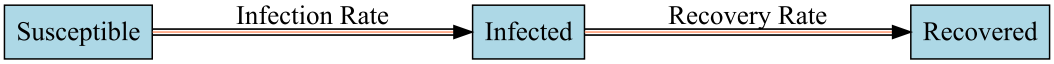
```
## This model uses Insight Maker version 38, whereas the package was based on Insight Maker version 37.
```

```
sim = simulate(sfm, only_stocks = TRUE)
plot(sim, colors = colors)
```

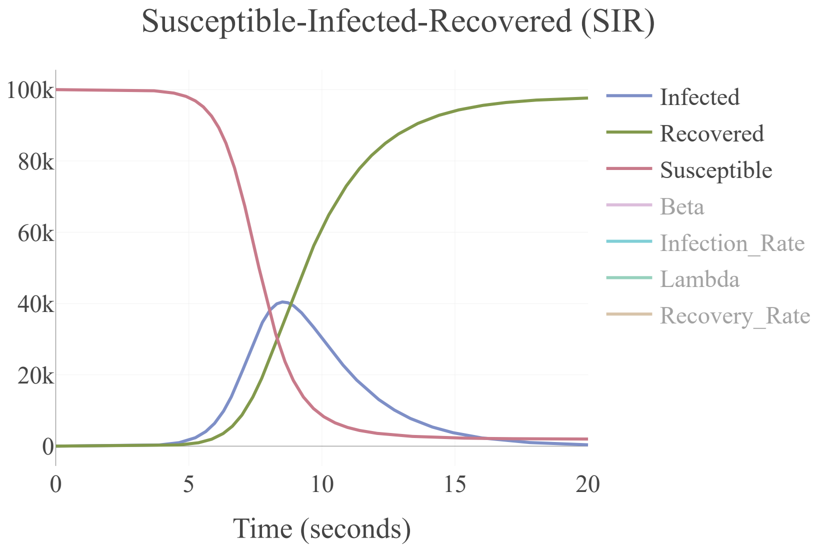


```
sfm = xmile("SIR")
```

```
plot(sfm)
```


Susceptible
Infection Rate
Infected
Recovery Rate
Recovered

```
sim = simulate(sfm)
plot(sim)
```


Susceptible-Infected-Recovered (SIR)
100k
80k
60k
40k
20k
Time (seconds)
Infected
Recovered
Susceptible
Beta
Infection_Rate
Lambda
Recovery_Rate