`gamar`: an R interface to the GAMA agent-based simulation platform

Marc Choisy
Lucie Contamin
Ho Bich Hai
Nicolas Marilleau
Jean-Daniel Zucker

Population dynamics modeling

Increased level of detail

population-based

differential equations

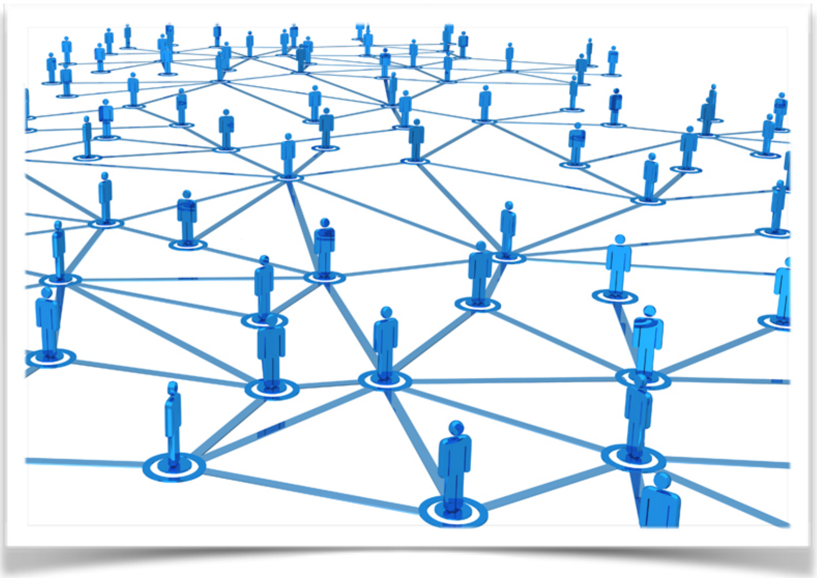
```
bvpSolve
dde
deSolve
diffeqr
PBSddesolve
```

individual-based

Gillespie algorithms

```
adaptivetau
GillespieSSA
```

agent-based



Agents are

- **autonomous**
- **heterogeneous**
- **active**
- **adaptive**

Population dynamics modeling

Increased level of detail

population-based

differential equations

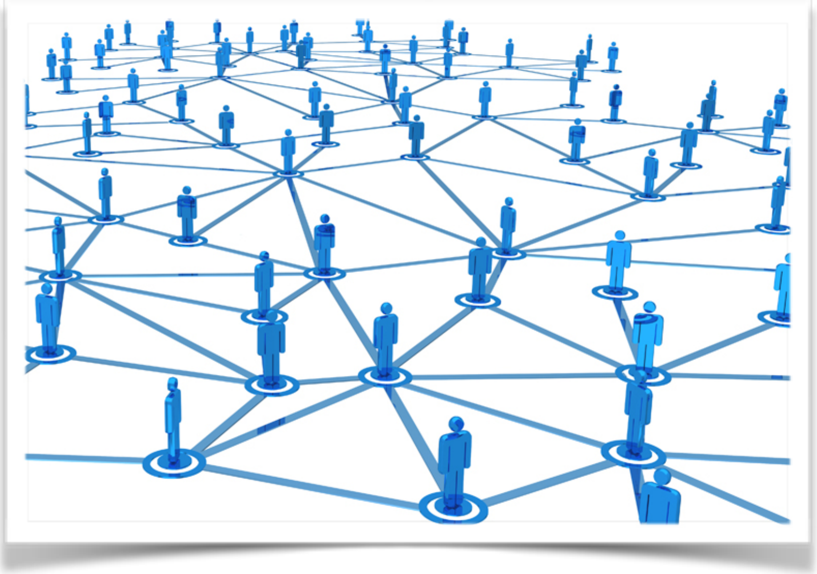
```
bvpSolve
dde
deSolve
diffeqr
PBSddesolve
```

individual-based

Gillespie algorithms

```
adaptivetau
GillespieSSA
```

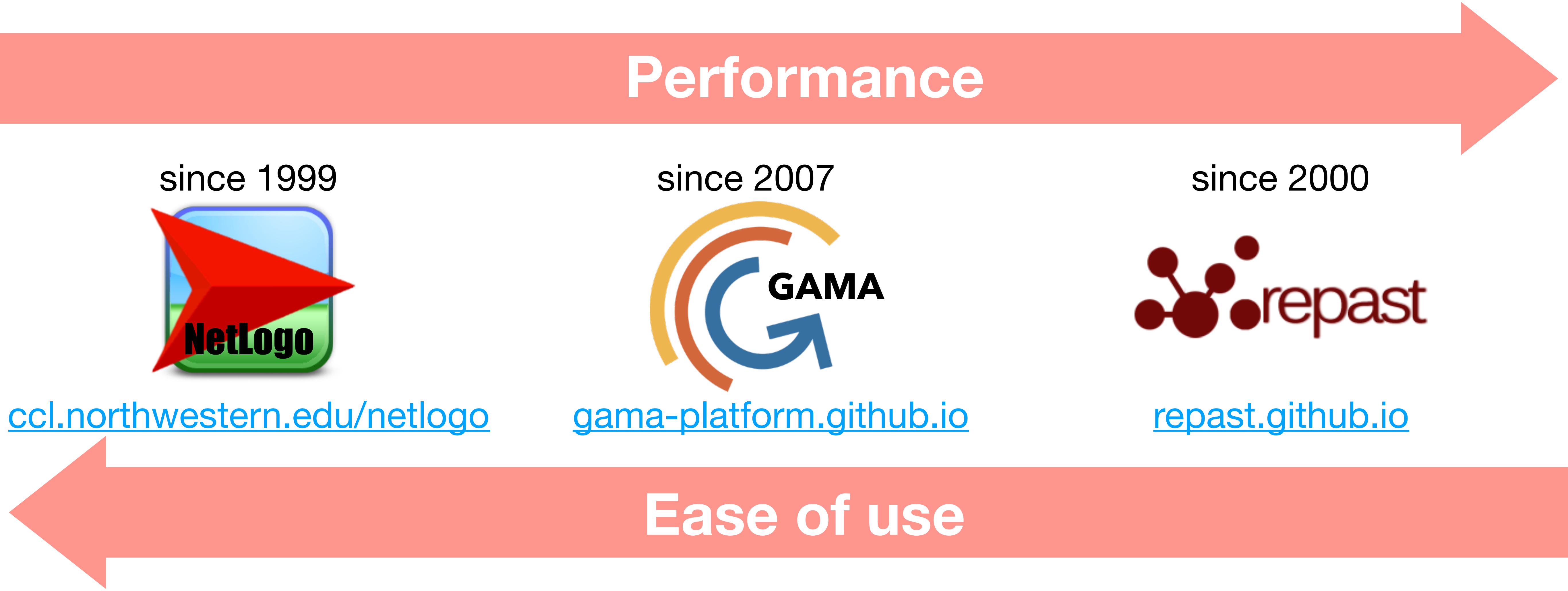
agent-based



Agents are

- **autonomous**
- **heterogeneous**
- **active**
- **adaptive**

Agent-based modeling platforms



Performance

since 1999

ccl.northwestern.edu/netlogo

since 2007

gama-platform.github.io

since 2000

repast.github.io

Ease of use

```
NetLogoR
RNetLogo
```

```
rrepast
```

Agent-based modeling platforms

Performance

since 1999

NetLogo

ccl.northwestern.edu/netlogo

since 2007

GAMA

gama-platform.github.io

since 2000

repast

repast.github.io

Ease of use

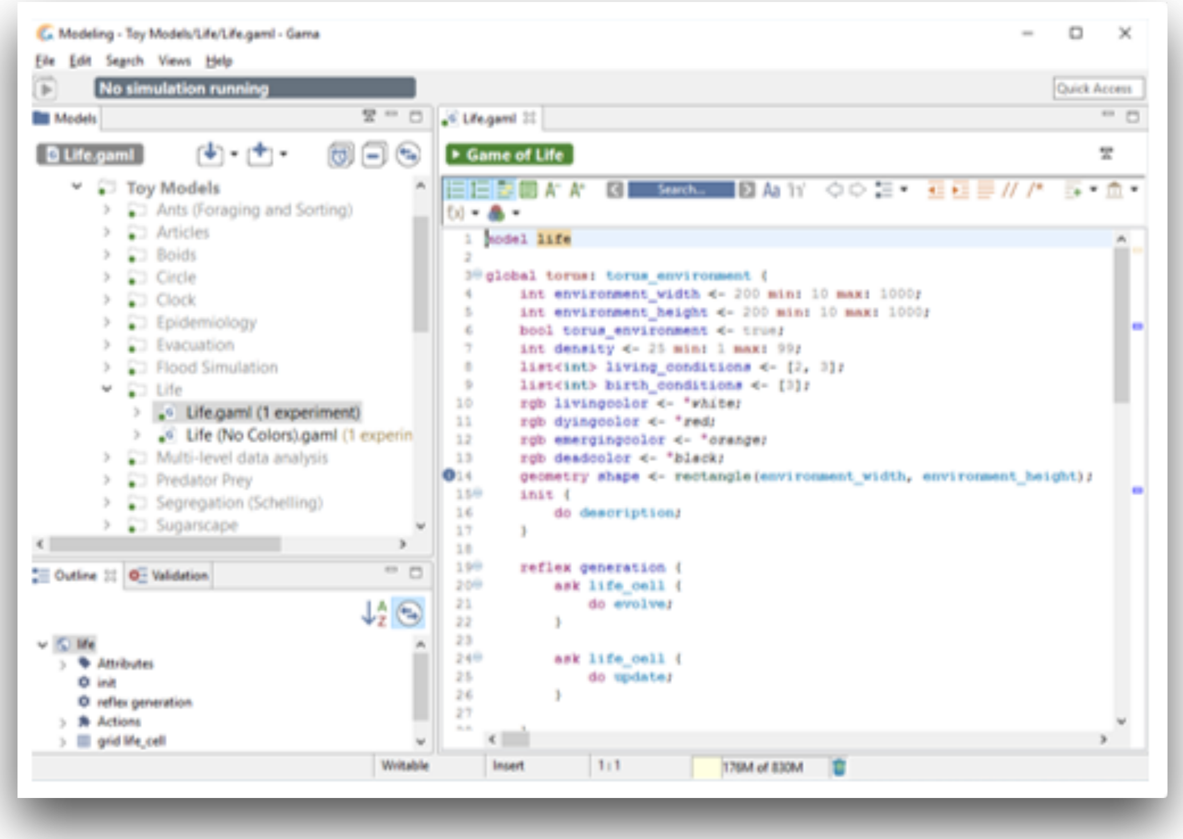
`NetLogoR`
`RNetLogo`

`rrepast`

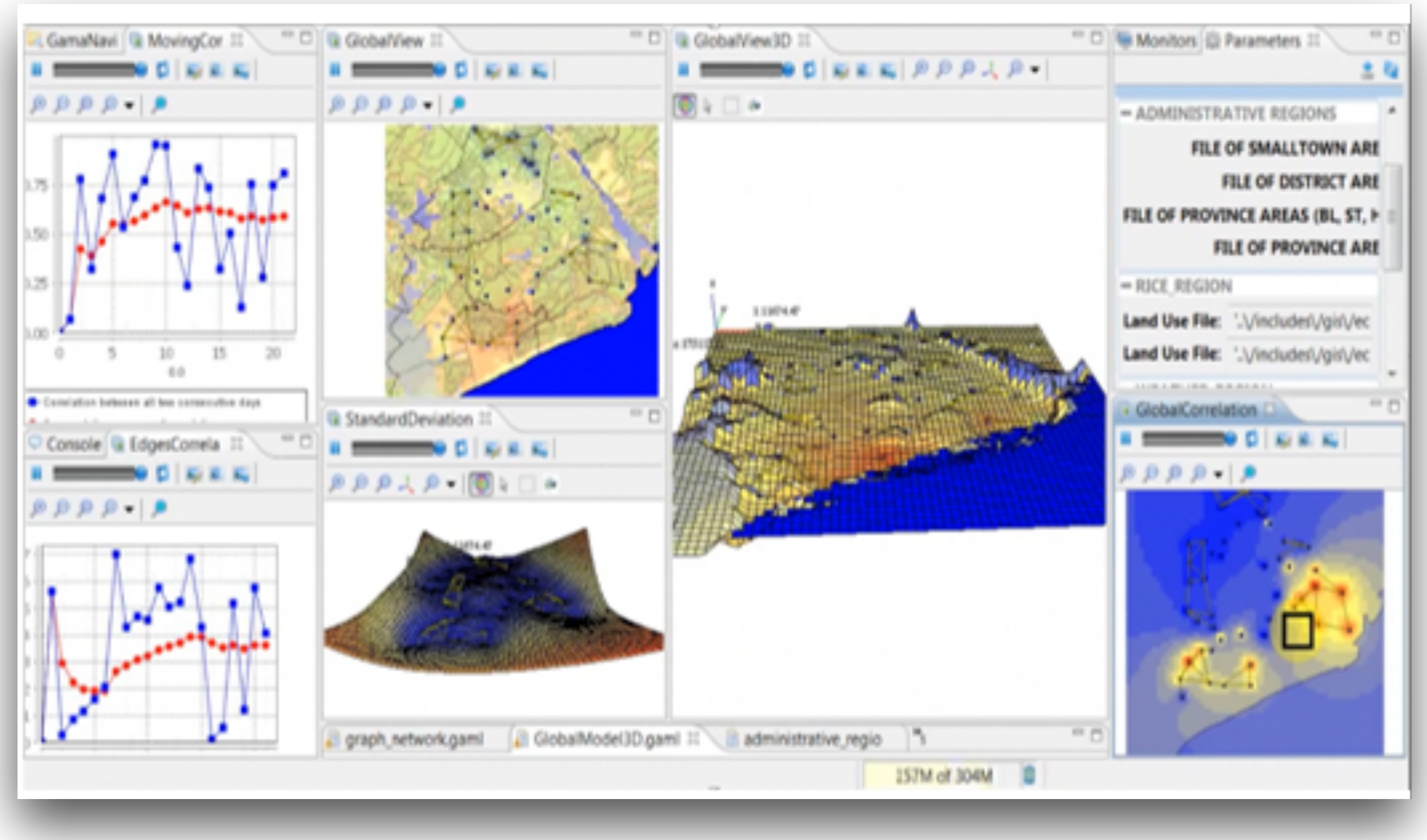
The GAMA simulation platform

- a language: GAML
- a user-interface
- a fast and parallelized engine
- integration with GIS data

https://gama-platform.github.io



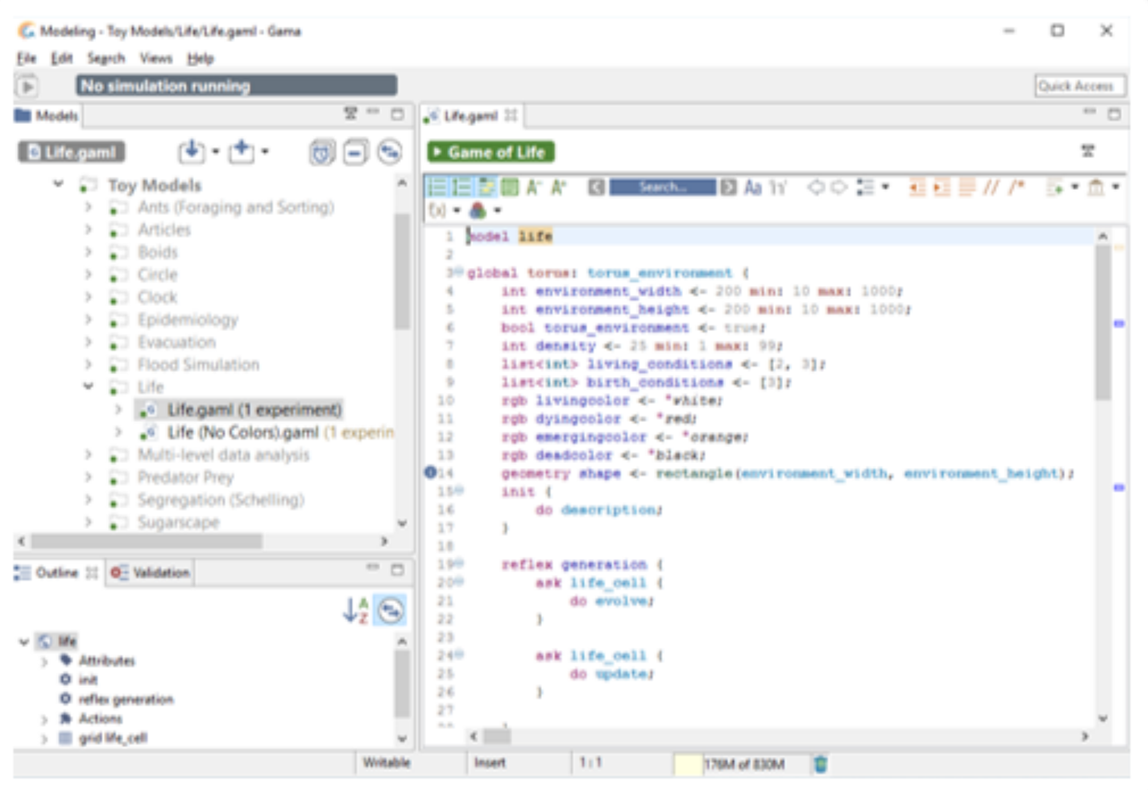
```
model life

global torus: torus_environment {
    int environment_width <- 200 min: 10 max: 1000;
    int environment_height <- 200 min: 10 max: 1000;
    bool torus_environment <- true;
    int density <- 25 min: 1 max: 99;
    list<int> living_conditions <- [2, 3];
    list<int> birth_conditions <- [3];
    rgb livingcolor <- #white;
    rgb dyingcolor <- #red;
    rgb emergingcolor <- #orange;
    rgb deadcolor <- #black;
    geometry shape <- rectangle(environment_width, environment_height);
    init {
        do description;
    }

    reflex generation {
        ask life_cell {
            do evolve;
        }

        ask life_cell {
            do update;
        }
```



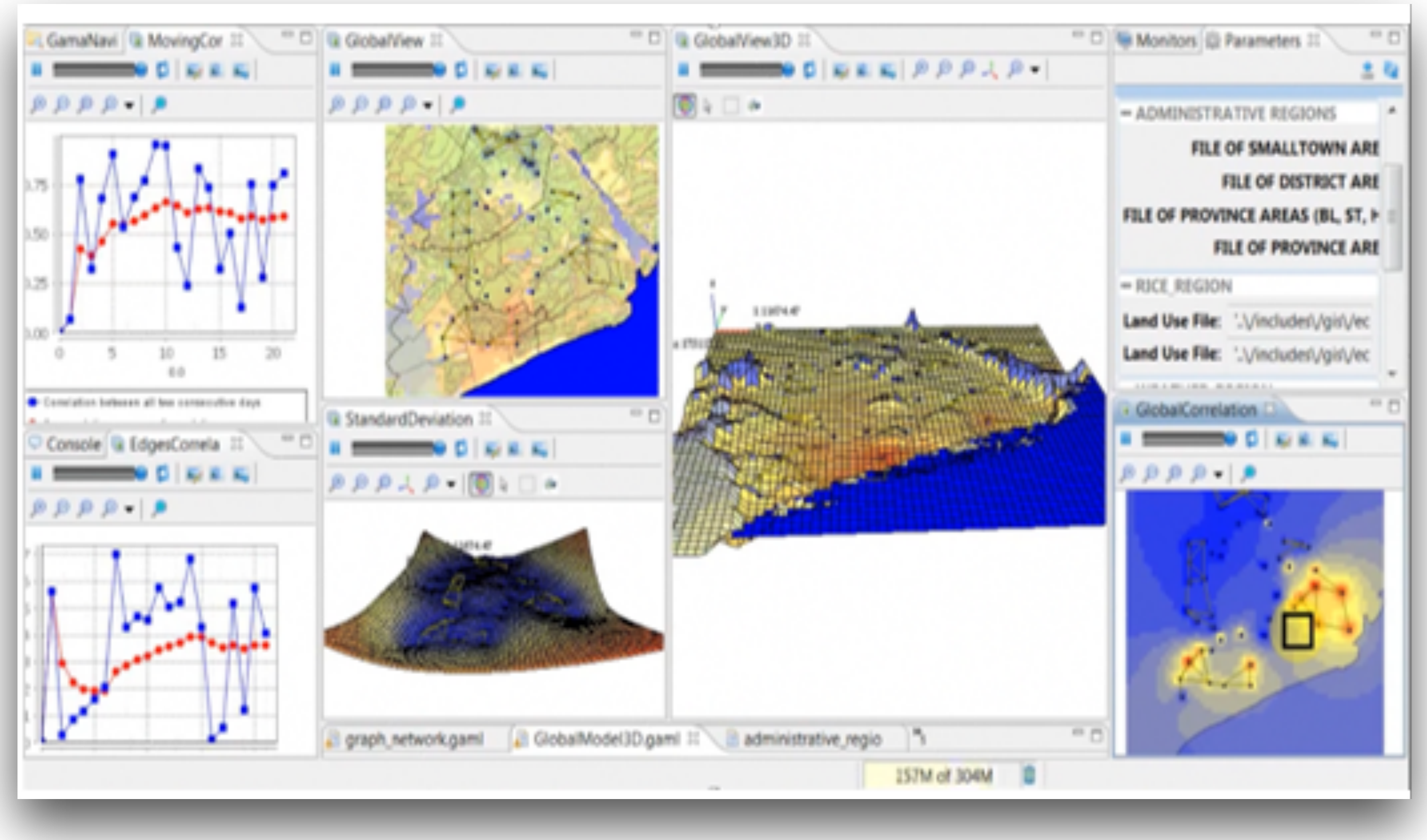
The GAMA simulation platform

- a language: GAML
- a user-interface
- a fast and parallelized engine
- integration with GIS data

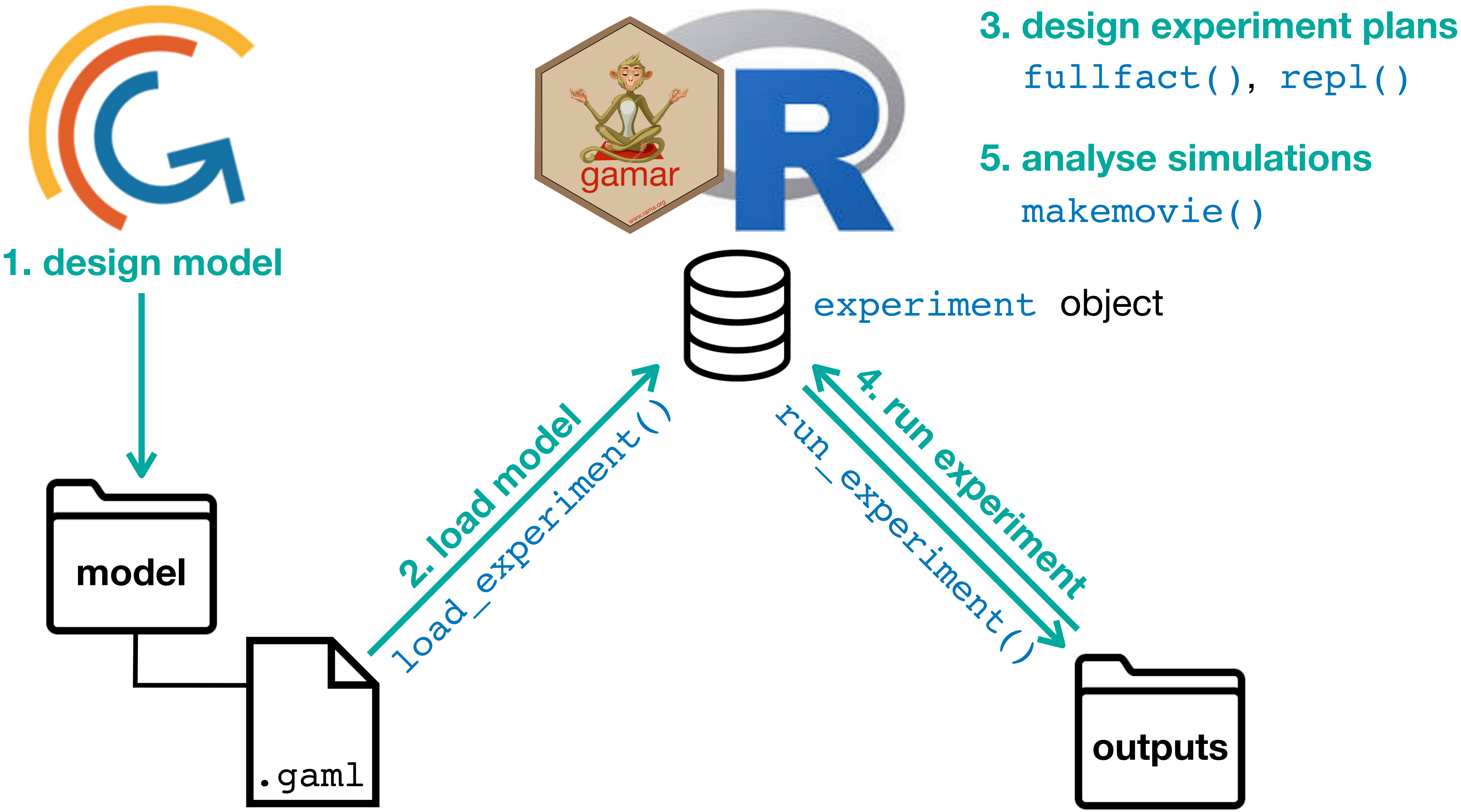
https://gama-platform.github.io



```
model life

global torus: torus_environment {
    int environment_width <- 200 min: 10 max: 1000;
    int environment_height <- 200 min: 10 max: 1000;
    bool torus_environment <- true;
    int density <- 25 min: 1 max: 99;
    list<int> living_conditions <- [2, 3];
    list<int> birth_conditions <- [3];
    rgb livingcolor <- #white;
    rgb dyingcolor <- #red;
    rgb emergingcolor <- #orange;
    rgb deadcolor <- #black;
    geometry shape <- rectangle(environment_width, environment_height);
    init {
        do description;
    }

    reflex generation {
        ask life_cell {
            do evolve;
        }

        ask life_cell {
            do update;
        }
```



The `gamar` R package



1. design model

2. load model `load_experiment()`

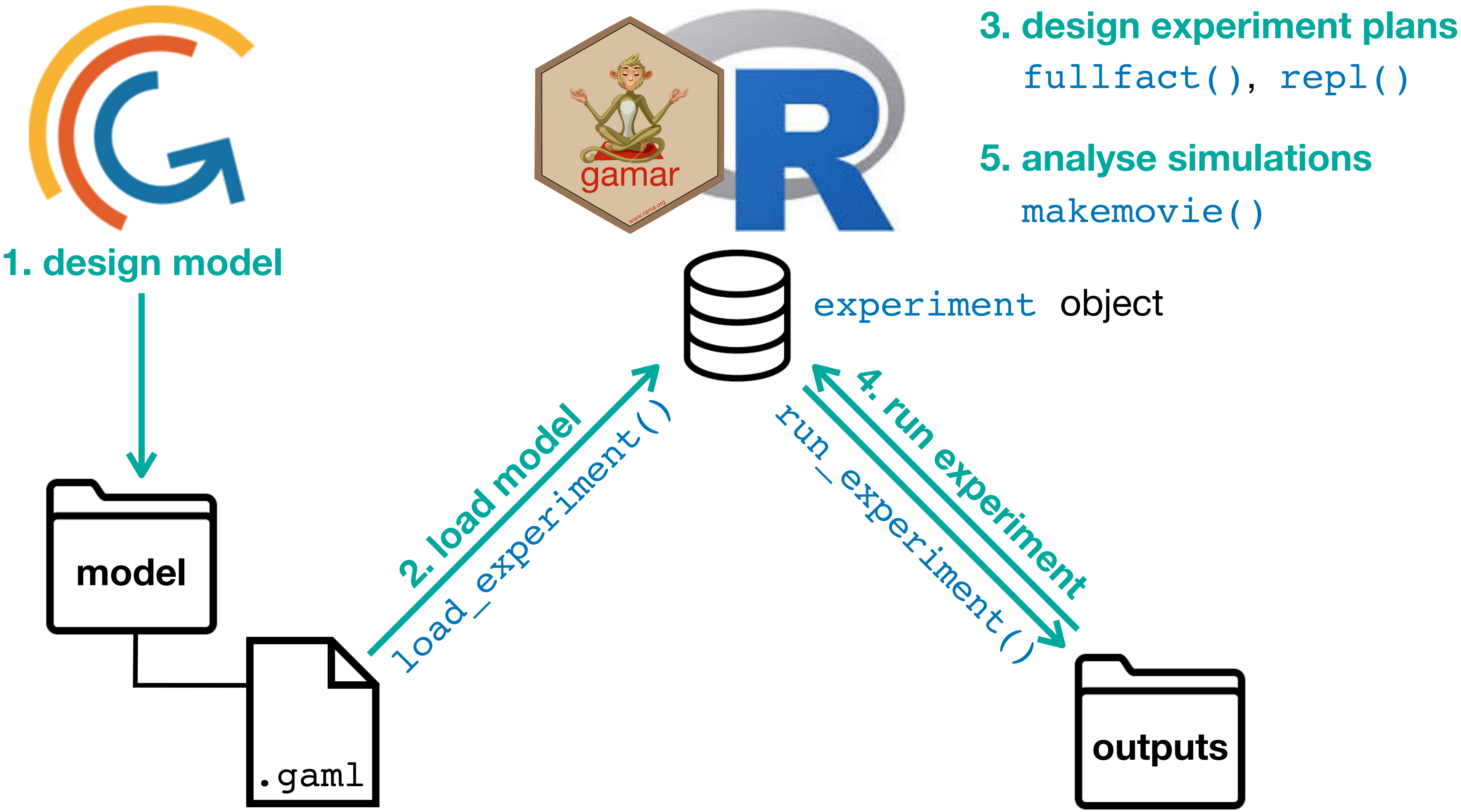
3. design experiment plans `fullfact()`, `repl()`

4. run experiment `run_experiment()`

5. analyse simulations `makemovie()`

`experiment` object

The `gamar` R package



1. design model

model

`.gaml`

2. load model
`load_experiment()`

`experiment` object

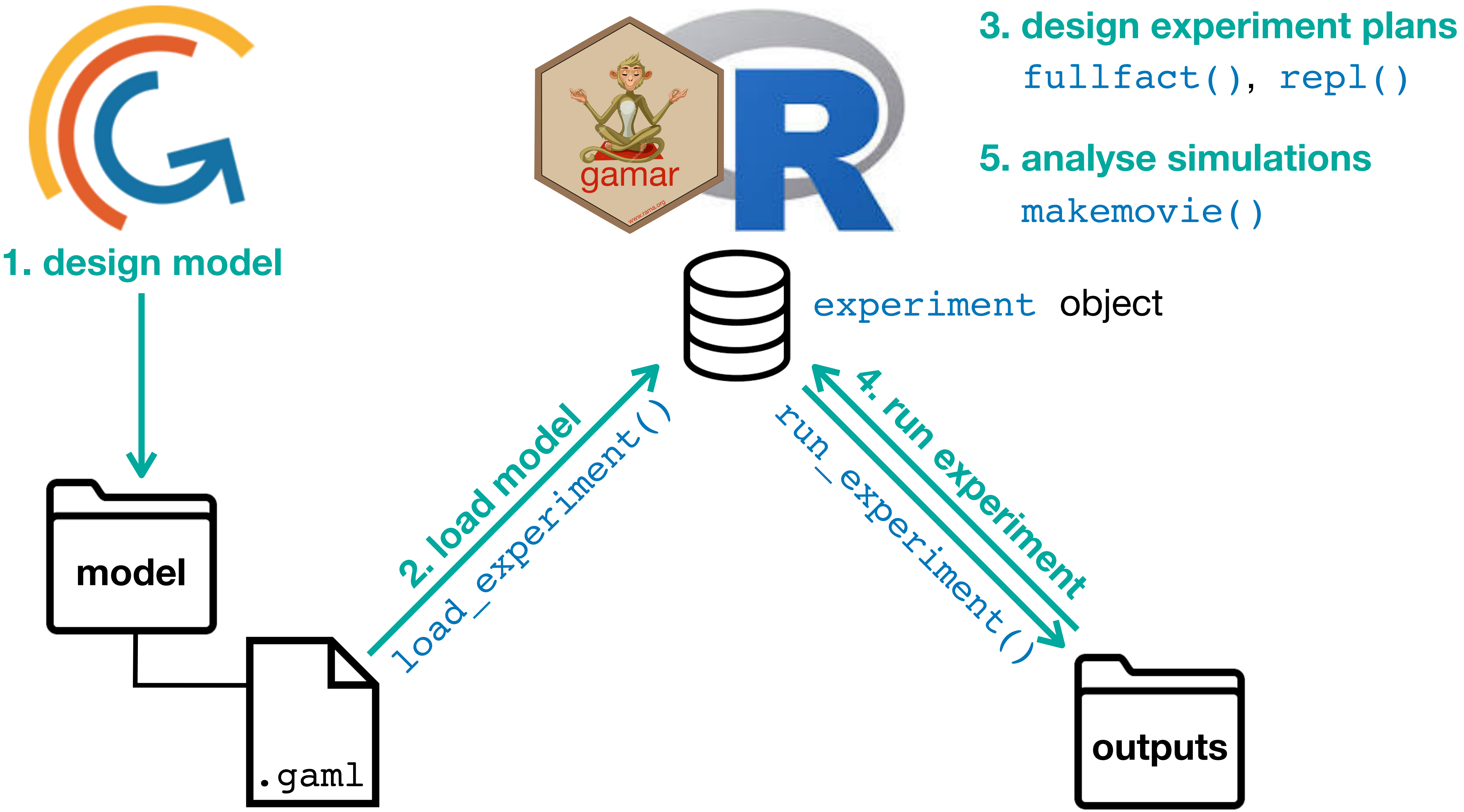
3. design experiment plans
`fullfact()`, `repl()`

4. run experiment
`run_experiment()`

5. analyse simulations
`makemovie()`

outputs

The `gamar` R package



1. design model

3. design experiment plans

`fullfact(), repl()`

5. analyse simulations

`makemovie()`

`experiment` object

2. load model

`load_experiment()`

4. run experiment

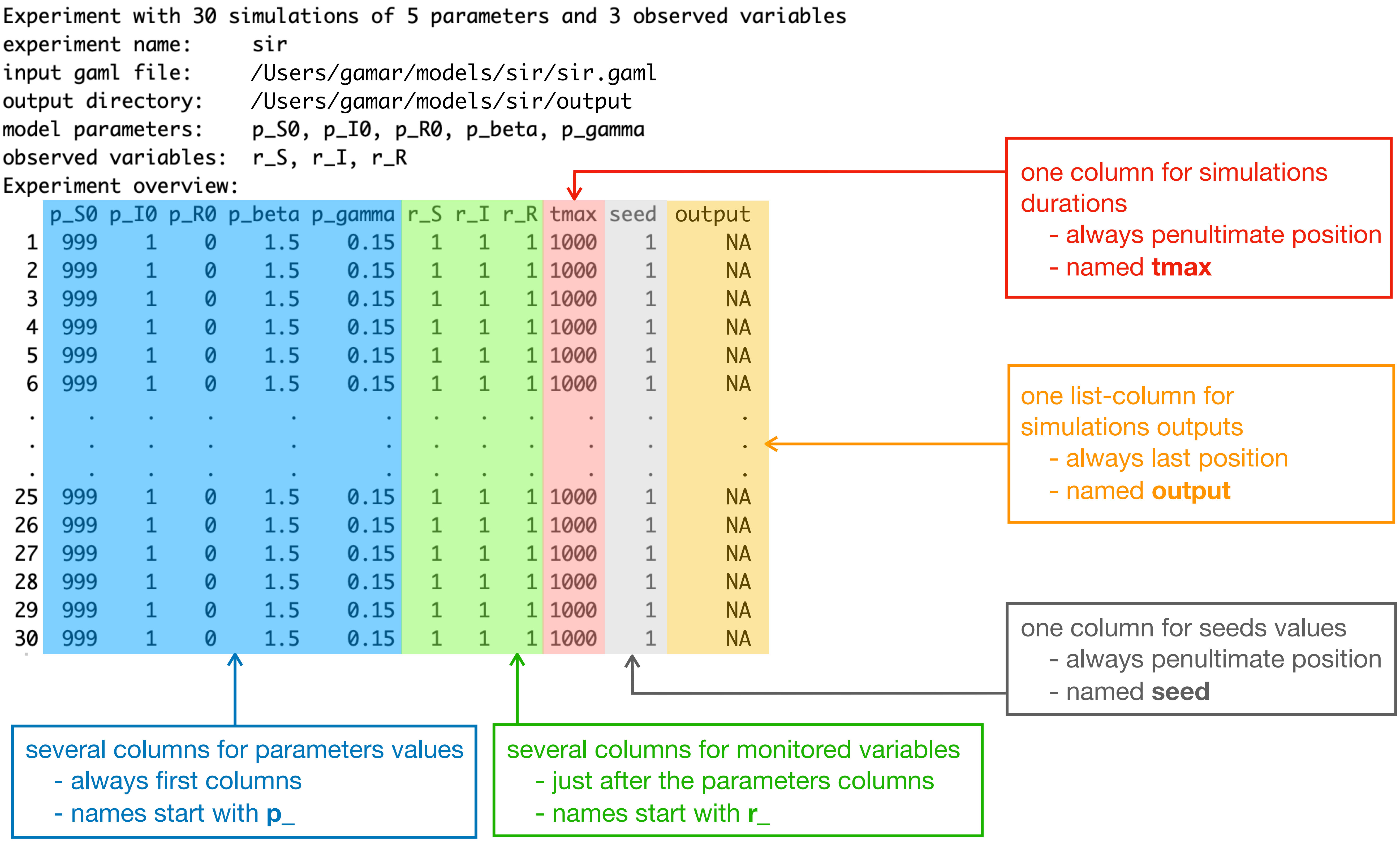
`run_experiment()`

model

`.gaml`

outputs

The `experiment` class



```
Experiment with 30 simulations of 5 parameters and 3 observed variables
experiment name:      sir
input gaml file:      /Users/gamar/models/sir/sir.gaml
output directory:     /Users/gamar/models/sir/output
model parameters:     p_S0, p_I0, p_R0, p_beta, p_gamma
observed variables:   r_S, r_I, r_R
Experiment overview:
   p_S0 p_I0 p_R0 p_beta p_gamma r_S r_I r_R tmax seed output
1   999    1    0    1.5    0.15   1   1   1 1000    1     NA
2   999    1    0    1.5    0.15   1   1   1 1000    1     NA
3   999    1    0    1.5    0.15   1   1   1 1000    1     NA
4   999    1    0    1.5    0.15   1   1   1 1000    1     NA
5   999    1    0    1.5    0.15   1   1   1 1000    1     NA
6   999    1    0    1.5    0.15   1   1   1 1000    1     NA
.     .    .    .      .       .   .   .   .    .    .      .
.     .    .    .      .       .   .   .   .    .    .      .
.     .    .    .      .       .   .   .   .    .    .      .
25  999    1    0    1.5    0.15   1   1   1 1000    1     NA
26  999    1    0    1.5    0.15   1   1   1 1000    1     NA
27  999    1    0    1.5    0.15   1   1   1 1000    1     NA
28  999    1    0    1.5    0.15   1   1   1 1000    1     NA
29  999    1    0    1.5    0.15   1   1   1 1000    1     NA
30  999    1    0    1.5    0.15   1   1   1 1000    1     NA
```

one column for simulations durations
- always penultimate position
- named **tmax**

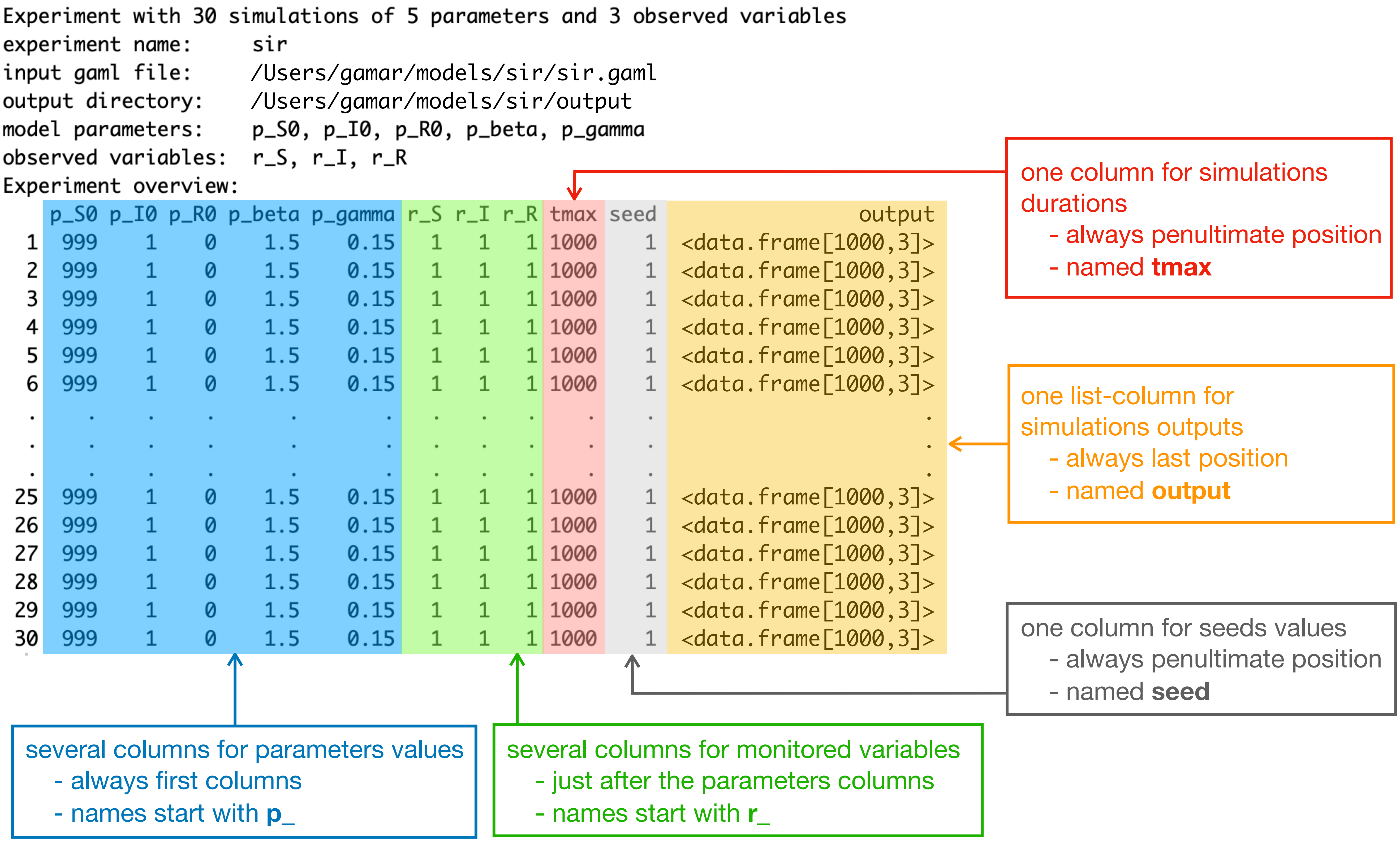
one list-column for simulations outputs
- always last position
- named **output**

one column for seeds values
- always penultimate position
- named **seed**

several columns for parameters values
- always first columns
- names start with **p_**

several columns for monitored variables
- just after the parameters columns
- names start with **r_**

The `experiment` class



```
Experiment with 30 simulations of 5 parameters and 3 observed variables
experiment name:     sir
input gaml file:     /Users/gamar/models/sir/sir.gaml
output directory:    /Users/gamar/models/sir/output
model parameters:    p_S0, p_I0, p_R0, p_beta, p_gamma
observed variables:  r_S, r_I, r_R
Experiment overview:
   p_S0 p_I0 p_R0 p_beta p_gamma r_S r_I r_R tmax seed               output
1   999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
2   999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
3   999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
4   999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
5   999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
6   999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
.     .    .    .      .       .   .   .   .    .    .                    .
.     .    .    .      .       .   .   .   .    .    .                    .
.     .    .    .      .       .   .   .   .    .    .                    .
25  999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
26  999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
27  999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
28  999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
29  999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
30  999    1    0    1.5    0.15   1   1   1 1000    1 <data.frame[1000,3]>
```

one column for simulations durations
- always penultimate position
- named **tmax**

one list-column for simulations outputs
- always last position
- named **output**

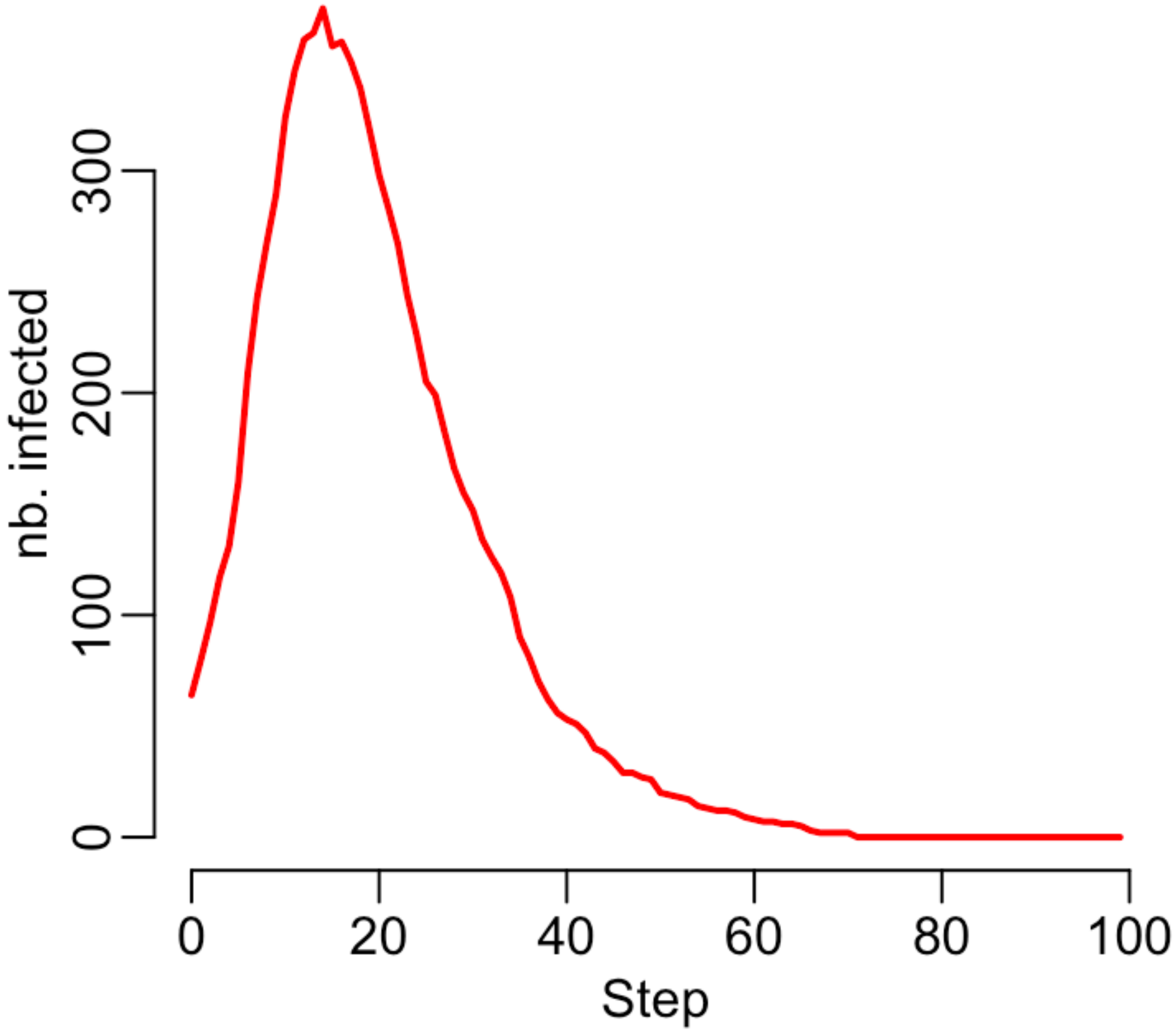
one column for seeds values
- always penultimate position
- named **seed**

several columns for parameters values
- always first columns
- names start with **p_**

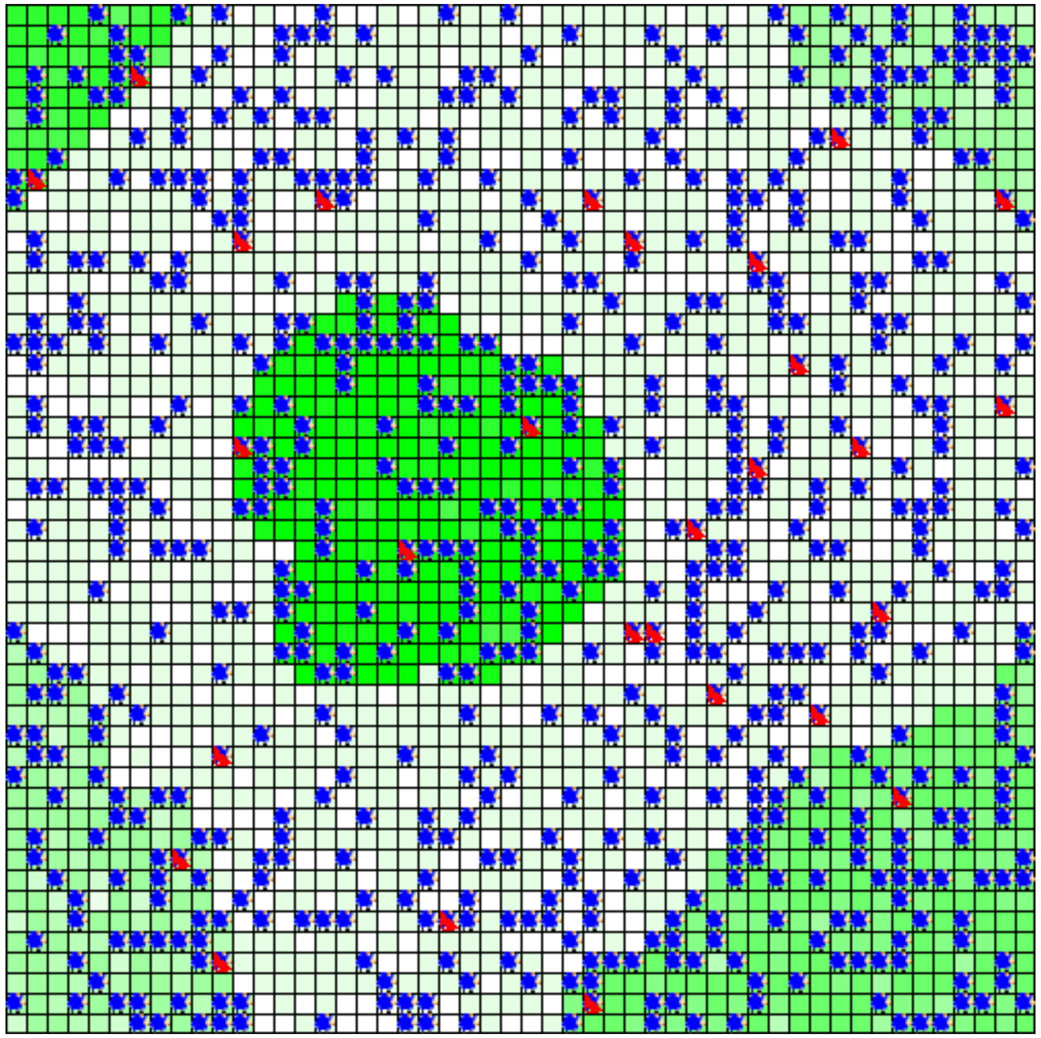
several columns for monitored variables
- just after the parameters columns
- names start with **r_**

Visualizing simulation results

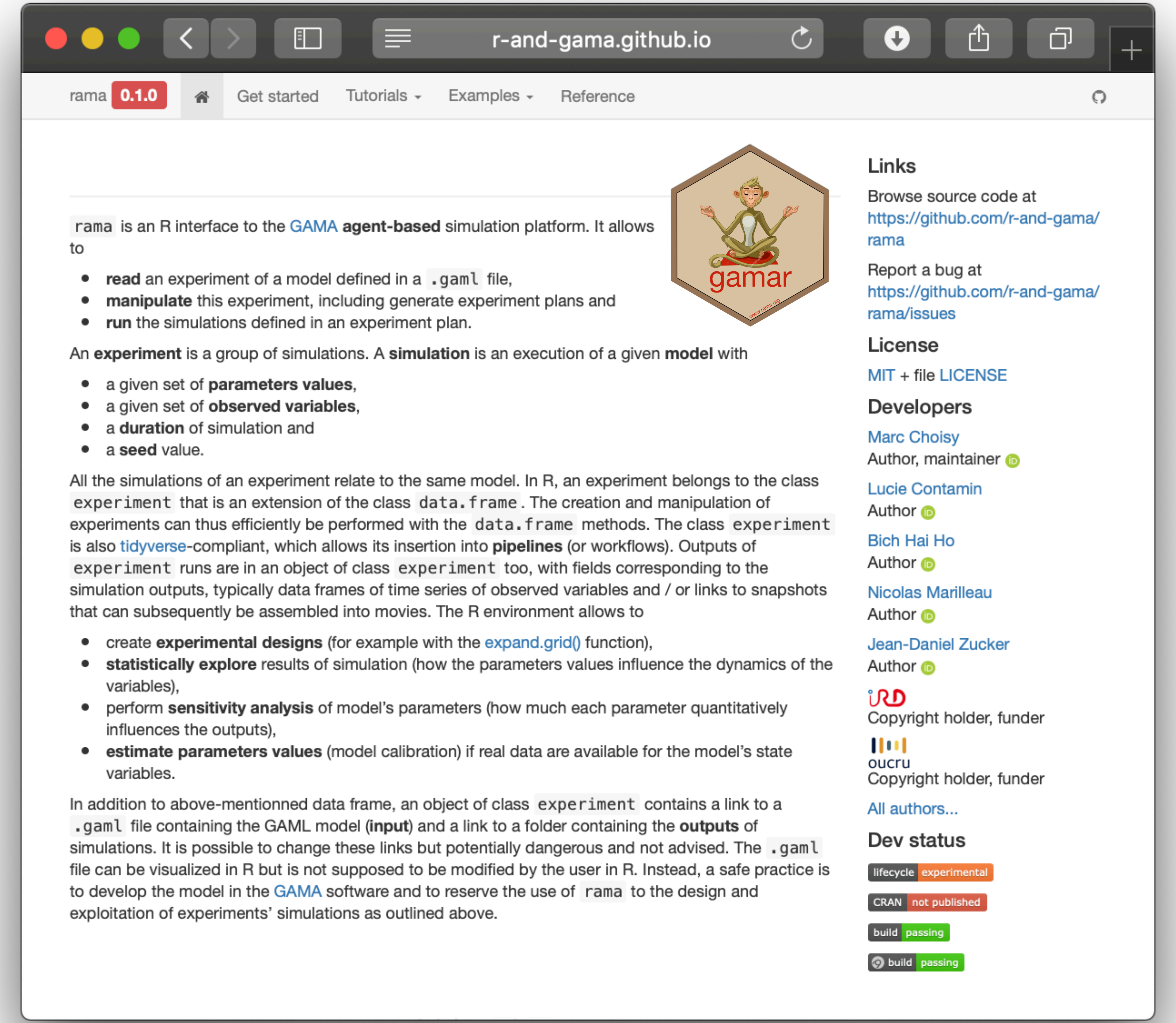
```
with(sir$output[[1]],
     plot(Step, r_I,
          ylab = "nb. infected")
```



```
path_to_movie <-
  make_movie(pp$output[[1]],
             "r_main_display")
```



https://r-and-gama.github.io/gamar



`rama` is an R interface to the GAMA **agent-based** simulation platform. It allows to

- **read** an experiment of a model defined in a `.gaml` file,
- **manipulate** this experiment, including generate experiment plans and
- **run** the simulations defined in an experiment plan.

An **experiment** is a group of simulations. A **simulation** is an execution of a given **model** with

- a given set of **parameters values**,
- a given set of **observed variables**,
- a **duration** of simulation and
- a **seed** value.

All the simulations of an experiment relate to the same model. In R, an experiment belongs to the class `experiment` that is an extension of the class `data.frame`. The creation and manipulation of experiments can thus efficiently be performed with the `data.frame` methods. The class `experiment` is also tidyverse-compliant, which allows its insertion into **pipelines** (or workflows). Outputs of `experiment` runs are in an object of class `experiment` too, with fields corresponding to the simulation outputs, typically data frames of time series of observed variables and / or links to snapshots that can subsequently be assembled into movies. The R environment allows to

- create **experimental designs** (for example with the expand.grid() function),
- **statistically explore** results of simulation (how the parameters values influence the dynamics of the variables),
- perform **sensitivity analysis** of model's parameters (how much each parameter quantitatively influences the outputs),
- **estimate parameters values** (model calibration) if real data are available for the model's state variables.

In addition to above-mentionned data frame, an object of class `experiment` contains a link to a `.gaml` file containing the GAML model (**input**) and a link to a folder containing the **outputs** of simulations. It is possible to change these links but potentially dangerous and not advised. The `.gaml` file can be visualized in R but is not supposed to be modified by the user in R. Instead, a safe practice is to develop the model in the GAMA software and to reserve the use of `rama` to the design and exploitation of experiments' simulations as outlined above.

Links

Browse source code at https://github.com/r-and-gama/rama

Report a bug at https://github.com/r-and-gama/rama/issues

License

MIT + file LICENSE

Developers

Marc Choisy
Author, maintainer

Lucie Contamin
Author

Bich Hai Ho
Author

Nicolas Marilleau
Author

Jean-Daniel Zucker
Author

IRD
Copyright holder, funder

oucru
Copyright holder, funder

All authors...

Dev status

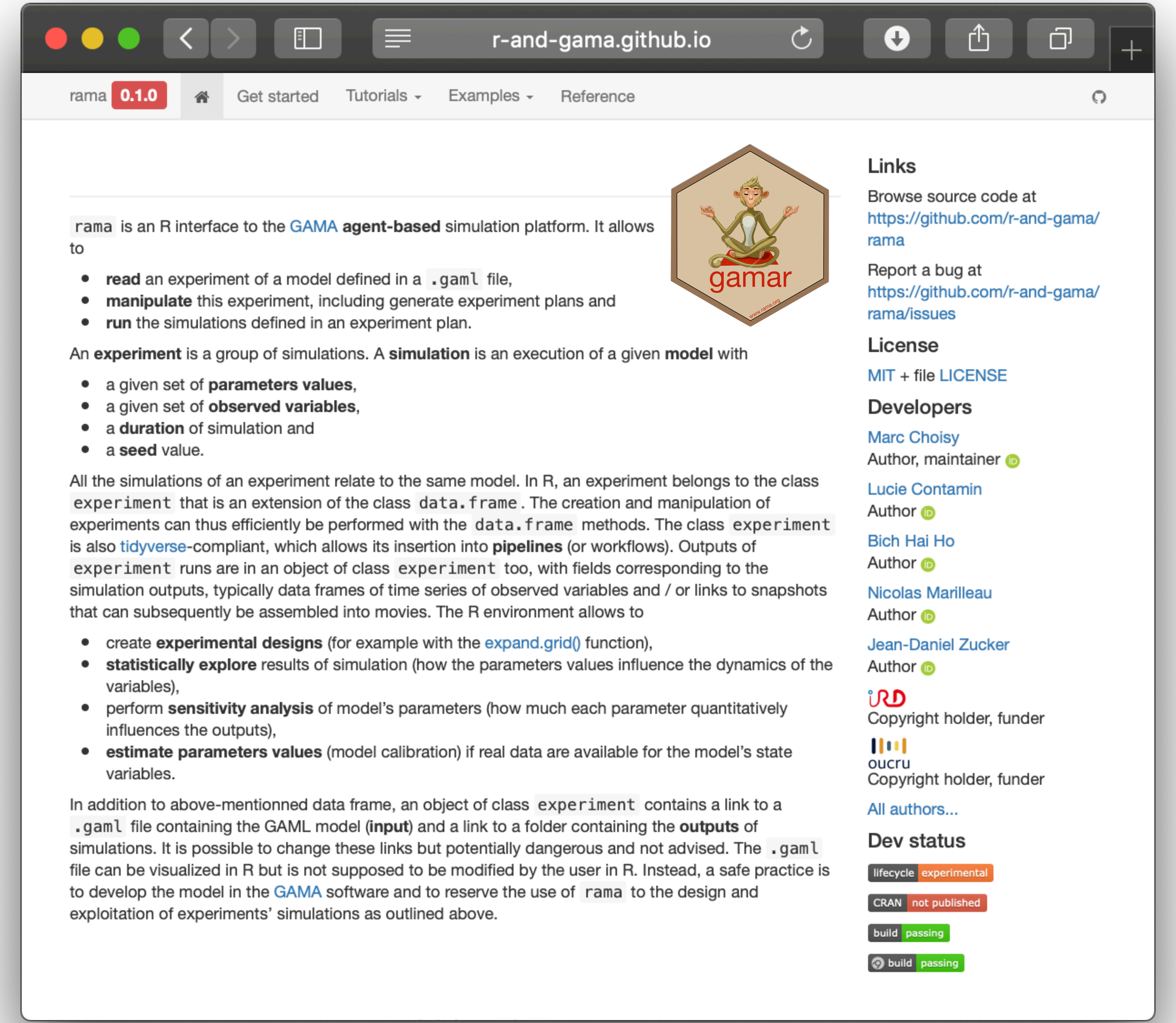
lifecycle experimental

CRAN not published

build passing

build passing

https://r-and-gama.github.io/gamar



`rama` is an R interface to the GAMA **agent-based** simulation platform. It allows to

- **read** an experiment of a model defined in a `.gaml` file,
- **manipulate** this experiment, including generate experiment plans and
- **run** the simulations defined in an experiment plan.

An **experiment** is a group of simulations. A **simulation** is an execution of a given **model** with

- a given set of **parameters values**,
- a given set of **observed variables**,
- a **duration** of simulation and
- a **seed** value.

All the simulations of an experiment relate to the same model. In R, an experiment belongs to the class `experiment` that is an extension of the class `data.frame`. The creation and manipulation of experiments can thus efficiently be performed with the `data.frame` methods. The class `experiment` is also tidyverse-compliant, which allows its insertion into **pipelines** (or workflows). Outputs of `experiment` runs are in an object of class `experiment` too, with fields corresponding to the simulation outputs, typically data frames of time series of observed variables and / or links to snapshots that can subsequently be assembled into movies. The R environment allows to

- create **experimental designs** (for example with the expand.grid() function),
- **statistically explore** results of simulation (how the parameters values influence the dynamics of the variables),
- perform **sensitivity analysis** of model's parameters (how much each parameter quantitatively influences the outputs),
- **estimate parameters values** (model calibration) if real data are available for the model's state variables.

In addition to above-mentionned data frame, an object of class `experiment` contains a link to a `.gaml` file containing the GAML model (**input**) and a link to a folder containing the **outputs** of simulations. It is possible to change these links but potentially dangerous and not advised. The `.gaml` file can be visualized in R but is not supposed to be modified by the user in R. Instead, a safe practice is to develop the model in the GAMA software and to reserve the use of `rama` to the design and exploitation of experiments' simulations as outlined above.

Links

Browse source code at
https://github.com/r-and-gama/rama

Report a bug at
https://github.com/r-and-gama/rama/issues

License

MIT + file LICENSE

Developers

Marc Choisy
Author, maintainer

Lucie Contamin
Author

Bich Hai Ho
Author

Nicolas Marilleau
Author

Jean-Daniel Zucker
Author

IRD
Copyright holder, funder

oucru
Copyright holder, funder

All authors...

Dev status

lifecycle experimental

CRAN not published

build passing

build passing