Package: epiworldRShiny (via r-universe)

September 12, 2024

Type Package

Title A 'shiny' Wrapper of the R Package 'epiworldR'

Version 0.1-0

Date 2024-05-31

Description R 'shiny' web apps for epidemiological Agent-Based Models. It provides a user-friendly interface to the Agent-Based Modeling (ABM) R package 'epiworldR' (Meyer et al., 2023) <DOI:10.21105/joss.05781>. Some of the main features of the package include the Susceptible-Infected-Susceptible (SIS), Susceptible-Infected-Recovered (SIR), and Susceptible-Exposed-Infected-Recovered (SEIR) models. 'epiworldRShiny' provides a web-based user interface for running various epidemiological ABMs, simulating interventions, and visualizing results interactively.

URL `https://github.com/UofUEpiBio/epiworldRShiny/`, `https://uofuepibio.github.io/epiworldRShiny/`,

BugReports `https://github.com/UofUEpiBio/epiworldRShiny/issues/`

License MIT + file LICENSE

RoxygenNote 7.3.1

Encoding UTF-8

Roxygen list(markdown = TRUE)

Suggests knitr

Imports shinyjs, utils, shinydashboard, DT, ggplot2, epiworldR, shinycssloaders, plotly

Depends shiny

Repository https://uofuepibio.r-universe.dev

RemoteUrl https://github.com/uofuepibio/epiworldrshiny

RemoteRef HEAD

RemoteSha a4da9edd91446860663e46a380d2bb3337aa7e1c

Contents

`epiworldRenv` *Access to the epiworldR environment.*

Description

This function is for internal use only.

Usage

```
epiworldRenv()
```

Value

Returns the `epiworldR_env` environment.

`epiworldRShiny` *epiworldRShiny App Launcher*

Description

Fires up the R Shiny App. You can find more examples and documentation at the package's website: `https://UofUEpiBio.github.io/epiworldRShiny/`.

`run_app` is a wrapper for the `epiworldRShiny` function. It is a convenience function to run the app.

Usage

```
epiworldRShiny(...)

run_app(...)
```

Arguments

`...`	Currently ignored.

Value

Loads and opens the R shiny app for the epiworldR package

`epiworldrshiny-ui` *epiworldRShiny UI builder functions*

Description

All functions in this section are for internal use only. They are used to build the UI for the epiworldRShiny app.

Usage

```
text_input_disease_name(model_name)

slider_prevalence(model_name)

numeric_input_ndays(model_name)

slider_input_rate(model_name, rate_name, value, maxval = 1, input_label = NULL)

network_input(model_name)

npis_input(model_name)

seed_input(model_name)

models_setup()

population_input(model_name)

simulate_button(model_name)
```

Arguments

`model_name`	Name of the epiworldR model.
`rate_name`	Name of the rate.
`value`	Initial value for the slider.
`maxval`	Maxiumum value for the slider.
`input_label`	Aids in creating the appropriate slider name.

Value

- Unless otherwise specified, returns an object of class shiny.tag.
- `network_input` returns an object of class shiny::tagList (`shiny.tag.list`).
- `npis_input` returns an object of class shiny::tagList (`shiny.tag.list`).
- `models_setup` returns an object of class list.
- `population_input` returns an object of class shiny.tag.list.

Examples

```
text_input_disease_name("SEIRD")
slider_prevalence("SEIRD")
numeric_input_ndays("SEIRD")
slider_input_rate("SEIRD", "transmission", value = 0.3, maxval = 1,
input_label = NULL)
network_input("SEIRD")
npis_input("SEIRD")
seed_input("SEIRD")
models_setup()
population_input("SEIRD")
simulate_button("SEIRD")
```

`find_scale` *Find the right plotting-scale*

Description

This function determines the scale of the y-axis for plot_epi.

Usage

```
find_scale(x)
```

Arguments

`x`	The maximum value found in the model state counts

Value

An integer representing the scale for the y-axis. A max counts value of 10000 will return a scale of 1, 100000 will return a scale of 1000, 1000000 will return a scale of 10000.

See Also

Other Server side functions: `plot_epi()`, `plot_reproductive_epi()`, `pop_generator()`

Examples

```
find_scale(100000)
```

`interventions_add_all` *NPI adding function*

Description

NPI adding function

Usage

```
interventions_add_all(model, modelname, input)
```

Arguments

`model`	epiworldR model.
`modelname`	Specified model.
`input`	User epiworldR model selection.

Value

Returns an object of class epiworld_model, where model is substituted with the model name.

See Also

Other interventions: `npi_add_masking()`, `npi_add_school_closure()`, `pi_add_vaccine()`

`npi_add_masking` *Masking intervention (non-pharmaceutical intervention)*

Description

Masking intervention (non-pharmaceutical intervention)

Usage

```
npi_add_masking(model, preval, transmission_reduction)
```

Arguments

`model`	epiworldR model.
`preval`	Prevalence of masking within the population.
`transmission_reduction`	Reduction in transmission probability due to masking

Value

Returns an object of class epiworld_model, where model is substituted with the model name.

See Also

Other interventions: `interventions_add_all()`, `npi_add_school_closure()`, `pi_add_vaccine()`

Examples

```
library(epiworldR) # for ModelSEIRCONN function
model <- ModelSEIRCONN("COVID-19", n = 1000, prevalence = 0.05,
                       contact_rate = 4, transmission_rate = 0.1,
                       incubation_days = 7, recovery_rate = 0.14)
run(model, ndays = 100, seed = 123)
npi_add_masking(model, preval = .8, transmission_reduction = .7)
```

`npi_add_school_closure` *School closure intervention (non-pharmaceutical intervention)*

Description

School closure intervention (non-pharmaceutical intervention)

Usage

```
npi_add_school_closure(model, preval, day, transmission_reduction)
```

Arguments

`model`	epiworldR model.
`preval`	Prevalence of school closure within the population.
`day`	Day in the simulation where school closure goes into effect.
`transmission_reduction`	Reduction in transmission probability due to school closure.

Value

Returns an object of class epiworld_model, where model is substituted with the model name.

See Also

Other interventions: `interventions_add_all()`, `npi_add_masking()`, `pi_add_vaccine()`

Examples

```
library(epiworldR) # for ModelSEIRCONN function
model <- ModelSEIRCONN("COVID-19", n = 1000, prevalence = 0.05,
                        contact_rate = 4, transmission_rate = 0.1,
                        incubation_days = 7, recovery_rate = 0.14)
run(model, ndays = 100, seed = 123)
npi_add_school_closure(model, preval = .8, transmission_reduction = .3, day = 10)
```

`pi_add_vaccine` *Vaccination (pharmaceutical intervention)*

Description

Vaccination (pharmaceutical intervention)

Usage

```
pi_add_vaccine(
  model,
  preval,
  susceptibility_reduction,
  transmission_reduction,
  recovery_enhancer,
  death_reduction
)
```

Arguments

`model`	epiworldR model.
`preval`	Initial prevalence of the vaccine.
`susceptibility_reduction`	Reduction in susceptibility probability due to vaccine.
`transmission_reduction`	Reduction in transmission probability due to vaccine.
`recovery_enhancer`	Probability increase in recovery due to vaccine.
`death_reduction`	Reduction in death probability due to vaccine.

Value

Returns an object of class epiworld_model, where model is substituted with the model name.

See Also

Other interventions: `interventions_add_all()`, `npi_add_masking()`, `npi_add_school_closure()`

Examples

```
library(epiworldR) # for ModelSEIRCONN function
model <- ModelSEIRCONN("COVID-19", n = 1000, prevalence = 0.05,
                       contact_rate = 4, transmission_rate = 0.1,
                       incubation_days = 7, recovery_rate = 0.14)
run(model, ndays = 100, seed = 123)
pi_add_vaccine(model, preval = .8, susceptibility_reduction = .9,
                transmission_reduction = .5, recovery_enhancer = .5,
                death_reduction = .9)
```

`plot_epi` *Plot daily incidence*

Description

This function generates a plot of the model states over time

Usage

```
plot_epi(model, mark_max)
```

Arguments

`model`	The number of individuals in the population.
`mark_max`	The state which will have a mark at the peak

Value

A plot displaying each state from the model over the course of the simulation

See Also

Other Server side functions: `find_scale()`, `plot_reproductive_epi()`, `pop_generator()`

Examples

```
library(epiworldR) # for ModelSEIRCONN function
model <- ModelSEIRCONN("COVID-19", n = 1000, prevalence = 0.05,
                       contact_rate = 4, transmission_rate = 0.1,
                       incubation_days = 7, recovery_rate = 0.14)
run(model, ndays = 100, seed = 123)
plot_epi(model, mark_max = "Infected")
```

`plot_reproductive_epi` *Plot Rt*

Description

This function generates a plot of the reproductive number over time

Usage

```
plot_reproductive_epi(model)
```

Arguments

`model`	The model object

Value

A plot displaying the reproductive number for the model over the course of the simulation

See Also

Other Server side functions: `find_scale()`, `plot_epi()`, `pop_generator()`

Examples

```
library(epiworldR) # for ModelSEIRCONN function
model <- ModelSEIRCONN("COVID-19", n = 1000, prevalence = 0.05,
                       contact_rate = 4, transmission_rate = 0.1,
                       incubation_days = 7, recovery_rate = 0.14)
run(model, ndays = 100, seed = 123)
plot_reproductive_epi(model)
```

`pop_generator` *Population generator*

Description

This function generates a population matrix with specified characteristics.

Usage

```
pop_generator(
  n,
  prop_hispanic = 0.5,
  prop_female = 0.5,
  prop_19_59_60plus = c(0.3, 0.6)
)
```

Arguments

`n`	The number of individuals in the population.
`prop_hispanic`	The proportion of individuals who are Hispanic. Default is 0.5.
`prop_female`	The proportion of individuals who are female. Default is 0.5.
`prop_19_59_60plus`	A vector of length 3 representing the proportions of individuals in the age groups 0-19, 20-59, and 60+. Default is c(0.5, 0.3, 0.2).

Value

A matrix representing the generated population, with columns for age groups (0-19, 20-59, 60+), NotHispanic, and Female.

See Also

Other Server side functions: `find_scale()`, `plot_epi()`, `plot_reproductive_epi()`

Examples

```
pop_generator(n = 1000, prop_hispanic = .5, prop_female = .5,
              prop_19_59_60plus = c(.3, .6))
```

Index