Figures - CM AM

Shannon Gallagher

May 15, 2018

To download and load the package `simCAM` (Simulations of Compartment and Agent-based Models), run

```
devtools::install_github("shannong19/simCAM")
library(simCAM)
```

You will need the following libraries for plotting:

```
library(ggplot2)
library(RColorBrewer)
library(reshape2)
library(gridExtra)
```

SIR-system

To run the stochastic SIR-CM, run the following commands. First run the deterministic SIR-CM to obtain the probabilities of transition for a given set of parameters.

```
## Parameters to run model
T <- 100 # number of time steps
init_vals <- c(950, 50, 0) #initial values
beta <- .1
gamma <- .03
N <- sum(init_vals)
step <- 1 # time step in deterministic SIR-CM
results <- SIR(T, init_vals, beta,
                   gamma, step, SIR_inner, do_plot = FALSE)

## Extract the probabilities of transition
probs <- extract_probs_sir(results, beta, gamma)
```

Next, run the stochastic SIR-CM. We have set the number of simulations/runs equal to $L = 50$. In the manuscript, we use 5000 runs.

```
L <- 50 # number of runs

## Set up of initial values
S <- matrix(0, nrow = T + 1, ncol = L)
I <- matrix(0, nrow = T + 1, ncol = L)
R <- matrix(0, nrow = T + 1, ncol = L)
cm_sim_list <- list(S = S, I = I, R = R)
cm_sim_list <- fill_inits_sir(init_vals, cm_sim_list)


## Run the simulation
cm_sim_list <- run_cm_sir(T, init_vals,
probs, L, cm_sim_list)
counts <- cm_sim_list$S
```

Then, run the stochastic SIR-AM, also with $L = 50$ simulations. The function `summarize_agents()` puts the output agents into a format that is identical to stochastic SIR-CM output.

```
## Initialize
L <- 50
T <- 100
agents <- vector("list", L)
agents <- initialize_agents(T, N, init_vals, agents)

## Run the AM
out_agents <- run_am_sir(T, probs, L, agents)

## Summarize the agents
agents_sims <- summarize_agents(out_agents)
am_sim_list <- agents_sims
```

We then create our first set of figures. Figure (1) is the mean proportion of agents in each state for a given time for the SIR-system.

```
## Average
cm_mean <- summary_sims(cm_sim_list, var_names = c("S", "I", "R")) * 100 / N
am_mean <- summary_sims(am_sim_list, var_names = c("S", "I", "R")) * 100 / N
g <- plot_overlap(cm_mean, am_mean, plot_dash = TRUE,
             size = 2, L=L)
```

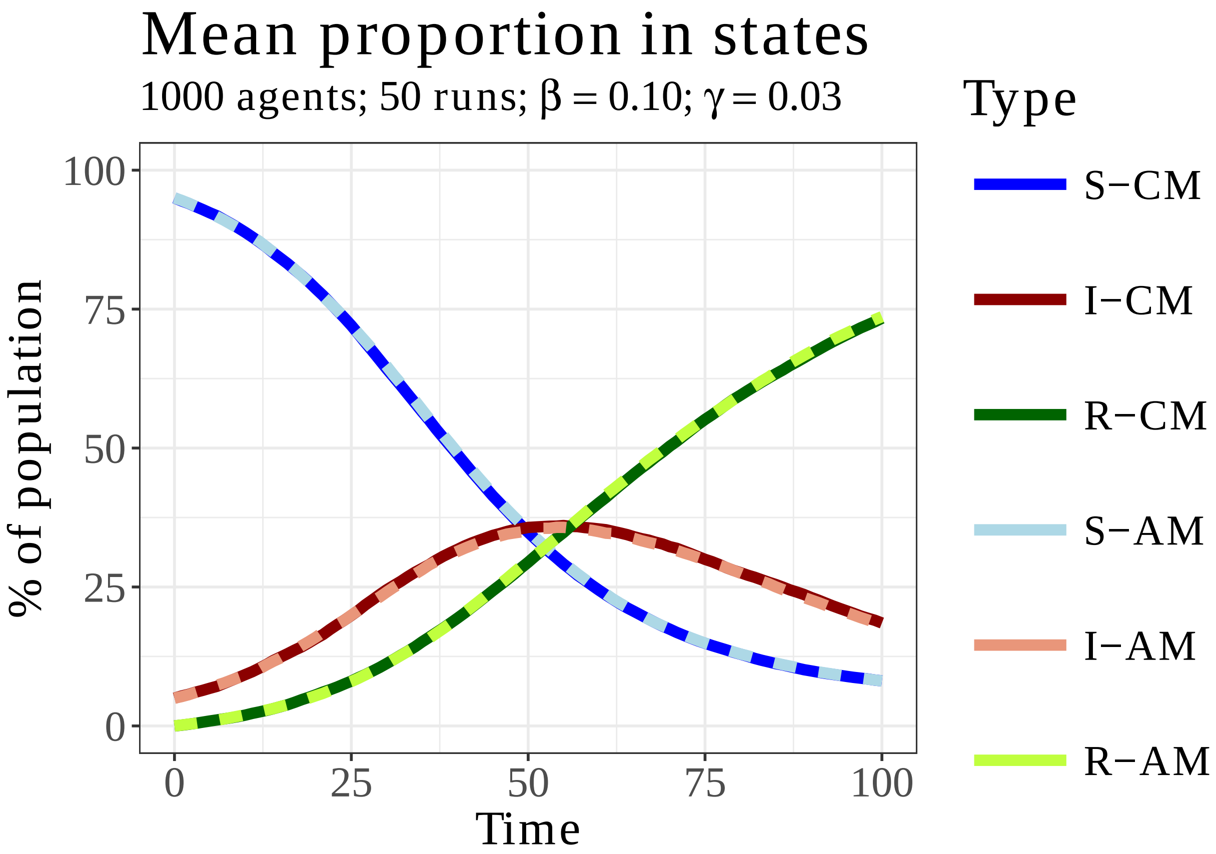


Figure (2) is the variance of of the agents in each state for a given time for the SIR-system.

```
## Variance
## Plotting the variance
```

```
am_var <- summary_sims(am_sim_list,
  fxn = rowVar,
var_names = c("S", "I", "R"))
cm_var <- summary_sims(cm_sim_list,
  fxn = rowVar,
  var_names = c("S", "I", "R"))
g <- plot_overlap(cm_var, am_var,
  summary_name = "Variance in states",
  ylim = c(0, max(max(cm_var), max(am_var))),
  plot_dash = FALSE,
  size = 2)
```

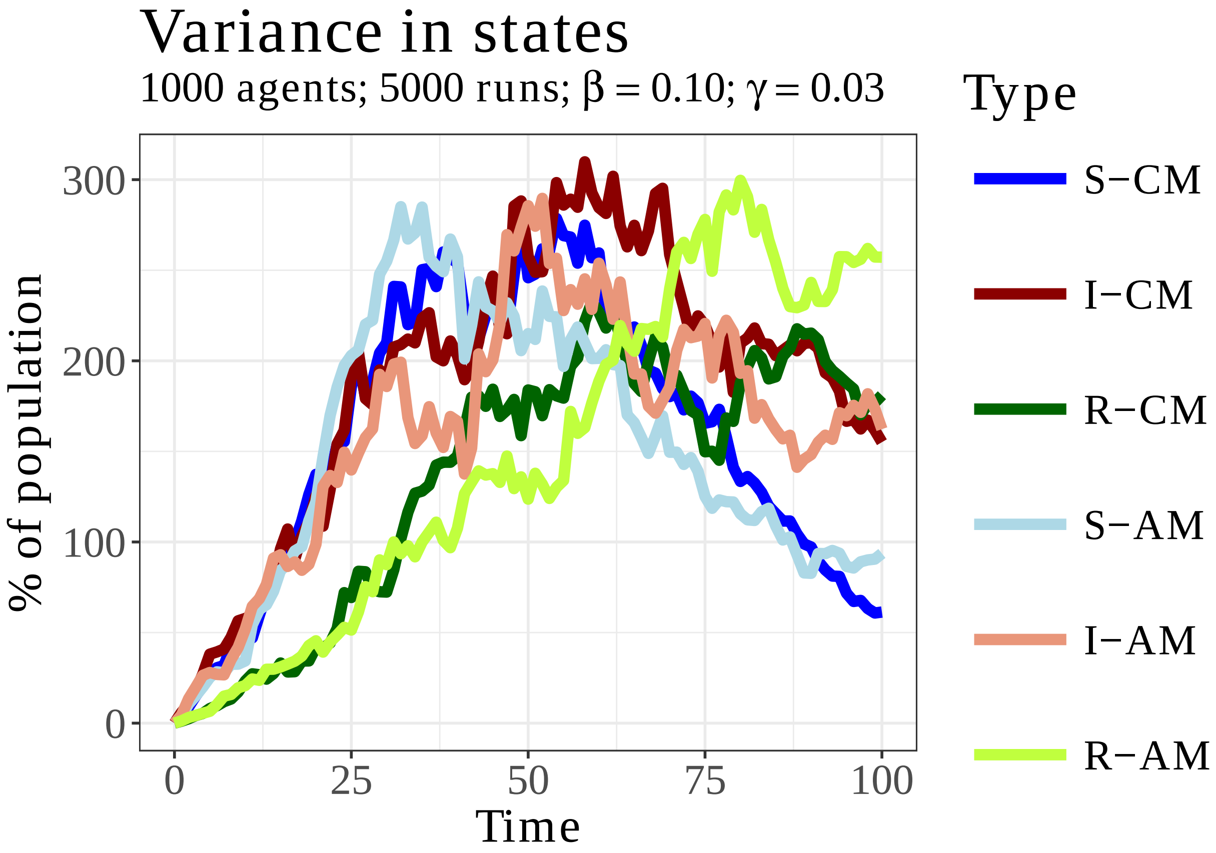
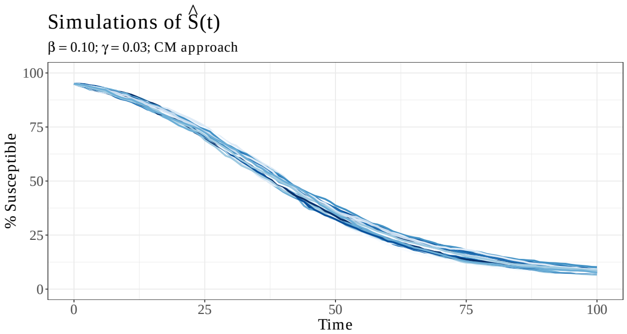
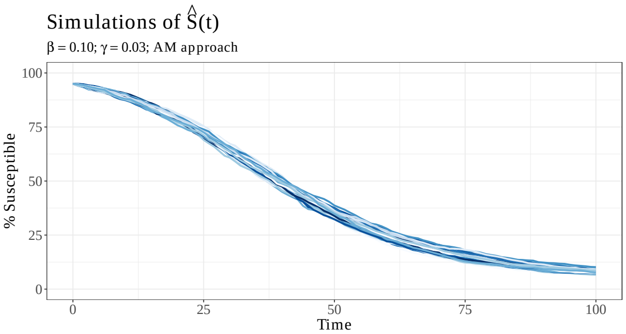


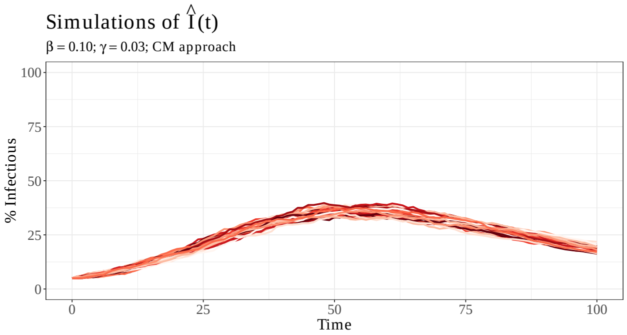
Figure (3) is the actual simulations (sample paths) for each state, for both the CM and AM.

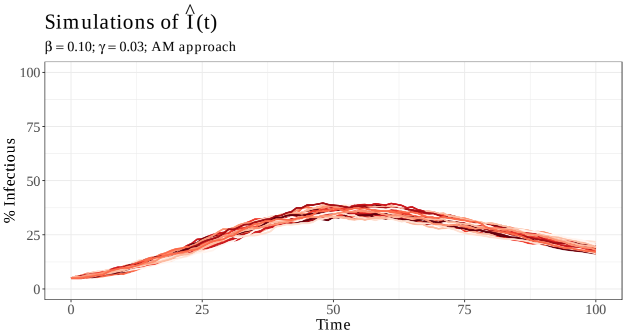
```
cols <- c("Blues", "Reds", "Greens")
cm_titles <- c("Susceptible",
"Infectious",
"Recovered")
cm_symbols <- c("\\hat{S}(t)",
"\\hat{I}(t)",
"\\hat{R}")
g_list_cm <- lapply(1:length(cm_sim_list),
function(ind) {
plot_draws_sir(
cm_sim_list[[ind]],
beta,
gamma,
```

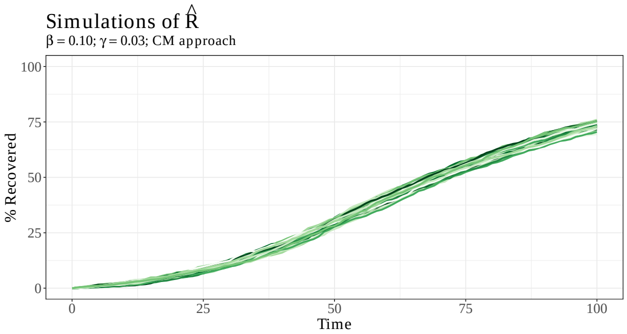
```r
N,
L,
col = cols[ind],
cat_title = cm_titles[ind],
tex_symbol = cm_symbols[ind],
approach = "CM"
)
})


g_list_am <- lapply(1:length(am_sim_list),
function(ind) {
plot_draws_sir(
cm_sim_list[[ind]],
beta,
gamma,
N,
L,
col = cols[ind],
cat_title = cm_titles[ind],
tex_symbol = cm_symbols[ind],
approach = "AM"
)
})

## Plot on a grid
do.call("grid.arrange", c(g_list_cm, g_list_am, ncol=2, as.table = FALSE))
```

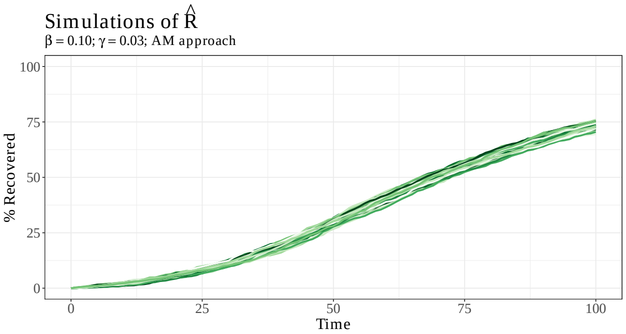










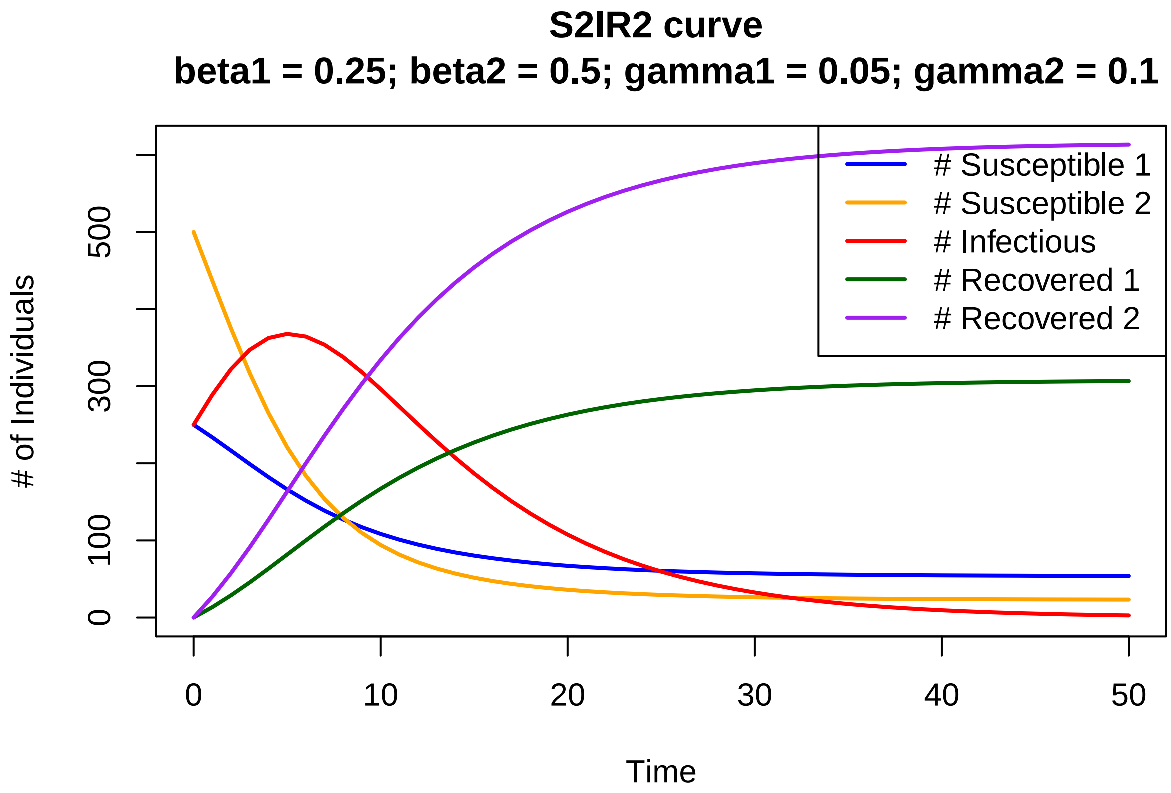


S^2IR^2-system

We also have similar simulations for the S^2IR^2-system.

We first extract the probabilities from the deterministic S^2IR^2-CM with the following ste of initial parameters.

```
T <- 50
init_vals <- c(250, 500, 250, 0, 0)
N <- sum(init_vals)
beta1 <- .25
beta2 <- .5
gamma1 <- .05
gamma2 <- .1
step <- 1
inner_fxn <- SIR2_inner
results <- SIR2(T, init_vals, beta1,
  beta2, gamma1, gamma2)
```



```
## Extract the probabilities of transition
p <- extract_probs(results)
```

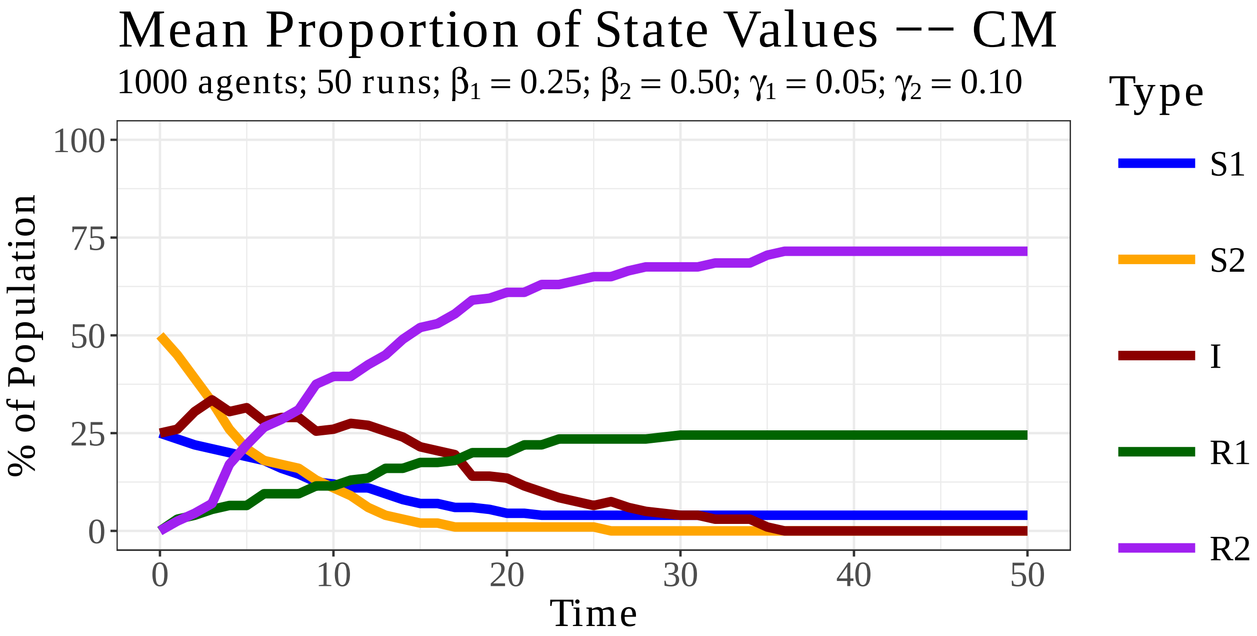
We then run the stochastic S^{2}IR^2-CM and stochastic S^{2}IR^2-AM.

```
## CM
L <- 50 # number of runs
S1 <- matrix(0, nrow = T + 1, ncol = L)
S2 <- matrix(0, nrow = T + 1, ncol = L)
I <- matrix(0, nrow = T + 1, ncol = L)
R1 <- matrix(0, nrow = T + 1, ncol = L)
R2 <- matrix(0, nrow = T + 1, ncol = L)
cm_sim_list <- list(
S1 = S1,
S2 = S2,
I = I,
R1 = R1,
R2 = R2
)
cm_sim_list <- fill_in_init_vals(init_vals, cm_sim_list)
cm_sim_list <- run_cm(T, init_vals, p, L = L, cm_sim_list)

## AM
L <- 50
T <- 50
agents <- vector("list", L)
agents <- initialize_agents(T, N, init_vals, agents)

out_agents <- run_am(T, p, L, agents)
```

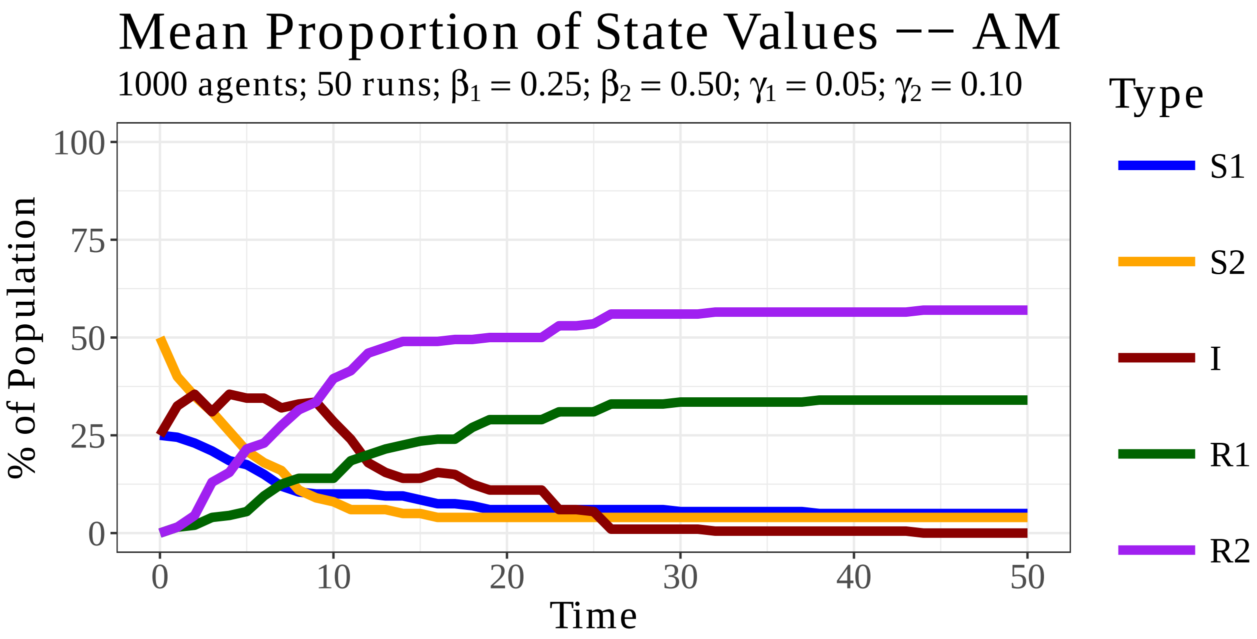
```
## Put agents in same form as CM output
agents_sims <- summarize_agents(out_agents, n_states = 5)
am_sim_list <- agents_sims
```

We then make our analagous plots of the first three figures, Figures (5)-(7).

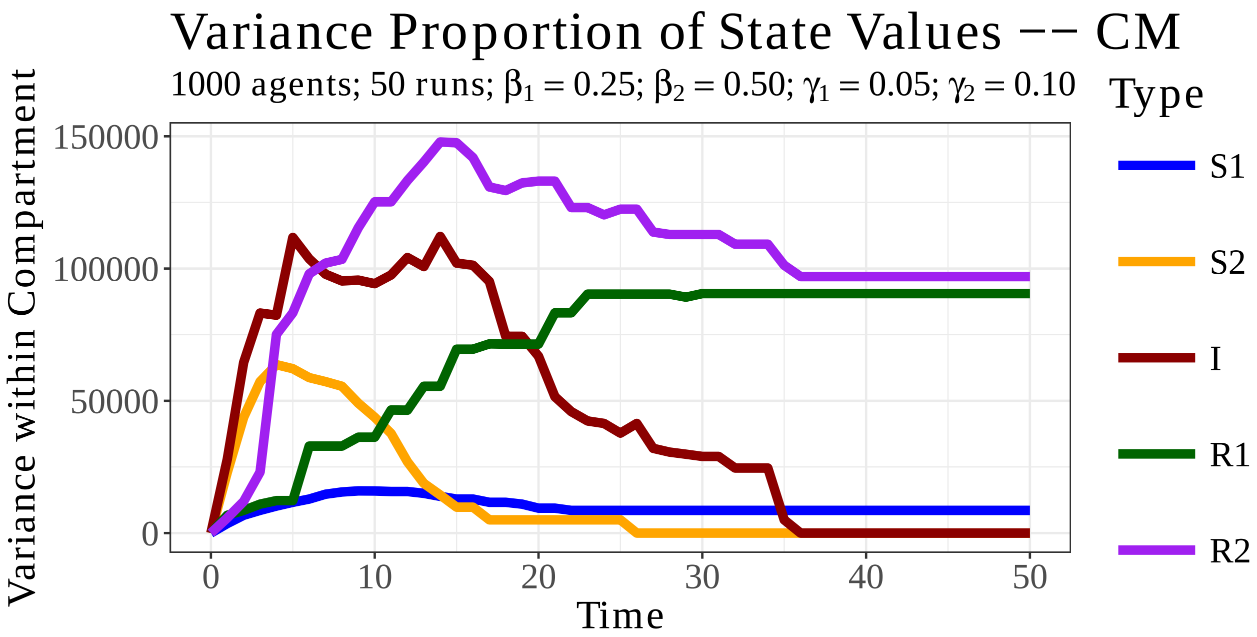
```
## Mean
## CM
cm_mean <- summary_sims(cm_sim_list)
g_avg_cm <-
plot_summary(
cm_mean,
L = L,
beta1 = beta1,
beta2 = beta2,
gamma1 = gamma1,
gamma2 = gamma2
)
g_avg_cm
```



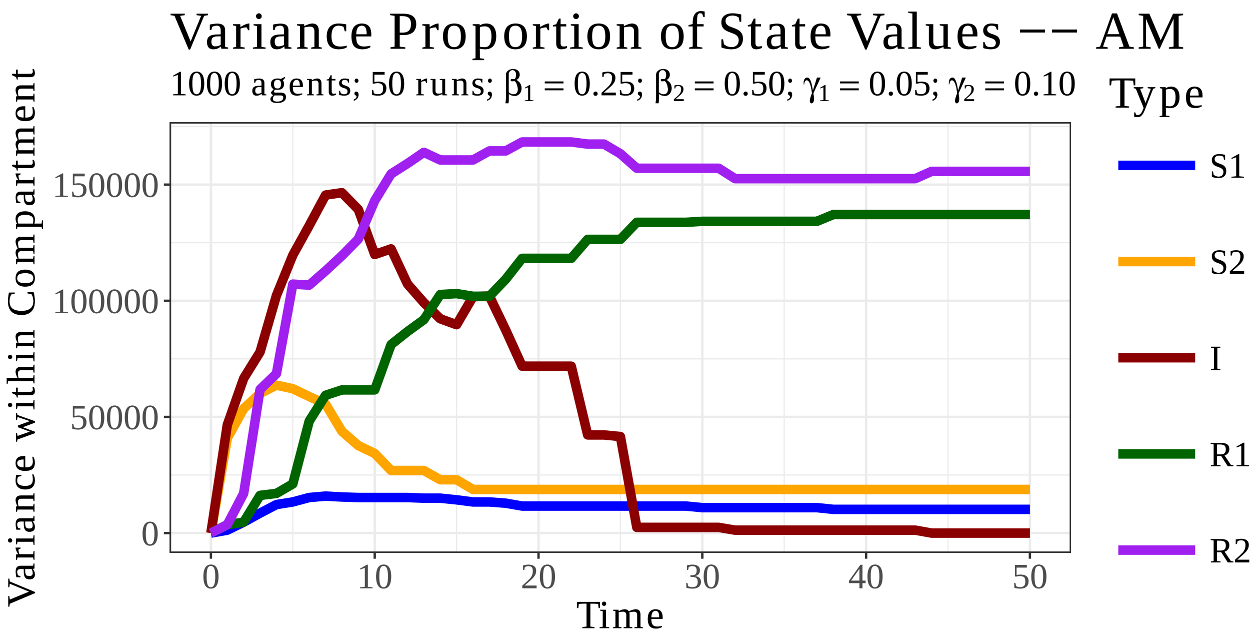
```
## AM
am_mean <- summary_sims(am_sim_list)


g_avg_am <- plot_summary(
am_mean,
approach = "AM",
L = L,
beta1 = beta1,
gamma1 = gamma1,
beta2 = beta2,
gamma2 = gamma2
)
g_avg_am
```



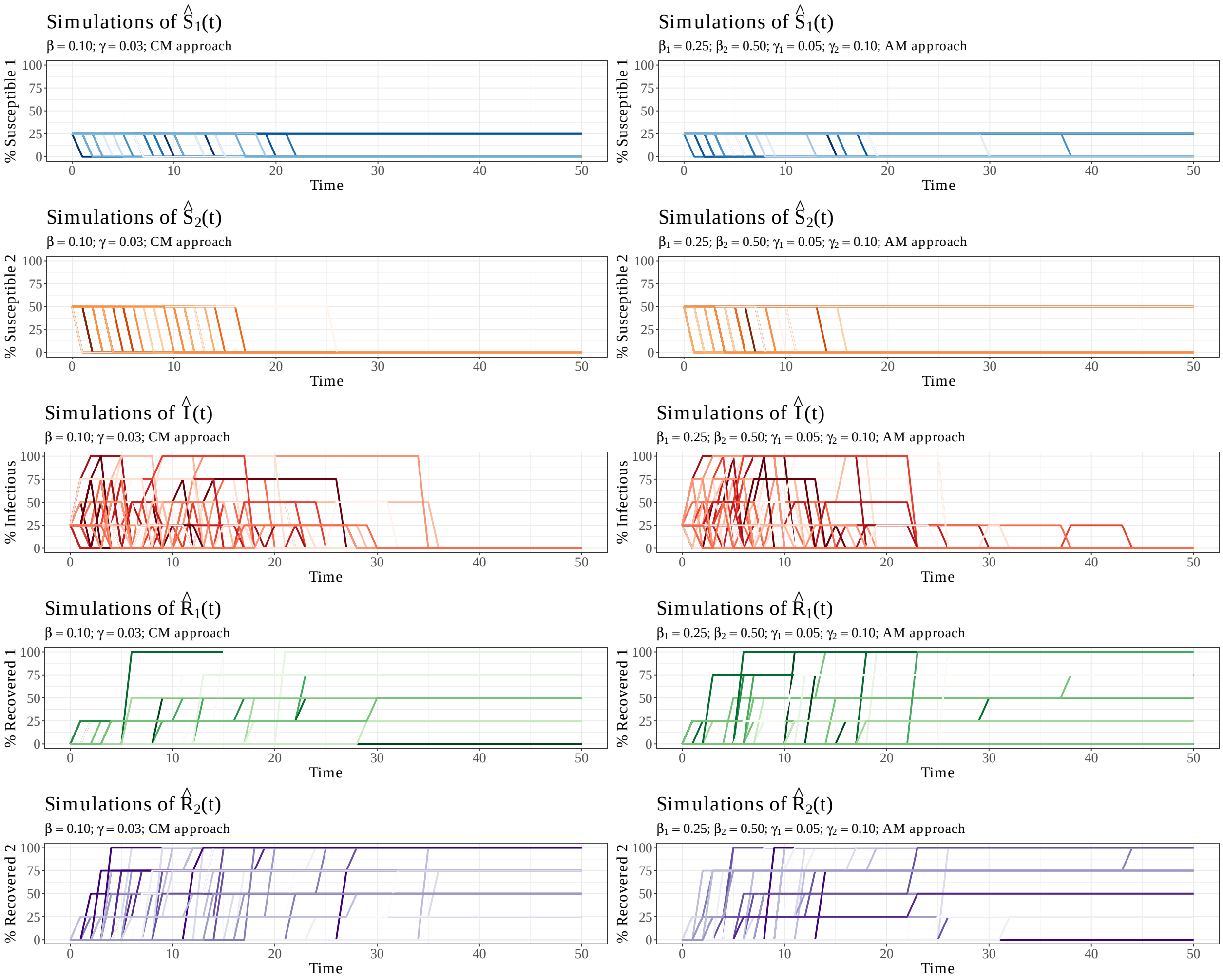
```
## Variance


## CM
cm_var <- summary_sims(cm_sim_list, fxn = rowVar)
g_var_cm <-
plot_summary(
cm_var,
sum_name = "Variance Proportion",
ylab = "Variance within Compartment",
L = L,
beta1 = beta1,
gamma1 = gamma1,
beta2 = beta2,
gamma2 = gamma2
)
g_var_cm
```



```
## AM
am_var <- summary_sims(am_sim_list, fxn = rowVar)
g_var_am <-
plot_summary(
am_var,
sum_name = "Variance Proportion",
ylab = "Variance within Compartment",
approach = "AM",
L = L,
beta1 = beta1,
gamma1 = gamma1,
beta2 = beta2,
gamma2 = gamma2
)
g_var_am
```



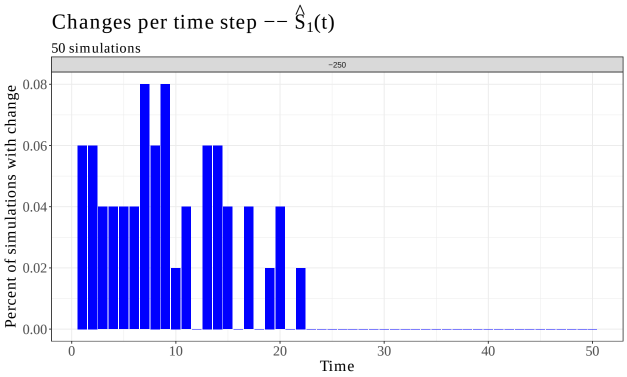
```
## Plotting all the draws
cols <- c("Blues", "Oranges", "Reds", "Greens", "Purples")
cm_titles <- c("Susceptible 1",
"Susceptible 2",
"Infectious",
"Recovered 1",
"Recovered 2")
cm_symbols <- c("\\hat{S}_1(t)",
"\\hat{S}_2(t)",
"\\hat{I}(t)",
"\\hat{R}_1(t)",
"\\hat{R}_2(t)")

g_list_cm <- lapply(1:length(cm_sim_list),
                    function(ind) {
                    plot_draws_sir(
                    cm_sim_list[[ind]],
                    beta,
                    gamma,
                    N,
                    L,
                    col = cols[ind],
                    cat_title = cm_titles[ind],
                    tex_symbol = cm_symbols[ind],
                    approach = "CM"
                    )
                    })

g_list_am <- lapply(1:length(am_sim_list),
function(ind) {
plot_draws_s2ir2(
am_sim_list[[ind]],
beta1,
beta2,
gamma1,
gamma2,
N,
L,
col = cols[ind],
cat_title = cm_titles[ind],
tex_symbol = cm_symbols[ind],
approach = "AM"
)
})

## Plotting
do.call("grid.arrange", c(g_list_cm, g_list_am, ncol=2, as.table = FALSE))
```

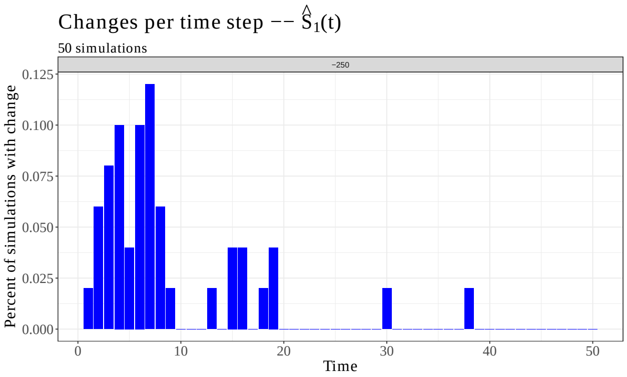


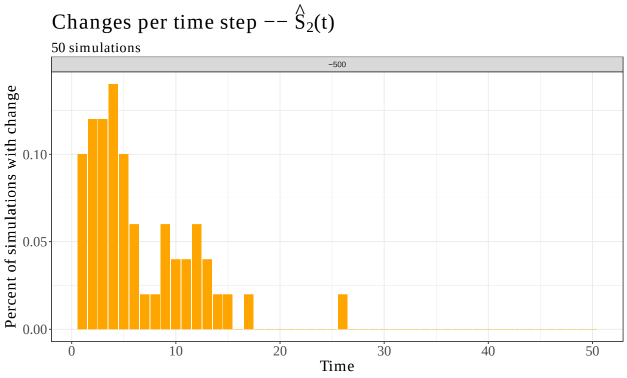
Time Distributions

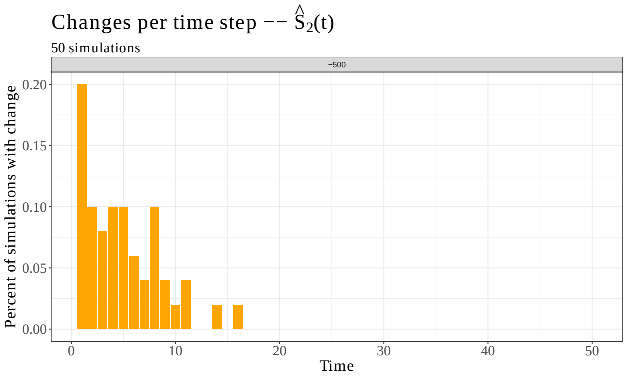
Figures (8)-(11) pertain to transitions from compartments over time. These graphs are produced with the following.

```
cols <- c("blue", "orange", "darkred", "darkgreen", "purple")
cm_titles <- c("Susceptible 1",
"Susceptible 2",
"Infectious",
"Recovered 1",
"Recovered 2")
cm_symbols <- c("\\hat{S}_1(t)",
"\\hat{S}_2(t)",
"\\hat{I}(t)",
"\\hat{R}_1(t)",
"\\hat{R}_2(t)")
g_list_cm_times <- lapply(1:length(cm_sim_list),
function(ind) {
plot_time_dist(
cm_sim_list[[ind]],
beta1,
beta2,
gamma1,
gamma2,
N,
L,
col = cols[ind],
cat_title = cm_titles[ind],
tex_symbol = cm_symbols[ind]
)
})

g_list_am_times <- lapply(1:length(am_sim_list),
function(ind) {
plot_time_dist(
am_sim_list[[ind]],
beta1,
beta2,
gamma1,
gamma2,
N,
L,
col = cols[ind],
cat_title = cm_titles[ind],
tex_symbol = cm_symbols[ind]
)
})


## Arrange
grid.arrange(g_list_cm_times[[1]],
g_list_am_times[[1]],
g_list_cm_times[[2]],
g_list_am_times[[2]],
ncol = 2)
```

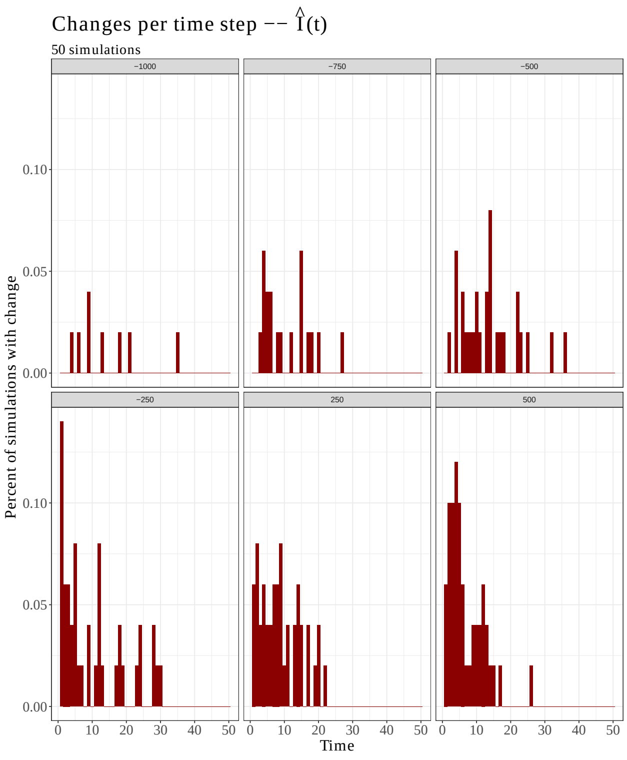


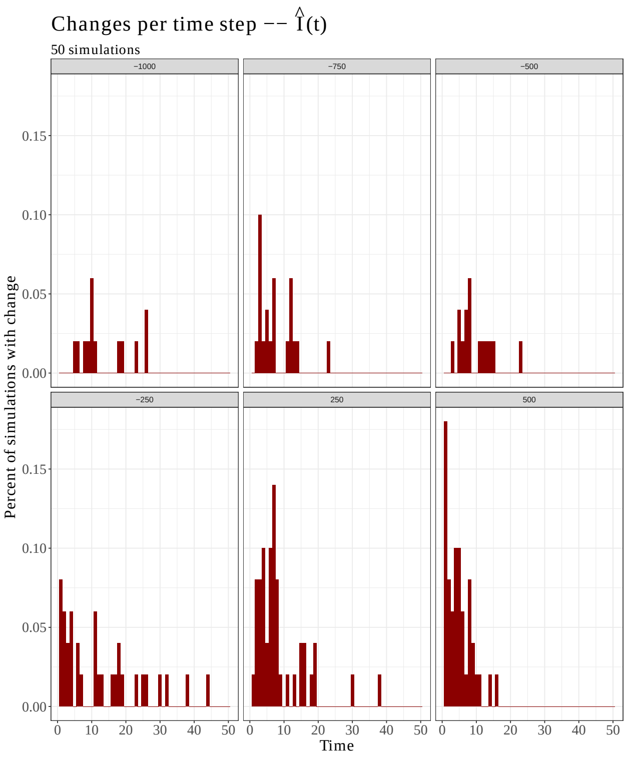




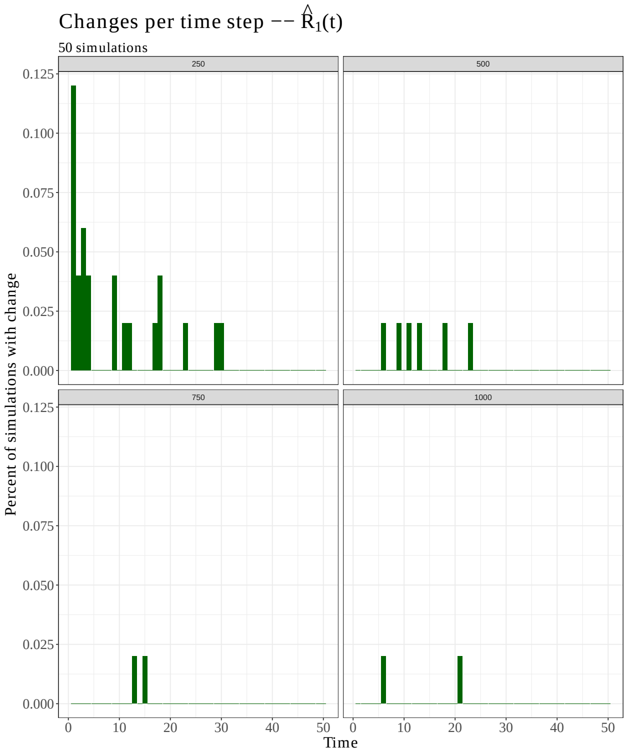


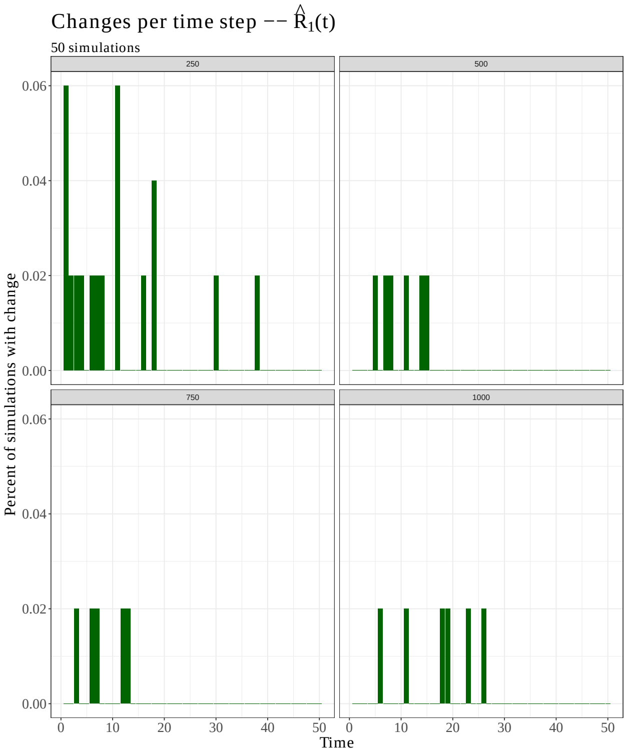
```
## I
grid.arrange(g_list_cm_times[[3]], g_list_am_times[[3]],
ncol = 2)
```



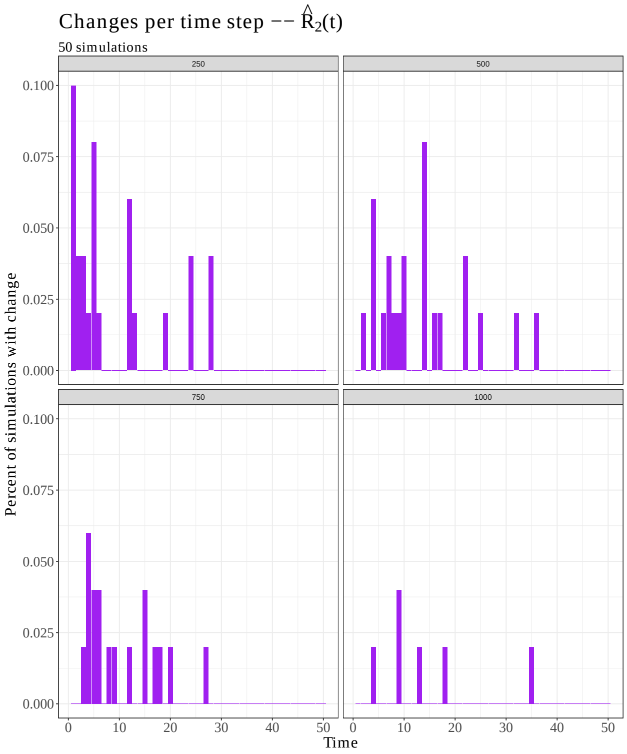


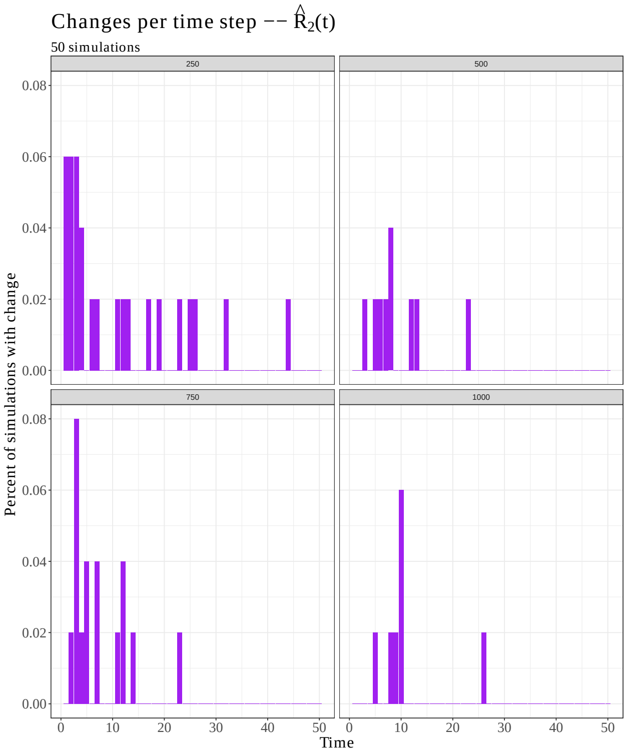
```
## R1
grid.arrange(g_list_cm_times[[4]], g_list_am_times[[4]],
ncol = 2)
```





```
## R2
grid.arrange(g_list_cm_times[[5]], g_list_am_times[[5]],
ncol = 2)
```


Changes per time step -- $\hat{R}_2(t)$
50 simulations
Percent of simulations with change
Time


Changes per time step -- $\hat{R}_2(t)$
50 simulations
Percent of simulations with change
Time