DES Replication and Output Visualization

Alireza Sheikh-Zadeh, PhD

Replication

To run a DES model, you must indicate the required number of observations. Since each run of the model produces entities for a limited simulated clock (e.g., one day), you need to specify the number of times you want Simmer to run the model (e.g., multiple days). Note that each replication must be independent of other replicas. Thus, the observations across replications form a random sample of the observations. Therefore, we can study and visualize the statistics across the replications.

Example: Replicate the clinic simulation 30 times.

```r
library(simmer)
set.seed(123)

# Set up a trajectory for a patient
patient <- trajectory("patients' path") %>%

  seize("nurse", 1) %>%
  timeout(function() rnorm(1, 15, 1)) %>%
  release("nurse", 1) %>%

  seize("doctor", 1) %>%
  timeout(function() rnorm(1, 20, 1)) %>%
  release("doctor", 1) %>%

  seize("administration", 1) %>%
  timeout(function() rnorm(1, 5, 1)) %>%
  release("administration", 1)

# It is easy to replicate a simulation multiple times using standard lapply R
functions, which is like a for-loop.
envs <- lapply(1:30, function(i) {
  simmer("Clinic") %>%
  add_resource("nurse", 2) %>%
  add_resource("doctor", 3) %>%
  add_resource("administration", 1) %>%
  add_generator("patient", patient, function() rnorm(1, 5, 0.5)) %>%
  run(540)
})

# We can look at the result of one replication, e.g. the 5th replication
envs[[5]]
```

```
## simmer environment: Clinic | now: 540 | next: 540.056134628675
## { Monitor: in memory }
## { Resource: nurse | monitored: TRUE | server status: 2(2) | queue status:
36(Inf) }
## { Resource: doctor | monitored: TRUE | server status: 3(3) | queue status:
0(Inf) }
## { Resource: administration | monitored: TRUE | server status: 1(1) | queue
status: 0(Inf) }
## { Source: patient | monitored: 1 | n_generated: 109 }
```

```
envs[[6]]
```

```
## simmer environment: Clinic | now: 540 | next: 540.063821848268
## { Monitor: in memory }
## { Resource: nurse | monitored: TRUE | server status: 2(2) | queue status:
34(Inf) }
## { Resource: doctor | monitored: TRUE | server status: 3(3) | queue status:
0(Inf) }
## { Resource: administration | monitored: TRUE | server status: 1(1) | queue
status: 0(Inf) }
## { Source: patient | monitored: 1 | n_generated: 107 }
```

Basic visualization tools

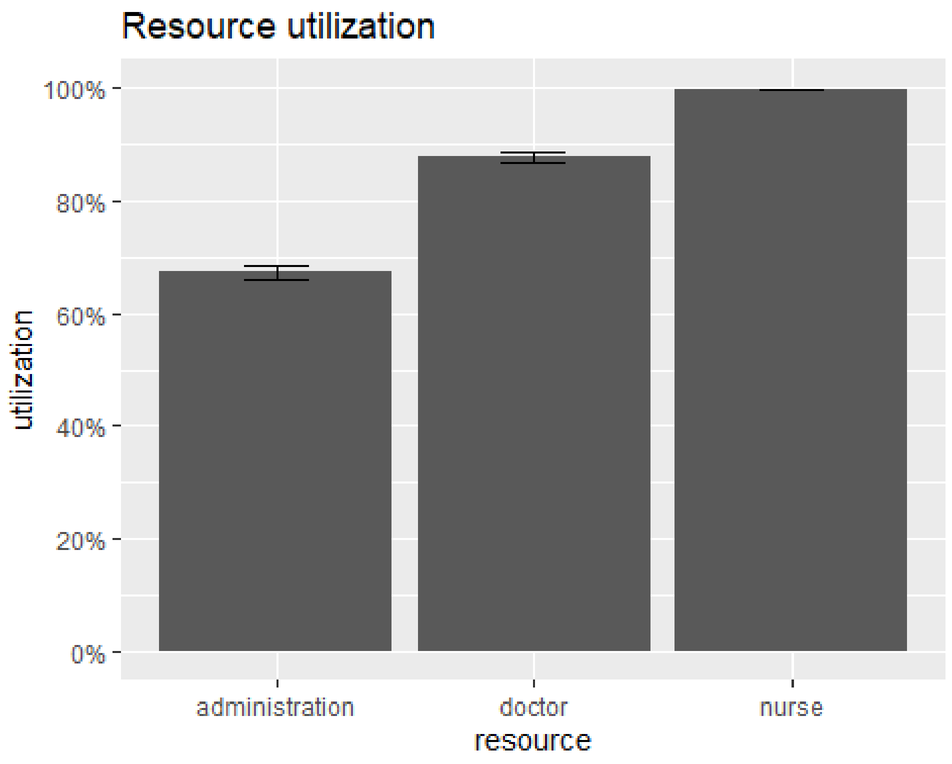
Plot of resources

Two metrics are available:

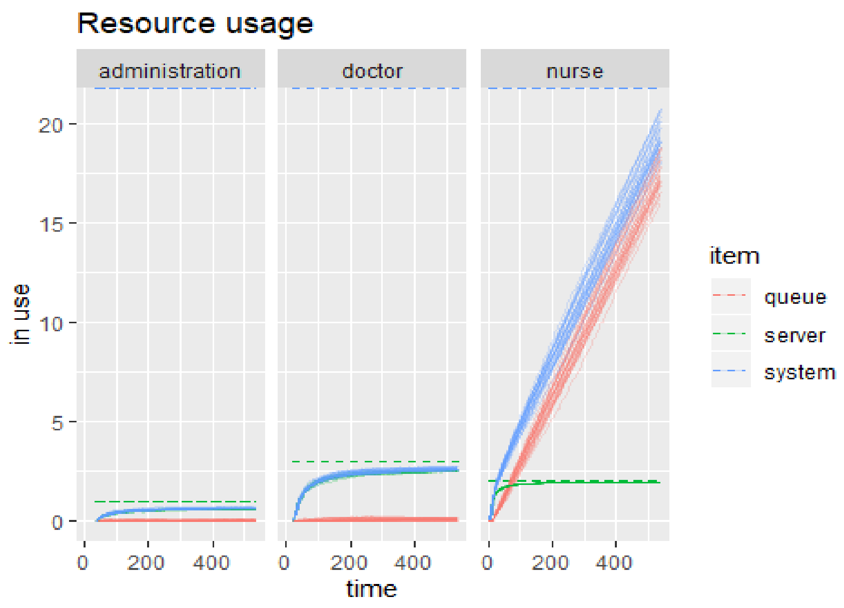
- The utilization of specified resources in the simulation.
- The usage of a resource over the simulation time frame.

```
library(simmer.plot)

resources <- get_mon_resources(envs)
head(resources)
```

```
##   resource      time server queue capacity queue_size system limit
## 1    nurse  4.719762      1     0        2        Inf      1   Inf
## 2    nurse 10.499116      2     0        2        Inf      2   Inf
## 3    nurse 15.563760      2     1        2        Inf      3   Inf
## 4    nurse 19.489585      2     0        2        Inf      2   Inf
## 5   doctor 19.489585      1     0        3        Inf      1   Inf
## 6    nurse 21.421293      2     1        2        Inf      3   Inf
##   replication
## 1           1
## 2           1
## 3           1
## 4           1
## 5             1
## 6             1

plot(resources, metric = "utilization")
```



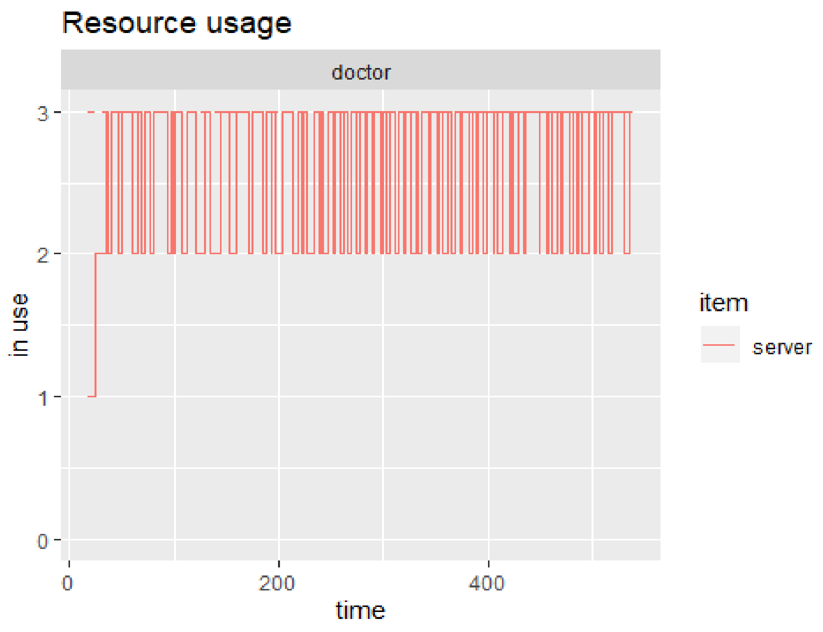
```
# It's the median occupancy ratio (no. of active servers divided by total cap
acity)

plot(resources, metric = "usage")
```



```
plot(resources, metric = "usage", c("nurse", "doctor"), items = "queue")$data

## # A tibble: 9,439 x 9
##    resource  time capacity queue_size limit replication item  value  mean
##    <fct>    <dbl>    <dbl>      <dbl> <dbl>       <int> <fct> <int> <dbl>
##  1 nurse     4.72        2        Inf   Inf           1 queue     0 0
##  2 nurse    10.5         2        Inf   Inf           1 queue     0 0
##  3 nurse    15.6         2        Inf   Inf           1 queue     1 0
##  4 nurse    19.5         2        Inf   Inf           1 queue     0 0.201
##  5 doctor   19.5         3        Inf   Inf           1 queue     0 0
##  6 nurse    21.4         2        Inf   Inf           1 queue     1 0.183
##  7 nurse    25.6         2        Inf   Inf           1 queue     0 0.316
##  8 doctor   25.6         3        Inf   Inf           1 queue     0 0
##  9 nurse    26.1         2        Inf   Inf           1 queue     1 0.310
## 10 nurse    31.3         2        Inf   Inf           1 queue     2 0.424
## # ... with 9,429 more rows

# item = queue: plot represent the cumulative average number of people in the
queue for each resource
# item = server: plot represent the cumulative average number of busy resourc
es
# item = system: plot represents the sum of the cumulative average number of
busy resources and the cumulative average number of people in the queue for e
ach resource
```

In the above graph, the individual lines are all independent replications. The smoothing performed is a cumulative average. Let's take a look now at the instantaneous behavior for a specific replica and resource. In the example below, the 6th replication is shown.

```
plot(get_mon_resources(envs[[6]]), metric = "usage", "doctor", items = "serve
r", steps = TRUE)
```

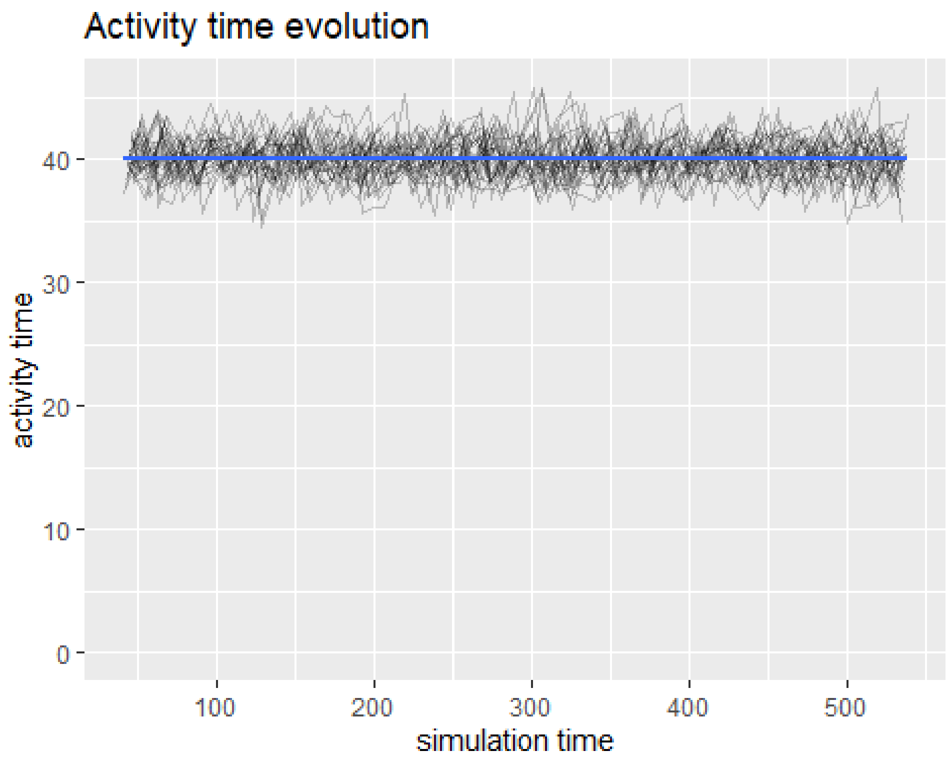


Plot of arrivals

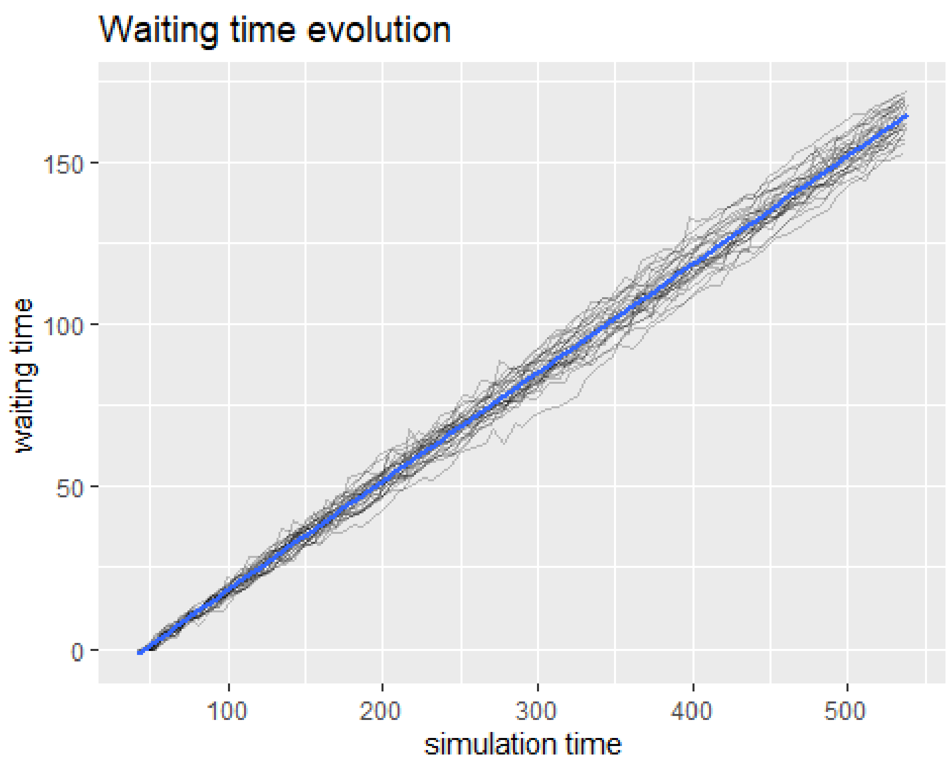
Three metrics available:

- Activity time.
- Waiting time.
- Flow time.

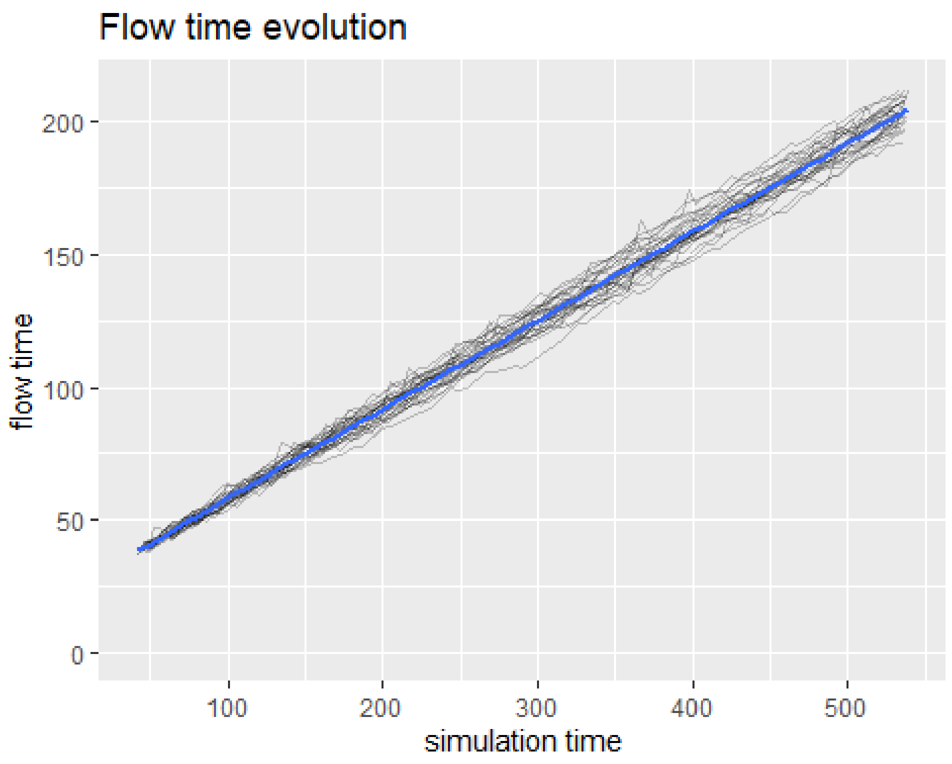
```
arrivals <- get_mon_arrivals(envs)
plot(arrivals, metric = "activity_time")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



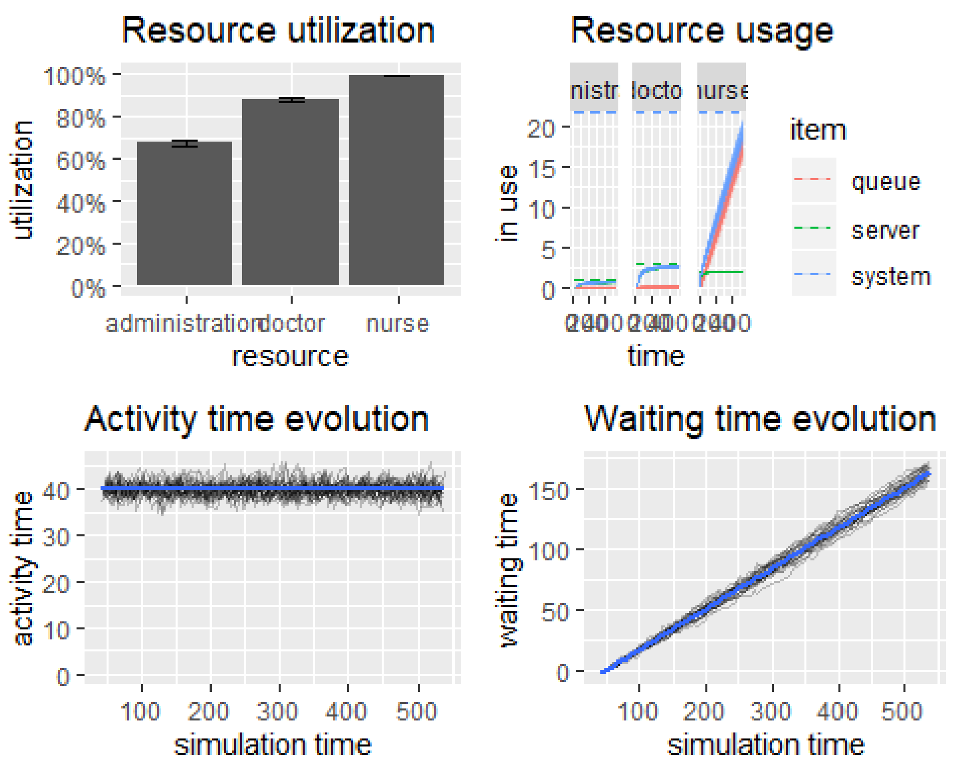
```
plot(arrivals, metric = "waiting_time")
```



```r
plot(arrivals, metric = "flow_time")
```



```r
# activity_time: is the time an entity spends with resources.
# waiting_time: is the time an entity spends in queues.
# flow_time: is the total time that the entity spends in the system (EndTime
- StartTime)

library(gridExtra)
p1 = plot(resources, metric = "utilization")
p2 = plot(resources, metric = "usage")
p3 = plot(arrivals, metric = "activity_time")
p4 = plot(arrivals, metric = "waiting_time")
grid.arrange(p1,p2,p3,p4)
```


Resource utilization
utilization
administration
doctor
nurse
resource
Resource usage
nistr
locto
iurse
in use
time
item
queue
server
system
Activity time evolution
activity time
simulation time
Waiting time evolution
waiting time
simulation time