

# CU25\_MODEL\_DEVELOPMENT\_03\_DES\_proceso

June 13, 2023

#

CU25\_Modelo de gestión de Lista de Espera Quirúrgica

## 1 IV. Model development

En este anexo se incluye el código utilizado durante el desarrollo de los modelos incluidos en el caso de uso.

### 1.1 Modelo DES - proceso masivo

```
[1]: Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

```
'es_ES.UTF-8/es_ES.UTF-8/es_ES.UTF-8/C/es_ES.UTF-8/C'
```

#### 1.1.1 Paquetes

```
[2]: ## PROCESO PREDICCIÓN Y SIMULACIÓN MASIVA
```

```
library(readr)
library(tidyverse)
library(tidymodels)
library(lubridate)
library(modeltime)
library(simmer)
library(simmer.plot)
library(tidyr)

## EN CASO DE QUE LA PREDICCIÓN AL FINAL DEL HORIZONTE RESULTE EN UN NÚMERO
↪ NEGATIVO, SE TOMA EL
## ÚLTIMO VALOR POSITIVO (O LO QUE ES LO MISMO, SE ELIMINAN LOS NEGATIVOS)
```

Attaching packages

tidyverse

1.3.2

ggplot2	3.4.2	dplyr	1.1.0
tibble	3.2.1	stringr	1.5.0
tidyr	1.3.0	forcats	0.5.2
purrr	1.0.1		

```

Conflicts
tidyverse_conflicts()
  dplyr::filter() masks stats::filter()
  dplyr::lag()    masks stats::lag()
Attaching packages                                tidymodels
1.0.0

```

```

  broom          1.0.3      rsample
1.1.1
  dials          1.1.0      tune
1.0.1
  infer          1.0.4      workflows
1.1.3
  modeldata      1.1.0      workflowsets
1.0.0
  parsnip        1.0.4      yardstick
1.1.0
  recipes        1.0.4

```

```

Conflicts
tidymodels_conflicts()
  scales::discard() masks
purrr::discard()
  dplyr::filter()   masks
stats::filter()
  recipes::fixed()  masks
stringr::fixed()
  dplyr::lag()      masks stats::lag()
  yardstick::spec() masks readr::spec()
  recipes::step()   masks stats::step()
• Search for functions across packages at
https://www.tidymodels.org/find/

```

Attaching package: ‘lubridate’

The following objects are masked from ‘package:base’:

```
date, intersect, setdiff, union
```

Attaching package: ‘simmer’

The following objects are masked from ‘package:lubridate’:

```
now, rollback
```

The following object is masked from 'package:dplyr':

```
select
```

The following object is masked from 'package:tidyr':

```
separate
```

Attaching package: 'simmer.plot'

The following objects are masked from 'package:simmer':

```
get_mon_arrivals, get_mon_attributes, get_mon_resources
```

## 1.2 Función

```
[3]: #s especialidad y zona
#periodo
#h horizonte para predicción serie
params_sim <- function(s, periodo, h){
  pred_h_personas <- modelo_pacientes |>
    pluck(s) |>
    modeltime_forecast(h = h) |>
    mutate(id = "personas")

  pred_h_tiempo <- modelo_tiempo |>
    pluck(s) |>
    modeltime_forecast(h = h) |>
    mutate(id = "tiempo")

  if(periodo == "ultimo"){

    pacientes_enCola <- modelo_pacientes |>
      pluck(s, ".calibration_data", 1) |>
      slice_max(fecha) |>
      pull(.actual)

    tiempo_medio_enCola <- modelo_tiempo |>
```

```

    pluck(s, ".calibration_data", 1) |>
    slice_max(fecha) |>
    pull(.actual)

  } else{

    pacientes_en_cola <- pred_h_personas |>
      filter(.value > 0) |>
      slice_tail(n = 1) |> pull(.value)

    tiempo_medio_en_cola <- pred_h_tiempo |>
      filter(.value > 0) |>
      slice_tail(n = 1) |> pull(.value)
  }

  if(pacientes_en_cola > 0 & tiempo_medio_en_cola > 0){
    lambda <- pacientes_en_cola / tiempo_medio_en_cola
    mu <- lambda/pacientes_en_cola
  } else{
    lambda <- mu <- NA
  }

  return(list(pacientes_en_cola = pacientes_en_cola,
              tiempo_medio_en_cola = tiempo_medio_en_cola,
              lambda = lambda,
              mu = mu,
              pred_h_tiempo = pred_h_tiempo,
              pred_h_personas = pred_h_personas))
}

```

### 1.2.1 Datos

```

[4]: ## Datos y modelos

modelo_pacientes <- read_rds("modelos_pacientes_xgboost.rds")
modelo_tiempo <- read_rds("modelos_tiempo_xgboost.rds")

capacidad <- read_csv("CU_25_05_07_01_capacidad.csv")

variables <- read_csv("VARIABLES.csv")

df <- data.frame(s = names(modelo_pacientes)) |>
  tidyr::separate(s, c("nombre_area", "Especialidad"), "\\.", remove = FALSE) |>
  left_join(capacidad)

```

Rows: 160 Columns: 4

### Column specification

Delimiter: ","

chr (3): id\_area, nombre\_area, Especialidad

dbl (1): capacidad

Use `spec()` to retrieve the full column specification for this data.

Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Rows: 2 Columns: 10

### Column specification

Delimiter: ","

chr (7): variable, tipo, control, nombre, etiqueta, descripcion, instrucciones

dbl (3): valor, min, max

Use `spec()` to retrieve the full column specification for this data.

Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Joining with `by = join\_by(nombre\_area, Especialidad)`

## 1.3 Simulación

```
[5]: ## COGER DE VARIABLES !!
H <- variables |>
  filter(variable == "HORIZONTE") |>
  pull(valor) |>
  as.numeric()

# H <- 2
NPER <- variables |>
  filter(variable == "NPER") |>
  pull(valor)

#simulación
message("Simulación según último valor")
l_sim_ultimo <- df |>
  # slice(1:2) |>
  pull(s) |>
  map(~{
    pars <- params_sim(s = .x, periodo = "ultimo", h = H)
    cap <- df |> filter(s == .x) |>
      pull(capacidad)
    env <- simmer("listasSim")
```

```

paciente <- trajectory("Trayectoria del paciente") %>%
  ## Operación
  seize("quirofano", 1) %>%
  timeout(function() rexp(1, pars$mu)) %>%
  release("quirofano", 1)

env %>%
  add_resource("quirofano", cap) %>%
  add_generator("inicial", paciente, at(rep(0, pars$pacientes_enCola))) |>
  add_generator("paciente", paciente, function() rpois(1, pars$lambda))

env %>%
  run(NPER)

recursos <- get_mon_resources(env)
llegadas <- get_mon_arrivals(env, ongoing = TRUE)
return(list(recursos = recursos |> mutate(id = .x),
            llegadas = llegadas |> mutate(id = .x),
            pred_h_personas = pars$pred_h_personas |> mutate(id = .x),
            pred_h_tiempo = pars$pred_h_tiempo |> mutate(id = .x)))

}, .progress = TRUE)
message("Simulación según horizonte")
l_sim_h <- df |>
  # slice(1:2) |>
  pull(s) |>
  map(~{
    pars <- params_sim(.x, periodo = "pred", h = H)
    cap <- df |> filter(s == .x) |>
      pull(capacidad)
    env <- simmer("listasSim")

    paciente <- trajectory("Trayectoria del paciente") %>%
      ## Operación
      seize("quirofano", 1) %>%
      timeout(function() rexp(1, pars$mu)) %>%
      release("quirofano", 1)

    env %>%
      add_resource("quirofano", cap) %>%
      add_generator("inicial", paciente, at(rep(0, pars$pacientes_enCola))) |>
      add_generator("paciente", paciente, function() rpois(1, pars$lambda))

    env %>%
      run(NPER)
  })

```

```

recursos <- get_mon_resources(env)
llegadas <- get_mon_arrivals(env, ongoing = TRUE)
return(list(recursos = recursos |> mutate(id = .x),
            llegadas = llegadas |> mutate(id = .x),
            pred_h_personas = pars$pred_h_personas |> mutate(id = .x),
            pred_h_tiempo = pars$pred_h_tiempo |> mutate(id = .x)))

}, .progress = TRUE)

sim_ultimo_recursos <- l_sim_ultimo |> map_dfr(~.x |> pluck("recursos"))
sim_ultimo_llegadas <- l_sim_ultimo |> map_dfr(~.x |> pluck("llegadas"))

sim_h_recursos <- l_sim_h |> map_dfr(~.x |> pluck("recursos"))
sim_h_llegadas <- l_sim_h |> map_dfr(~.x |> pluck("llegadas"))

pred_h_tiempo <- l_sim_h |> map_dfr(~.x |> pluck("pred_h_tiempo"))
pred_h_personas <- l_sim_h |> map_dfr(~.x |> pluck("pred_h_personas"))

write_rds(sim_ultimo_recursos, "sim_ultimo_recursos.rds")
write_rds(sim_ultimo_llegadas, "sim_ultimo_llegadas.rds")
write_rds(sim_h_recursos, "sim_h_recursos.rds")
write_rds(pred_h_tiempo, "pred_h_tiempo.rds")
write_rds(pred_h_personas, "pred_h_personas.rds")

```

Simulación según último valor

1% | ETA: 14m

1% | ETA: 12m

2% | ETA: 11m

2% | ETA: 11m

3% | ETA: 11m

4% | ETA: 10m

4% | ETA: 10m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

5% | ETA: 10m

6% | ETA: 10m

6% | ETA: 10m

7% | ETA: 10m

8% | ETA: 10m

8% | ETA: 10m

9% | ETA: 10m

9% | ETA: 10m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

10% | ETA: 10m

11% | ETA: 10m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

11% | ETA: 9m

12% | ETA: 9m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

12% | ETA: 9m

13% | ETA: 9m

14% | ETA: 9m

14% | ETA: 9m

15% | ETA: 9m

16% | ETA: 9m

16% | ETA: 9m

17% | ETA: 9m

18% | ETA: 9m

18% | ETA: 9m



19% | ETA: 8m

19% | ETA: 8m

20% | ETA: 8m

21% | ETA: 8m

21% | ETA: 8m

22% | ETA: 8m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

22% | ETA: 8m

23% | ETA: 8m

24% | ETA: 8m

24% | ETA: 8m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

25% | ETA: 8m

26% | ETA: 8m

26% | ETA: 7m

27% | ETA: 7m

28% | ETA: 7m

28% | ETA: 7m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

29% | ETA: 7m

29% | ETA: 7m

30% | ETA: 7m

31% | ETA: 7m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

31% | ETA: 7m

32% | ETA: 7m

32% | ETA: 7m

33% | ETA: 7m

34% | ETA: 7m

34% | ETA: 7m

35% | ETA: 7m

36% | ETA: 7m

36% | ETA: 6m

37% | ETA: 6m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

38% | ETA: 6m

38% | ETA: 6m

39% | ETA: 6m

39% | ETA: 6m

40% | ETA: 6m

41% | ETA: 6m

41% | ETA: 6m

42% | ETA: 6m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

42% | ETA: 6m

43% | ETA: 6m

Warning message in rpois(1, pars\$lambda):

"NAs produced"

44% | ETA: 6m

44% | ETA: 6m  
45% | ETA: 6m  
46% | ETA: 6m  
46% | ETA: 6m  
47% | ETA: 6m  
48% | ETA: 6m  
48% | ETA: 6m  
49% | ETA: 6m  
49% | ETA: 6m  
50% | ETA: 6m  
51% | ETA: 6m  
51% | ETA: 6m  
52% | ETA: 6m  
52% | ETA: 5m  
53% | ETA: 5m  
54% | ETA: 5m  
54% | ETA: 5m  
55% | ETA: 5m  
56% | ETA: 5m  
56% | ETA: 5m  
57% | ETA: 5m  
57% | ETA: 5m  
58% | ETA: 5m  
59% | ETA: 5m

```

59% | ETA: 5m

60% | ETA: 5m

61% | ETA: 5m

61% | ETA: 4m

62% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
62% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
63% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
64% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
64% | ETA: 4m

65% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
66% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
66% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
67% | ETA: 4m

68% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"
68% | ETA: 4m

Warning message in rpois(1, pars$lambda):
"NAs produced"

```

69% | ETA: 4m  
69% | ETA: 4m  
70% | ETA: 4m  
71% | ETA: 4m  
71% | ETA: 4m  
72% | ETA: 4m  
72% | ETA: 3m  
73% | ETA: 3m  
74% | ETA: 3m  
74% | ETA: 3m  
75% | ETA: 3m  
76% | ETA: 3m  
76% | ETA: 3m  
77% | ETA: 3m  
78% | ETA: 3m  
78% | ETA: 3m  
79% | ETA: 3m  
79% | ETA: 3m  
80% | ETA: 2m  
81% | ETA: 2m  
81% | ETA: 2m  
82% | ETA: 2m  
82% | ETA: 2m  
83% | ETA: 2m

84% | ETA: 2m  
84% | ETA: 2m  
85% | ETA: 2m  
86% | ETA: 2m  
86% | ETA: 2m  
87% | ETA: 2m  
88% | ETA: 2m  
88% | ETA: 1m  
89% | ETA: 1m  
89% | ETA: 1m  
90% | ETA: 1m  
91% | ETA: 1m  
91% | ETA: 1m  
92% | ETA: 1m  
92% | ETA: 1m  
93% | ETA: 1m  
94% | ETA: 49s  
94% | ETA: 45s  
95% | ETA: 40s  
96% | ETA: 35s  
96% | ETA: 30s  
97% | ETA: 25s  
98% | ETA: 20s  
98% | ETA: 15s

99% | ETA: 10s

99% | ETA: 5s

Simulación según horizonte

1% | ETA: 20m

1% | ETA: 20m

2% | ETA: 23m

2% | ETA: 22m

3% | ETA: 22m

4% | ETA: 21m

4% | ETA: 21m

5% | ETA: 20m

6% | ETA: 20m

6% | ETA: 20m

7% | ETA: 20m

8% | ETA: 19m

8% | ETA: 19m

9% | ETA: 18m

9% | ETA: 18m

10% | ETA: 18m

11% | ETA: 17m

11% | ETA: 17m

12% | ETA: 17m

12% | ETA: 17m

13% | ETA: 17m

14% | ETA: 17m  
14% | ETA: 17m  
15% | ETA: 17m  
16% | ETA: 16m  
16% | ETA: 16m  
17% | ETA: 16m  
18% | ETA: 16m  
18% | ETA: 15m  
19% | ETA: 15m  
19% | ETA: 15m  
20% | ETA: 15m  
21% | ETA: 14m  
21% | ETA: 14m  
22% | ETA: 14m  
22% | ETA: 14m  
23% | ETA: 13m  
24% | ETA: 13m  
24% | ETA: 13m  
25% | ETA: 13m  
26% | ETA: 13m  
26% | ETA: 12m  
27% | ETA: 12m  
28% | ETA: 12m  
28% | ETA: 12m



29% | ETA: 12m  
29% | ETA: 11m  
30% | ETA: 11m  
31% | ETA: 11m  
31% | ETA: 11m  
32% | ETA: 11m  
32% | ETA: 11m  
33% | ETA: 11m  
34% | ETA: 10m  
34% | ETA: 10m  
35% | ETA: 10m  
36% | ETA: 10m  
36% | ETA: 10m  
37% | ETA: 10m  
38% | ETA: 10m  
38% | ETA: 10m  
39% | ETA: 9m  
39% | ETA: 9m  
40% | ETA: 9m  
41% | ETA: 9m  
41% | ETA: 9m  
42% | ETA: 9m  
42% | ETA: 9m  
43% | ETA: 9m

44% | ETA: 9m  
44% | ETA: 8m  
45% | ETA: 8m  
46% | ETA: 8m  
46% | ETA: 8m  
47% | ETA: 8m  
48% | ETA: 8m  
48% | ETA: 8m  
49% | ETA: 8m  
49% | ETA: 7m  
50% | ETA: 7m  
51% | ETA: 7m  
51% | ETA: 7m  
52% | ETA: 7m  
52% | ETA: 7m  
53% | ETA: 7m  
54% | ETA: 7m  
54% | ETA: 7m  
55% | ETA: 6m  
56% | ETA: 6m  
56% | ETA: 6m  
57% | ETA: 6m  
57% | ETA: 6m  
58% | ETA: 6m

59% | ETA: 6m  
59% | ETA: 6m  
60% | ETA: 6m  
61% | ETA: 5m  
61% | ETA: 5m  
62% | ETA: 5m  
62% | ETA: 5m  
63% | ETA: 5m  
64% | ETA: 5m  
64% | ETA: 5m  
65% | ETA: 5m  
66% | ETA: 5m  
66% | ETA: 5m  
67% | ETA: 5m  
68% | ETA: 4m  
68% | ETA: 4m  
69% | ETA: 4m  
69% | ETA: 4m  
70% | ETA: 4m  
71% | ETA: 4m  
71% | ETA: 4m  
72% | ETA: 4m  
72% | ETA: 4m  
73% | ETA: 4m

74% | ETA: 4m  
74% | ETA: 3m  
75% | ETA: 3m  
76% | ETA: 3m  
76% | ETA: 3m  
77% | ETA: 3m  
78% | ETA: 3m  
78% | ETA: 3m  
79% | ETA: 3m  
79% | ETA: 3m  
80% | ETA: 3m  
81% | ETA: 3m  
81% | ETA: 3m  
82% | ETA: 2m  
82% | ETA: 2m  
83% | ETA: 2m  
84% | ETA: 2m  
84% | ETA: 2m  
85% | ETA: 2m  
86% | ETA: 2m  
86% | ETA: 2m  
87% | ETA: 2m  
88% | ETA: 2m  
88% | ETA: 2m

89% | ETA: 2m  
89% | ETA: 1m  
90% | ETA: 1m  
91% | ETA: 1m  
91% | ETA: 1m  
92% | ETA: 1m  
92% | ETA: 1m  
93% | ETA: 1m  
94% | ETA: 50s  
94% | ETA: 45s  
95% | ETA: 40s  
96% | ETA: 35s  
96% | ETA: 30s  
97% | ETA: 25s  
98% | ETA: 20s  
98% | ETA: 15s  
99% | ETA: 10s  
99% | ETA: 5s