Notes on simulation

Jono Tuke

Wed 09 Dec 2020

Table of Contents

# Basic form for simulated data

The basic form, referred to as a simDB, is a data frame such that that each row is a second, it has the following columns

* **time:** the second of interest.
* **type:** the type of behaviour

The types of behaviour will of two forms:

* background
* event / state

So we are interested in the event / state behaviour.

## Notation

We will represent the data by

where in our case and

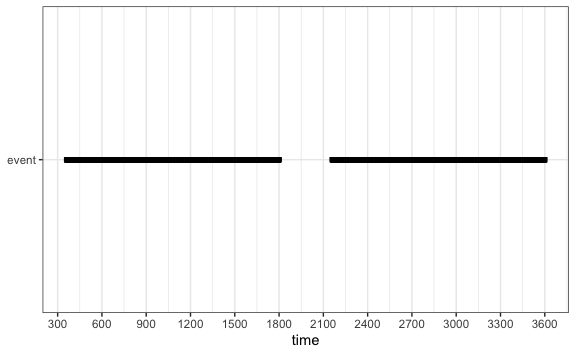
## Example

Consider the following data frame

tar\_load(example\_simDB)  
example\_simDB

## # A tibble: 3,600 x 2  
## time type   
## <int> <chr>   
## 1 1 background  
## 2 2 background  
## 3 3 background  
## 4 4 background  
## 5 5 background  
## 6 6 background  
## 7 7 background  
## 8 8 background  
## 9 9 background  
## 10 10 background  
## # … with 3,590 more rows

plot\_simDB(example\_simDB)



# Percent of time event occurs

The key measure we are interested in is the percent of the time that the event occurs. We stick to proportion as this is nicer mathematically, and then convert to percent for presentation.

It is defined as

get\_prop(example\_simDB) %>% scales::percent()

## [1] "80%"

# Simulation method

## Response frequency

In these simulations, we have a behaviour that is short (1 second). The behaviour occurs

* frequent (every 3 seconds),
* moderate (every 30 seconds), or
* infrequent (every 300 seconds).

The function sim\_response\_freq() produces a simDB with these properties. It works by splitting the time into blocks of length equal to the frequency, so for moderate frequency, we have blocks of 30 seconds, then one of the seconds in this time are randomly selected to be the event.

So we split the data

where

into blocks

Block 1

Block 2

So we have

Block

The final block is

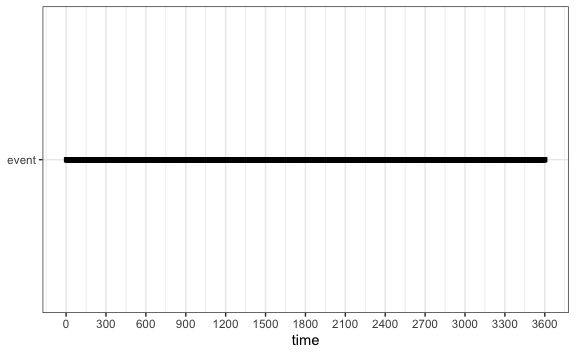
where

We then randomly select one of the time points in the block with equal probability,

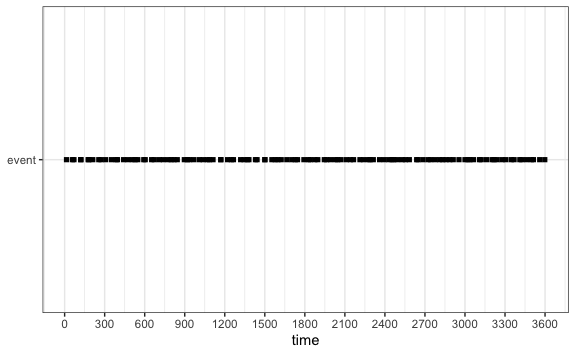
then we set

and

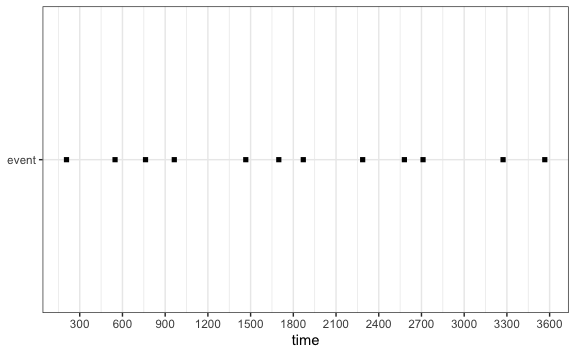
sim\_response\_freq(3) %>% plot\_simDB()



sim\_response\_freq(30) %>% plot\_simDB()



sim\_response\_freq(300) %>% plot\_simDB()



sim\_response\_freq(3) %>% get\_prop()

## [1] 0.3333333

sim\_response\_freq(30) %>% get\_prop()

## [1] 0.03333333

sim\_response\_freq(300) %>% get\_prop()

## [1] 0.003333333

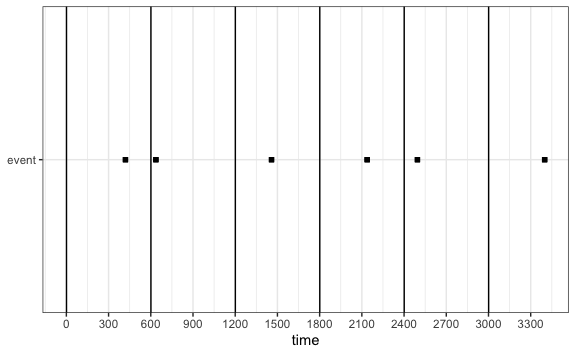
## Response duration

In this simulation, the duration is

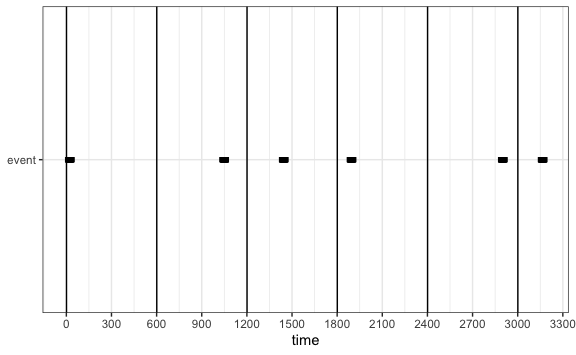
* short (3 second),
* medium (30 seconds) and
* long (300 seconds).

A single event occurs in each 10 minute interval, and is randomly placed in this interval.

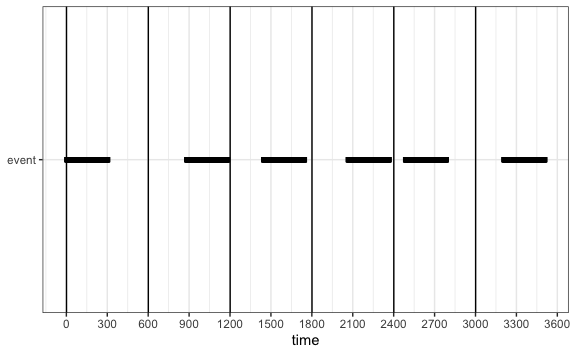
sim\_response\_duration(3) %>% plot\_simDB() + geom\_vline(xintercept = seq(1, 3600, 600))



sim\_response\_duration(30) %>% plot\_simDB() + geom\_vline(xintercept = seq(1, 3600, 600))



sim\_response\_duration(300) %>% plot\_simDB() + geom\_vline(xintercept = seq(1, 3600, 600))



sim\_response\_duration(3) %>% get\_prop()

## [1] 0.005

sim\_response\_duration(30) %>% get\_prop()

## [1] 0.05

sim\_response\_duration(300) %>% get\_prop()

## [1] 0.5

# Sampling methods

So we have two sampling methods:

## Pinpoint sampling

In pinpoint sampling, we sample at regular intervals, so for example, if the continuous data is represented by

where in our case and

Then if we have an interval of , then the pinpoint sample is

where

The function pin\_point\_sampling() takes an interval and a simDB and returns the proportion of the time-points that are an event, i.e.

pin\_point\_sampling(example\_simDB, 3)

## # A tibble: 1 x 3  
## method delta p  
## <chr> <dbl> <dbl>  
## 1 PP 3 0.8

pin\_point\_sampling(example\_simDB, 30)

## # A tibble: 1 x 3  
## method delta p  
## <chr> <dbl> <dbl>  
## 1 PP 30 0.8

pin\_point\_sampling(example\_simDB, 300)

## # A tibble: 1 x 3  
## method delta p  
## <chr> <dbl> <dbl>  
## 1 PP 300 0.667

## One-zero sampling

In one-zero sampling, we split the continuous data into contiguous blocks of length , i.e.,

Block 1

Block 2

So we have

Block

The final block is

where

Note that this final block may not be the same length as the previous blocks.

If we let the result for Block be denoted by , then we have

Then

The function one\_zero\_sampling() takes a simDB and a , and returns the proportion.

one\_zero\_sampling(example\_simDB, 3)

## # A tibble: 1 x 3  
## method delta p  
## <chr> <dbl> <dbl>  
## 1 01 3 0.802

one\_zero\_sampling(example\_simDB, 30)

## # A tibble: 1 x 3  
## method delta p  
## <chr> <dbl> <dbl>  
## 1 01 30 0.817

one\_zero\_sampling(example\_simDB, 300)

## # A tibble: 1 x 3  
## method delta p  
## <chr> <dbl> <dbl>  
## 1 01 300 0.833

# Simulation protocol

So for each simulation, we have

* method used - response frequency (RF), or response duration (RD),
* parameter - 3, 30, 300 interval (RF), or duration (RD).

This gives 6 different types of simulation, which we will repeat 100 times each giving 600 simulations.

Discussion with Eduardo

Frequency

Classification of frequency

|  |  |
| --- | --- |
| Percent | Classification |
| 25-50% | Frequent |
| 10-15% | Moderate |
| 1-2% | Infrequent |

# Sampling protocol

For each simulation, we have

* sampling method - pinpoint - intervals 3, 30, and 300, while for one-zero, we sample every 10 minutes for duration of 3, 30, 300 seconds.

So again we have 6 sampling procedures, that will give use altogether 3600 measurements.

# Results

For each combination of simulation method, simulation parameters, sample method, and sample parameters, we have 100 simulations. We have calculated the error rate (true proportion of time - estimated proportion of time events occur). These are given in the table below. The 95% intervals are based on percentile intervals.

## Summary statistics

Summary statistics of error rate for each simulation type

Grouped by simulation method and sampling method

Simulation parameters

Sampling parameters

Proportion of time event occurs

Mean error

Lower 95% percentile of error

Upper 95% percentile of error

RD - 01

3

5

0.005000000

6.666667e-03

0.003333333

0.010277778

3

50

0.005000000

8.236111e-02

0.078333333

0.092222222

3

500

0.005000000

6.837500e-01

0.495000000

0.745000000

30

5

0.050000000

6.805556e-03

0.004166667

0.008333333

30

50

0.050000000

7.736111e-02

0.047222222

0.102777778

30

500

0.050000000

6.587500e-01

0.450000000

0.825000000

300

5

0.500000000

6.652778e-03

0.003437500

0.008333333

300

50

0.500000000

8.208333e-02

0.069444444

0.083333333

300

500

0.500000000

4.262500e-01

0.375000000

0.500000000

RD - PP

3

5

0.005000000

-9.108145e-20

-0.002222222

0.003333333

3

50

0.005000000

1.111111e-03

-0.005000000

0.022777778

3

500

0.005000000

2.500000e-03

-0.005000000

0.120000000

30

5

0.050000000

0.000000e+00

0.000000000

0.000000000

30

50

0.050000000

-4.722222e-03

-0.036111111

0.019444444

30

500

0.050000000

-1.500000e-02

-0.050000000

0.200000000

300

5

0.500000000

0.000000e+00

0.000000000

0.000000000

300

50

0.500000000

0.000000e+00

0.000000000

0.000000000

300

500

0.500000000

-7.500000e-02

-0.250000000

0.125000000

RF - 01

3

5

0.333333333

6.666667e-01

0.666666667

0.666666667

3

50

0.333333333

6.666667e-01

0.666666667

0.666666667

3

500

0.333333333

6.666667e-01

0.666666667

0.666666667

30

5

0.033333333

1.333333e-01

0.133333333

0.133333333

30

50

0.033333333

9.666667e-01

0.966666667

0.966666667

30

500

0.033333333

9.666667e-01

0.966666667

0.966666667

300

5

0.003333333

1.333333e-02

0.013333333

0.013333333

300

50

0.003333333

1.633333e-01

0.163333333

0.163333333

300

500

0.003333333

9.016667e-01

0.871666667

0.996666667

RF - PP

3

5

0.333333333

1.916667e-03

-0.038888889

0.040277778

3

50

0.333333333

6.944444e-03

-0.118402778

0.118402778

3

500

0.333333333

-1.583333e-02

-0.333333333

0.291666667

30

5

0.033333333

-4.166667e-04

-0.009722222

0.009722222

30

50

0.033333333

9.722222e-04

-0.033333333

0.036111111

30

500

0.033333333

5.416667e-03

-0.033333333

0.157291667

300

5

0.003333333

2.916667e-04

-0.003333333

0.005000000

300

50

0.003333333

-2.777778e-04

-0.003333333

0.024444444

300

500

0.003333333

1.666667e-03

-0.003333333

0.121666667

RD: Response duration

RF: Response frequency

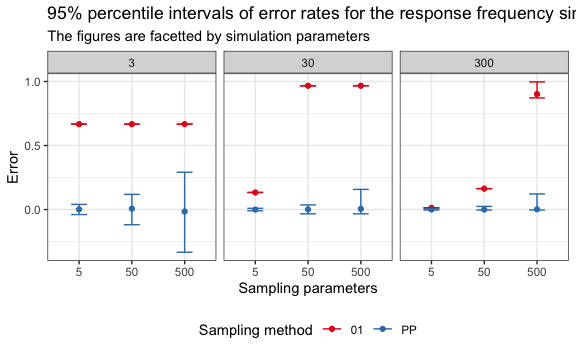
PP: Pinpoint sampling

01: One-zero sampling

Notes, easiest way to get table out, is to save as RTF, and then paste into word.

results\_tab %>%   
 gtsave("tabs/results.rtf")

## Response frequency simulation

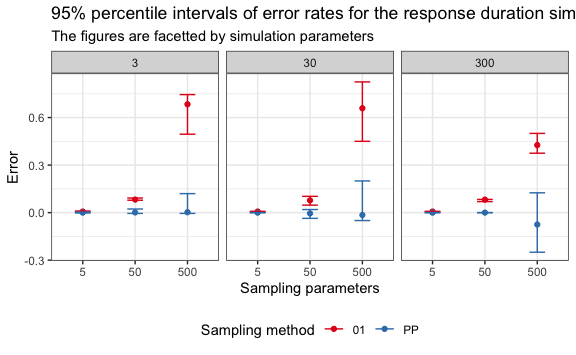


ggsave("figs/RF\_plot.jpg", width = 10, height = 6)

Notice that the zero-one sampling is biased in nearly all cases, over-estimating the true percent of the time that the event occurs. This overestimation gets worse as the sampling parameter increases.

With pinpoint sampling, we find that the method is unbiased, but the variability of the estimate increases as the interval between samples increases.

## Response duration simulation



ggsave("figs/RD\_plot.jpg", width = 10, height = 6)

Again we see that one-zero sampling is a poor estimator, not quite as consistently bad as with response frequency, but still pretty bad. Again, we see the over-estimation increases as the duration of the one-zero sampling increases.