Analysing data

Alexis Derumigny

03/03/2022

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

We first load the data of the simulations that we did.

```
source("loadingData.R")
```

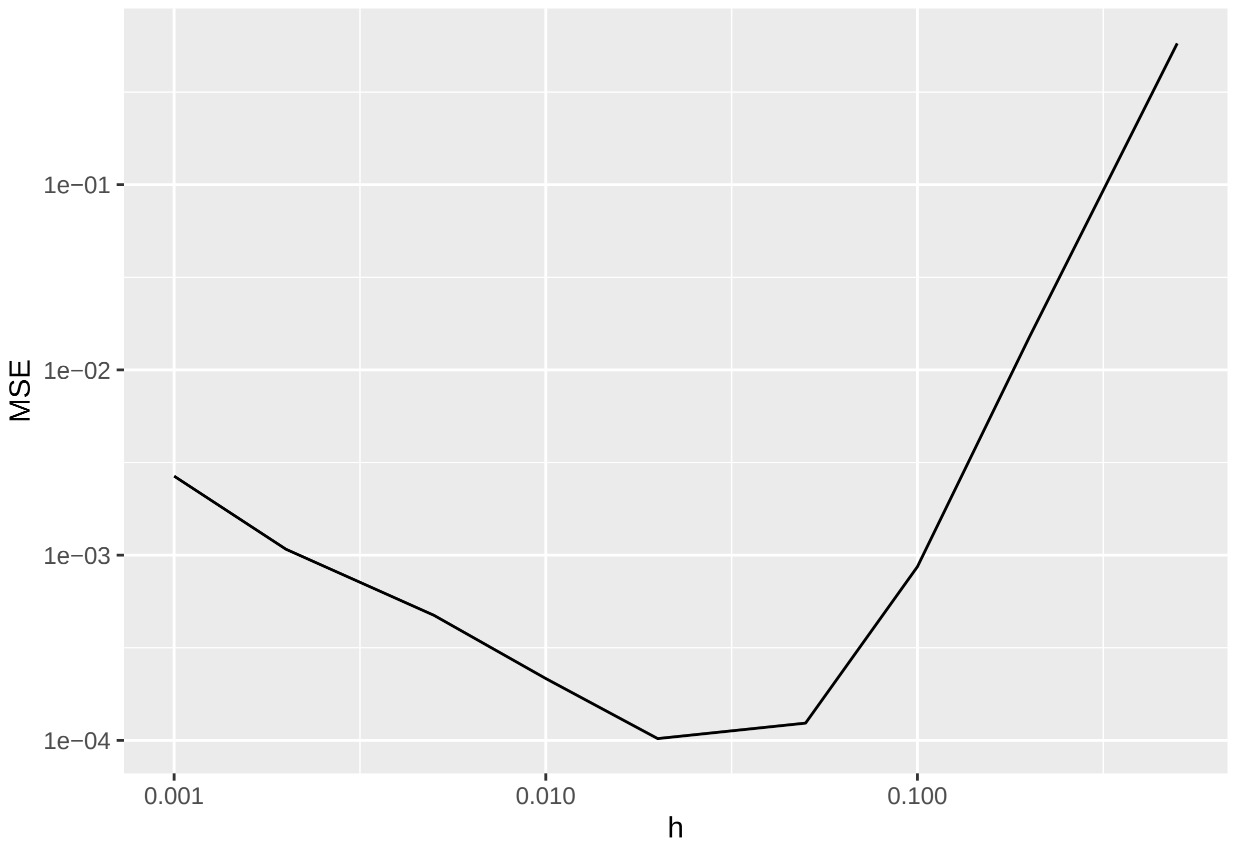
We can now print the summary statistics.

```
summarisedData %>%
  select(all_of(c("n", "h", "MSE", "meanComputationTime")))
```

```
## # A tibble: 9 x 4
##       n     h      MSE meanComputationTime
##   <int> <dbl>    <dbl>               <dbl>
## 1 50000 0.001 0.00267              0.00180
## 2 50000 0.002 0.00108              0.00130
## 3 50000 0.005 0.000473             0.00150
## 4 50000 0.01  0.000216             0.00100
## 5 50000 0.02  0.000102             0.00190
## 6 50000 0.05  0.000124             0.00180
## 7 50000 0.1   0.000866             0.00260
## 8 50000 0.2   0.0150               0.00380
## 9 50000 0.5   0.579                0.00480
```

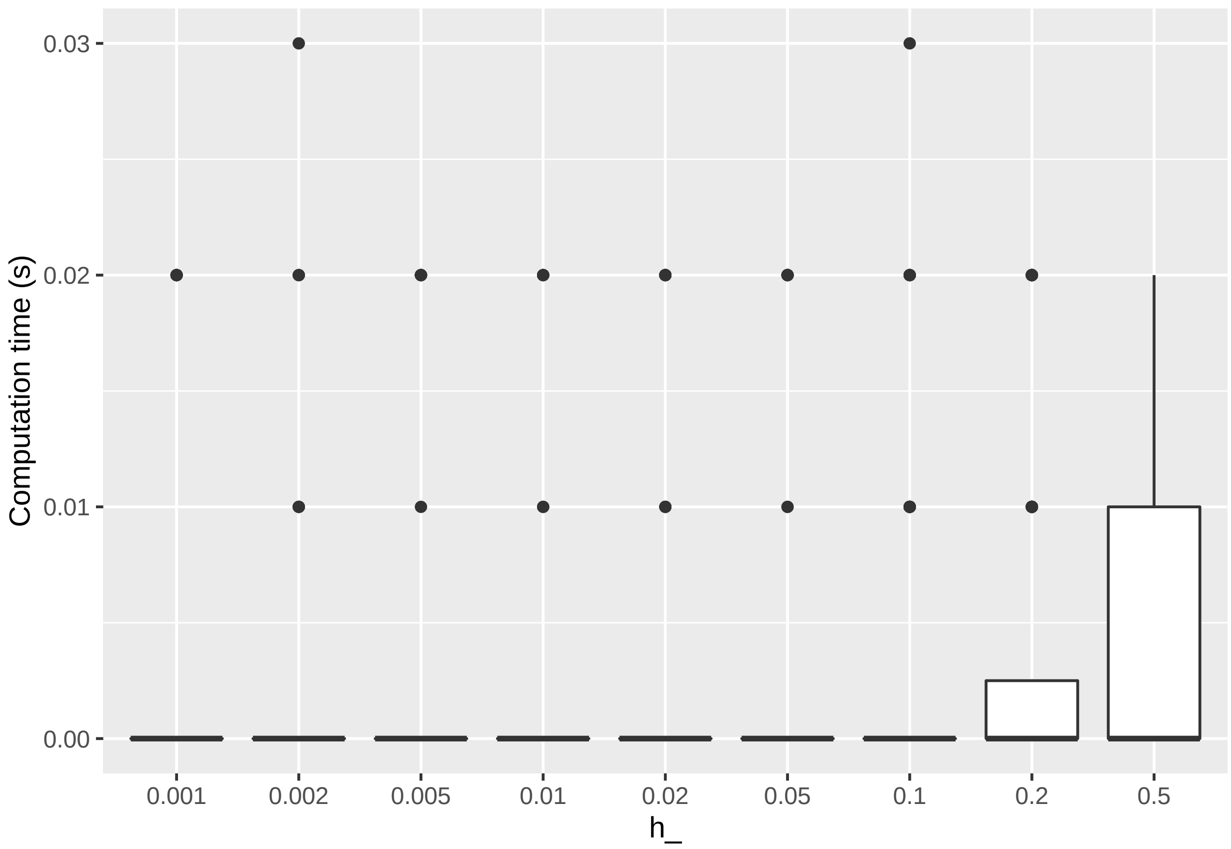
We plot now the mean-squared error as a function of the bandwidth h.

```
summarisedData %>%
  ggplot(aes(x = h, y = MSE)) +
  geom_line() +
  scale_x_log10() +
  scale_y_log10()
```



We can also plot the distribution of the computation time as a function of h.

```
totalData %>%
  mutate( h_ = factor(h, levels = sort(unique(h))) ) %>%
  ggplot(aes(x = h_, y = computationTime)) +
  geom_boxplot() +
  ylab("Computation time (s)")
```


Computation time (s)
0.03
0.02
0.01
0.00
0.001
0.002
0.005
0.01
0.02
0.05
0.1
0.2
0.5
h_