09 Estimating Cost Functions

Fabian Blasch

06/09/2022

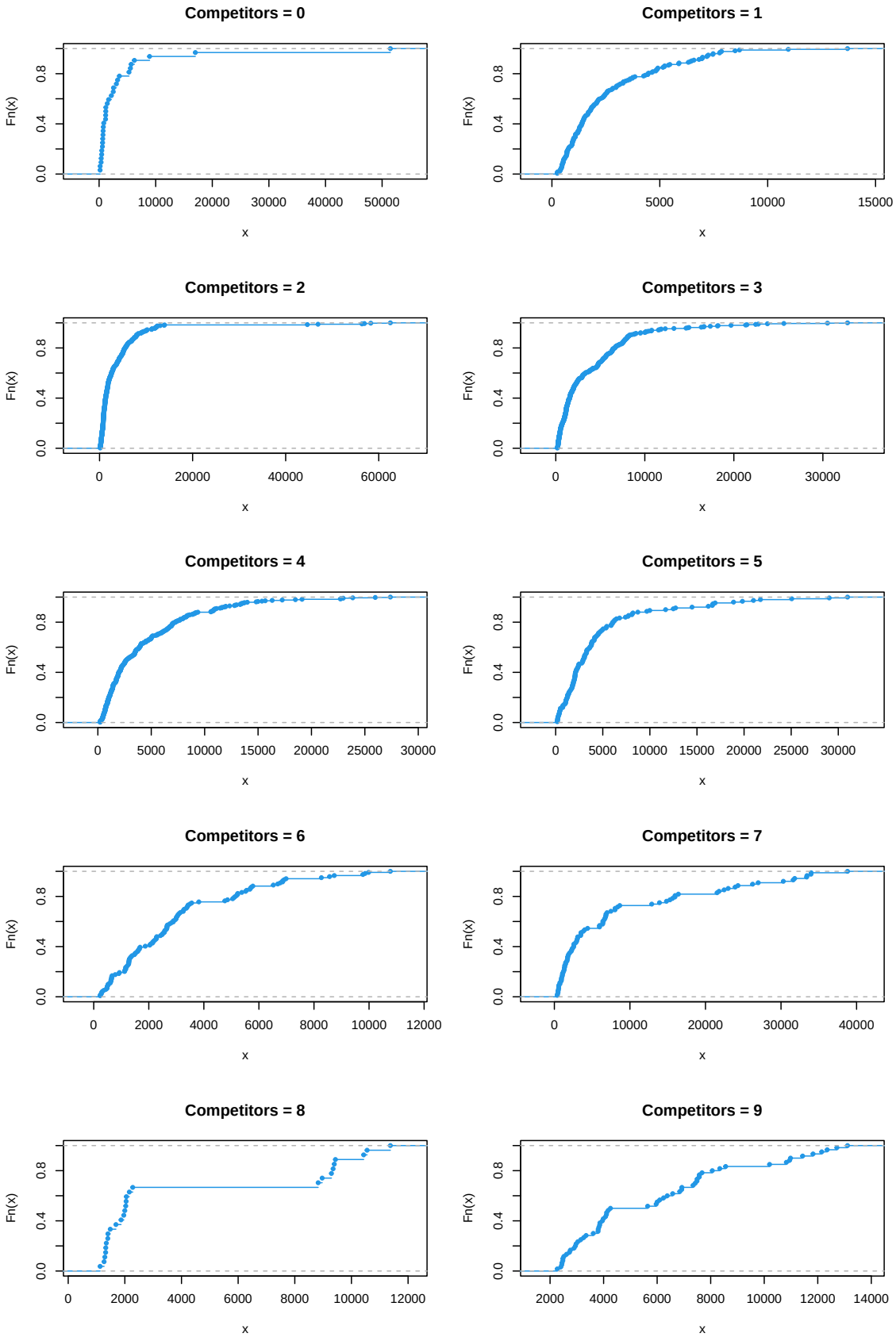
1 Load Data

```
# knitr options
knitr::opts_chunk$set(echo = TRUE,
                      fig.pos = "center",
                      fig.width = 8,
                      fig.height = 4,
                      fig.pos = "H")

# source AUX
source("./../Misc/Auxilliary.R")
source("./../Misc/model_eval.R")

# packages
get.package(c("lubridate", "glmnet", "glmnetUtils", "tidyverse", "EnvStats",
              "kdensity"))

# load data
dat_bids <- readRDS("./../../Data/Bid Tab RDS/Bids_df_split.RDS")
dat_aucs <- readRDS("./../../Data/Bid Tab RDS/Aucs_df_split.RDS")
dat_bids_ind <- readRDS("./../../Data/Bid Tab RDS/Bids_id_df.RDS")

# bid amount counted in thousands
dat_bids_ind[, "Total_Bid"] <- dat_bids_ind[, "Total_Bid"] / 1000
```

2 ECDFS

```
# split bids by competing firms and then again by auction
split(dat_bids_ind, dat_bids_ind[, "N_Comp"]) -> split_ncomp

# align
par(mfrow = c(5, 2))

# boxplots
invis.Map(\(x, n){
```

```
  # ecdf
  df <- ecdf(x[, "Total_Bid"])

  # plot
  plot(df, main = paste0("Competitors = ", n), col = 4)

  # ret
  return(df)

}, split_ncomp, 0:(length(split_ncomp) - 1)) -> bid_n_ecdfs
```


Competitors = 0
Competitors = 1
Competitors = 2
Competitors = 3
Competitors = 4
Competitors = 5
Competitors = 6
Competitors = 7
Competitors = 8
Competitors = 9
Fn(x)
x

3 Estimation

First we define the expected profit from procurement as

$$\pi(b, x) = (b - x)H(b)$$

where b is the bid and x is the signal associated with the contract at hand.

Let $H(b)$ be the cumulative distribution function of the lowest bid, the derivation from the cdf of the bids is trivial.

$$H(b) = 1 - (1 - F_b(b))^n$$

and thus

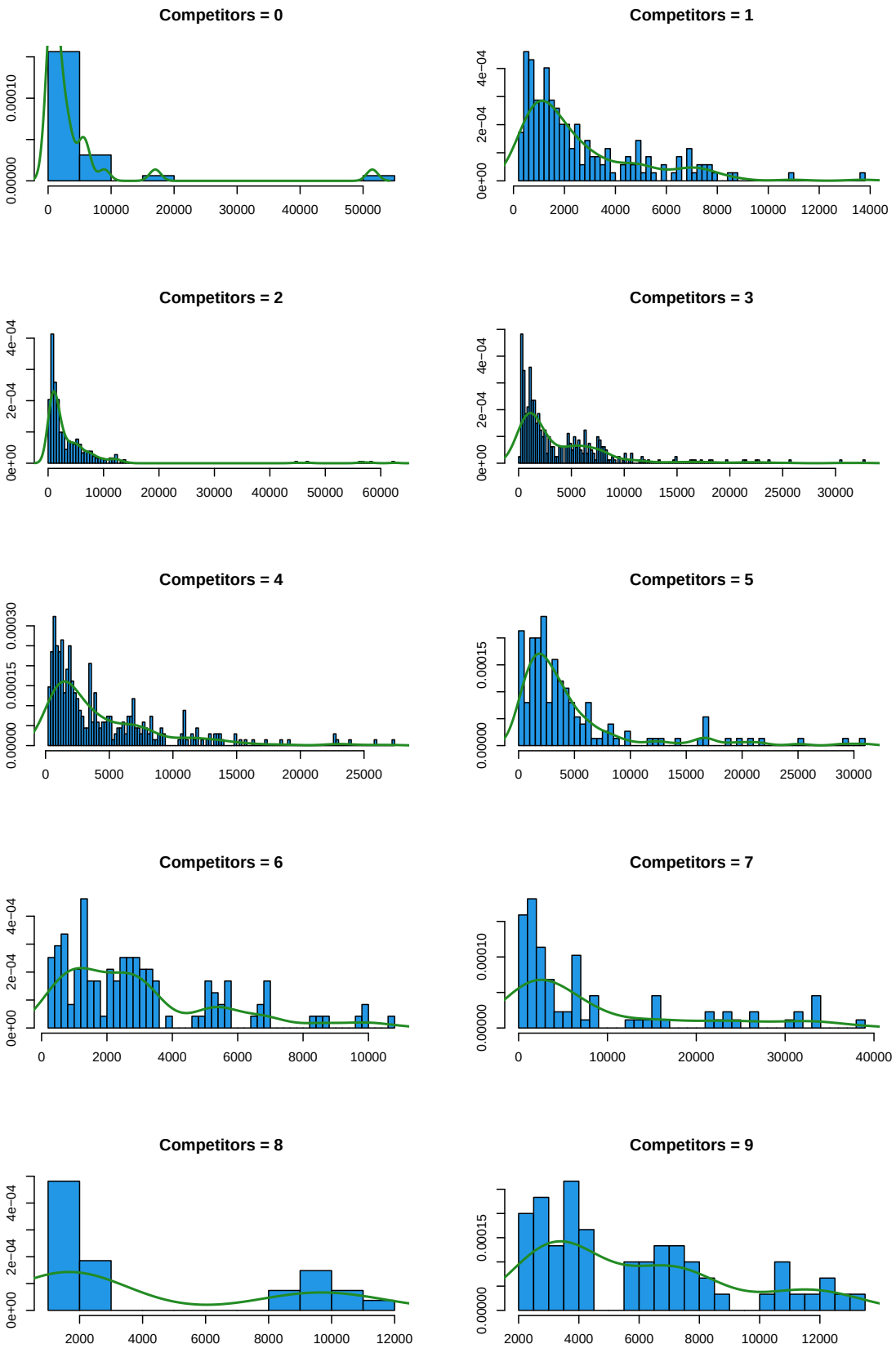
$$h(b) = \frac{\partial H(b)}{\partial b} = n(1 - F(b))^{n-1} f(b)$$

where n represents the number of competitors within an auction. Estimates are obtained for $n \in [0, 9] \cap \mathbb{N}$. Hence, $H(b)$ and $h(b)$ are directly identified from the data.

Accordingly we need to estimate the density as well as the cdf of bids from the data, conditional on auction characteristics and the nuber of competitors n. As we observe the bids, we may use a kernel regression to first estimate the density of the bids to then derive the cdf from the density. Subsequently, we may then obtain $H(b)$ and $h(b)$.

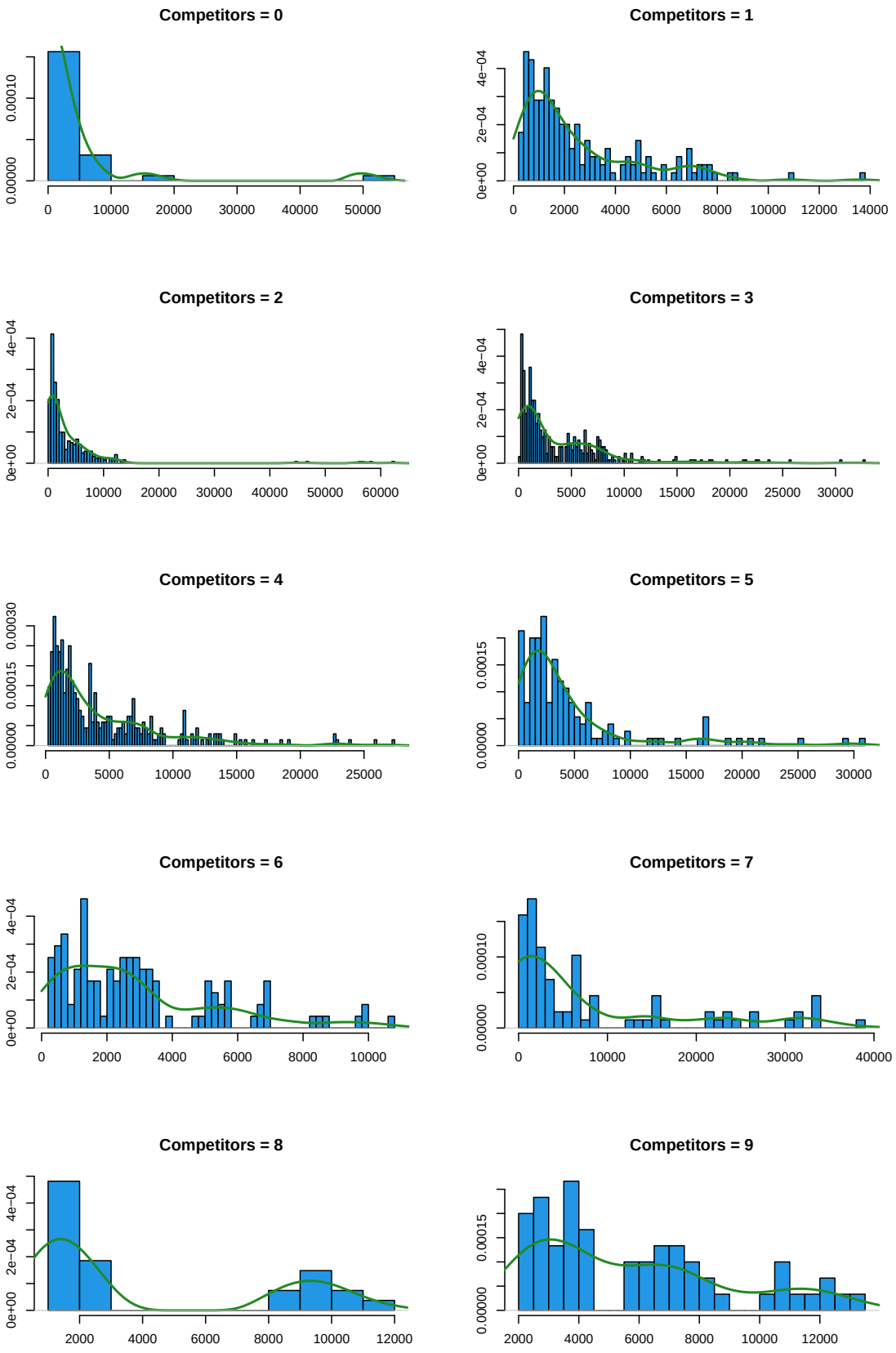
3.1 Densities via Base R and EnvStats

```
## Estimation using base R density (implemented via EnvStats)

# align
par(mfrow = c(5, 2), mar = c(5, 4, 4, 2) + 0.1)

# Bid density functions
invis.Map(\(inp, n){

  # ecdf
  dens <- \(x) demp(x, inp[, "Total_Bid"])

  # plot
  hist(inp[, "Total_Bid"], main = paste0("Competitors = ", n), col = 4, ylab = "",
       prob = TRUE, breaks = length(inp[, "Total_Bid"]) / 2,
       xlab = "")

  # lines
  # lines(dens(seq(1e5, 20e6, 1000)), col = "cornflowerblue")
  lines(density(inp[, "Total_Bid"]), col = "forestgreen", lwd = 2)

  # ret
  return(dens)

}, split_ncomp, 0:(length(split_ncomp) - 1)) -> bid_n_dens_1
```


Competitors = 0
0.00000
0.00010
0
10000
20000
30000
40000
50000
Competitors = 1
0e+00
2e−04
4e−04
0
2000
4000
6000
8000
10000
12000
14000
Competitors = 2
0e+00
2e−04
4e−04
0
10000
20000
30000
40000
50000
60000
Competitors = 3
0e+00
2e−04
4e−04
0
5000
10000
15000
20000
25000
30000
Competitors = 4
0.00000
0.00015
0.00030
0
5000
10000
15000
20000
25000
Competitors = 5
0.00000
0.00015
0
5000
10000
15000
20000
25000
30000
Competitors = 6
0e+00
2e−04
4e−04
0
2000
4000
6000
8000
10000
Competitors = 7
0.00000
0.00010
0
10000
20000
30000
40000
Competitors = 8
0e+00
2e−04
4e−04
2000
4000
6000
8000
10000
12000
Competitors = 9
0.00000
0.00015
2000
4000
6000
8000
10000
12000

3.2 Densities via kdensity

```
# align
par(mfrow = c(5, 2), mar = c(5, 4, 4, 2) + 0.1)

# Bid density functions
invis.Map(\(inp, n){

  # ecdf
  dens <- kdensity(inp[, "Total_Bid"], start = "exponential", kernel = "triweight",
                   support = c(0, Inf))

  # plot
  hist(inp[, "Total_Bid"], main = paste0("Competitors = ", n), col = 4, ylab = "",
       prob = TRUE, breaks = length(inp[, "Total_Bid"]) / 2,
       xlab = "")

  # lines
  lines(dens, col = "forestgreen", lwd = 2)

  # ret
  return(dens)

}, split_ncomp, 0:(length(split_ncomp) - 1)) -> bid_n_dens_2
```


Competitors = 0
0.00000
0.00010
0 10000 20000 30000 40000 50000
Competitors = 1
0e+00
2e−04
4e−04
0 2000 4000 6000 8000 10000 12000 14000
Competitors = 2
0e+00
2e−04
4e−04
0 10000 20000 30000 40000 50000 60000
Competitors = 3
0e+00
2e−04
4e−04
0 5000 10000 15000 20000 25000 30000
Competitors = 4
0.00000
0.00015
0.00030
0 5000 10000 15000 20000 25000
Competitors = 5
0.00000
0.00015
0 5000 10000 15000 20000 25000 30000
Competitors = 6
0e+00
2e−04
4e−04
0 2000 4000 6000 8000 10000
Competitors = 7
0.00000
0.00010
0 10000 20000 30000 40000
Competitors = 8
0e+00
2e−04
4e−04
2000 4000 6000 8000 10000 12000
Competitors = 9
0.00000
0.00015
2000 4000 6000 8000 10000 12000

3.3 Obtain CDFs from densitiy function

```
invis.Map(\(inp, n){

  # integrate to obtain cdf
  cdf <- \(x) integrate(inp(x), lower = 0, upper = x)$value

}, bid_n_dens_2, 0:(length(bid_n_dens_2) - 1)) -> emp_cdfs
```