Computation of minimum informative sizes

Derumigny, A. , Guyonvarch Y. & Girard, L.

September 8, 2022

```r
K4 = 9
eps = 0.1
kappa = 0.99
C0 = 1
p = 2
n = 1:100000

df_design = expand.grid(continuity = c(TRUE, FALSE),
                        no_skewness = c(TRUE, FALSE))
df_design$iid = TRUE

df_minSample = lapply(
  X = 1:nrow(df_design),

  FUN = function(i){
    regularity = if(df_design$iid[i]) {list(kappa = kappa)} else {list(C0 = C0, p = p)}

    return (
      data.frame(
        n = n ,
        Bound = Bound_BE(setup = list(continuity = df_design$continuity[i],
                                      iid = df_design$iid[i],
                                      no_skewness = df_design$no_skewness[i]),
                         n = n, K4 = K4, regularity = regularity,
                         eps = 0.1),

        eps = eps,
        continuity = df_design$continuity[i],
        iid = df_design$iid[i],
        no_skewness = df_design$no_skewness[i] ) )
}) %>%
  bind_rows() %>%
  mutate(continuity_ = paste0("continuity = ", continuity),
         iid_ = paste0("iid = ", iid),
         no_skewness_ = paste0("no_skewness = ", no_skewness))

table_to_print = lapply(X = c(0.1, 0.05, 0.01),
       FUN = function (alpha)
       {
         df = df_minSample %>%
           group_by(continuity, no_skewness) %>%
           filter(Bound < alpha) %>%
```

```
        summarise(n_max = min(n)-1) ;
        df$alpha = alpha
        return (df)
    } ) %>%
  bind_rows() %>%
  ungroup() %>%
  pivot_wider(names_from = all_of("alpha") ,
            values_from = "n_max") %>%
  mutate(name = case_when(
    !continuity & !no_skewness ~ "Thm. \\ref{thm:nocont_choiceEps}" ,
    !continuity & no_skewness ~ "Thm. \\ref{thm:nocont_choiceEps} unskewed" ,
    continuity & !no_skewness ~ "Cor. \\ref{cor:improvement_iid_case}" ,
    continuity & no_skewness ~ "Cor. \\ref{cor:improvement_iid_case} unskewed") ) %>%
  select(all_of(c("name", "0.1", "0.05", "0.01")))
```

```
## `summarise()` has grouped output by 'continuity'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'continuity'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'continuity'. You can override using the
## `.groups` argument.
```

```
gt::gt(table_to_print) %>%
  gt::fmt_passthrough(columns = all_of("name"), escape = FALSE) %>%
  gt::as_latex() %>% cat()
```

```
## \begin{longtable}{lrrr}
## \toprule
## name & 0.1 & 0.05 & 0.01 \\
## \midrule
## Thm. \ref{thm:nocont_choiceEps} & 2339 & 6705 & 55894 \\
## Thm. \ref{thm:nocont_choiceEps} unskewed & 443 & 1229 & 17934 \\
## Cor. \ref{cor:improvement_iid_case} & 1468 & 4069 & 27945 \\
## Cor. \ref{cor:improvement_iid_case} unskewed & 375 & 474 & 1062 \\
## \bottomrule
## \end{longtable}
```