

Plots

Eduardo Yuki Yada

Imports

```
library(tidyverse)
library(yaml)

options(dplyr.summarise.inform = FALSE)
```

Loading data

```
load('dataset/processed_data.RData')
load('dataset/processed_dictionary.RData')

columns_list <- yaml.load_file("./auxiliar/columns_list.yaml")

outcome_column <- params$outcome_column

df[columns_list$outcome_columns] <- lapply(df[columns_list$outcome_columns], as.character)
df[columns_list$outcome_columns] <- lapply(df[columns_list$outcome_columns], as.integer)
```

Plots

```
library(lubridate)

medianWithoutNA <- function(x) {
  median(x[which(!is.na(x))])
}

k = 6

df[columns_list$outcome_columns] <- lapply(df[columns_list$outcome_columns], as.numeric)

for (column in setdiff(columns_list$numerical_columns, columns_list$outcome_columns)){
  if (mean(is.na(df[[column]])) > 0.5) next

  df %>%
    mutate(rounded_date_admission_t0 = lubridate::floor_date(date_admission_t0,
                                                             'month')) %>%

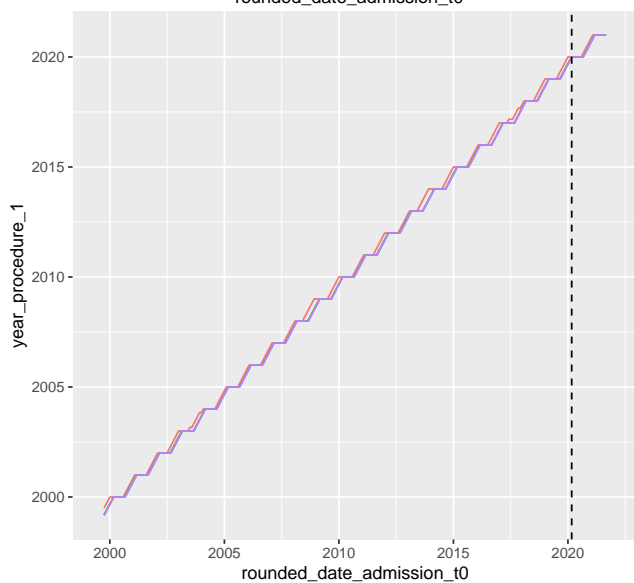
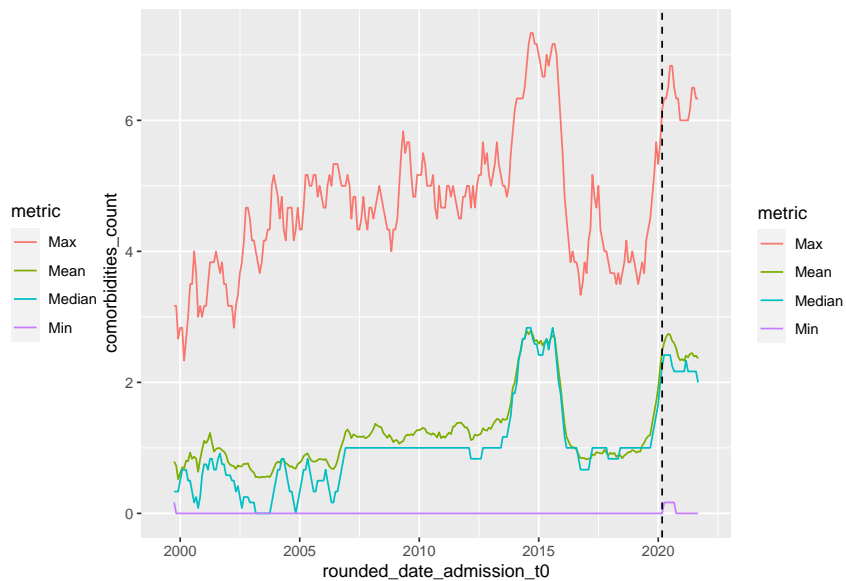
    group_by(rounded_date_admission_t0) %>%
    summarise('Mean' = mean(!sym(column), na.rm = T),
              'Min' = min(!sym(column), na.rm = T),
              'Median' = medianWithoutNA(!sym(column)),
              'Max' = max(!sym(column), na.rm = T)) %>%
    ungroup %>%
    arrange(rounded_date_admission_t0) %>%
    mutate(Mean = zoo::rollmean(Mean, k = k, fill = NA),
           Min = zoo::rollmean(Min, k = k, fill = NA),
           Median = zoo::rollmean(Median, k = k, fill = NA),
           Max = zoo::rollmean(Max, k = k, fill = NA)) %>%
```

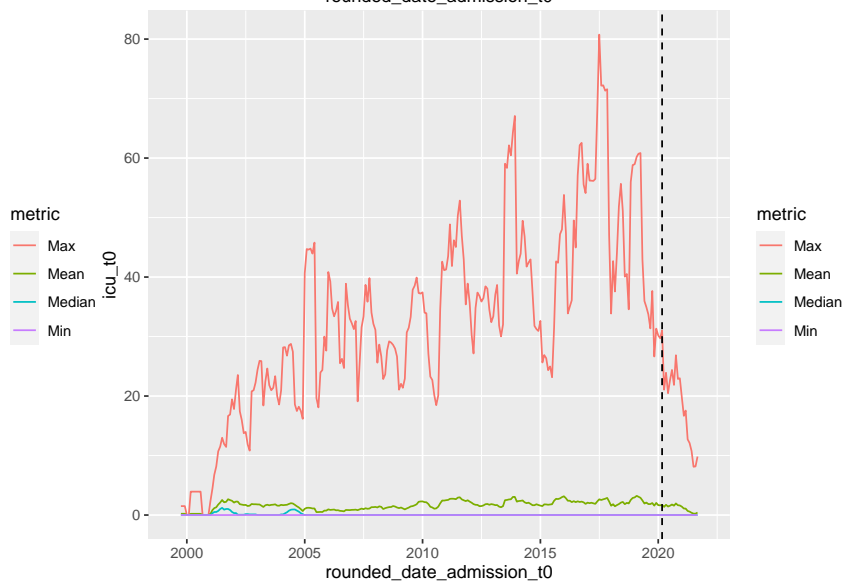
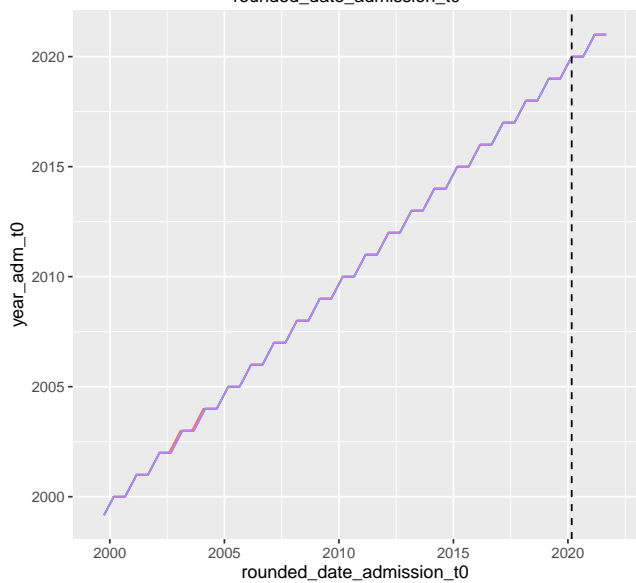
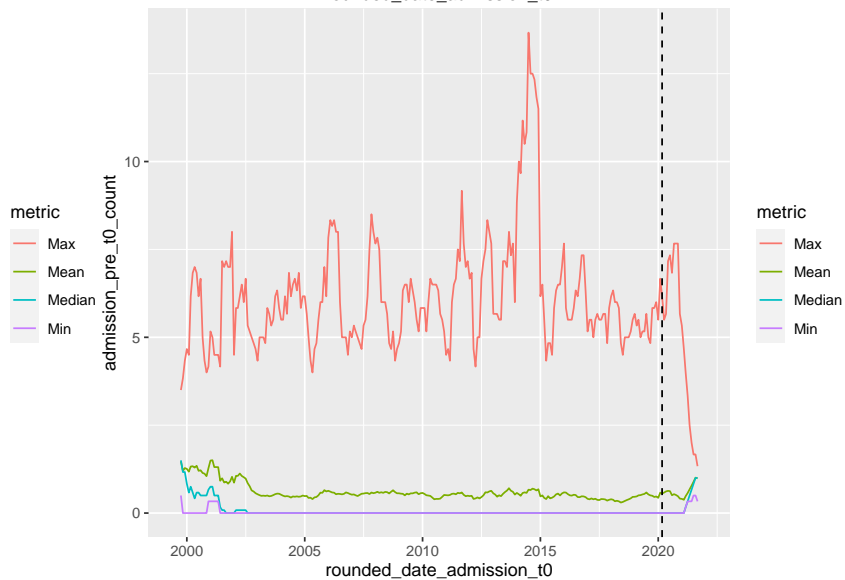
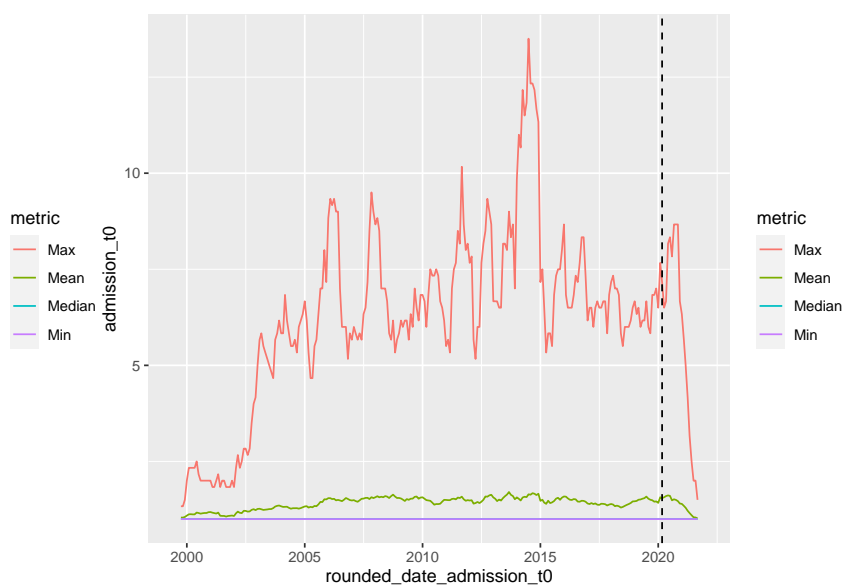
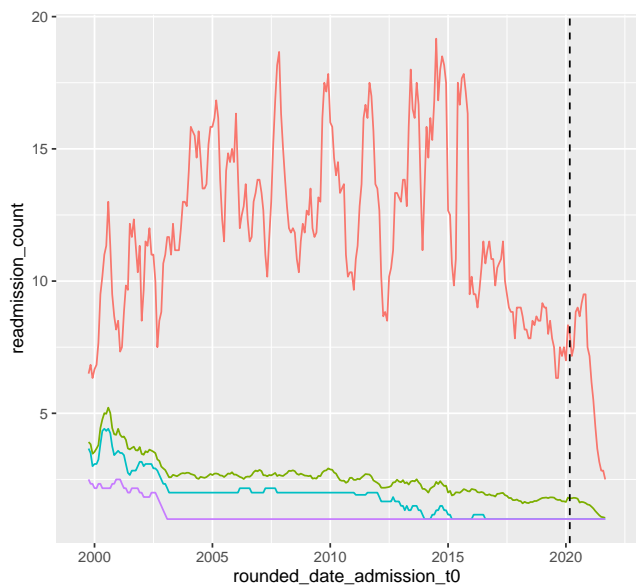
```

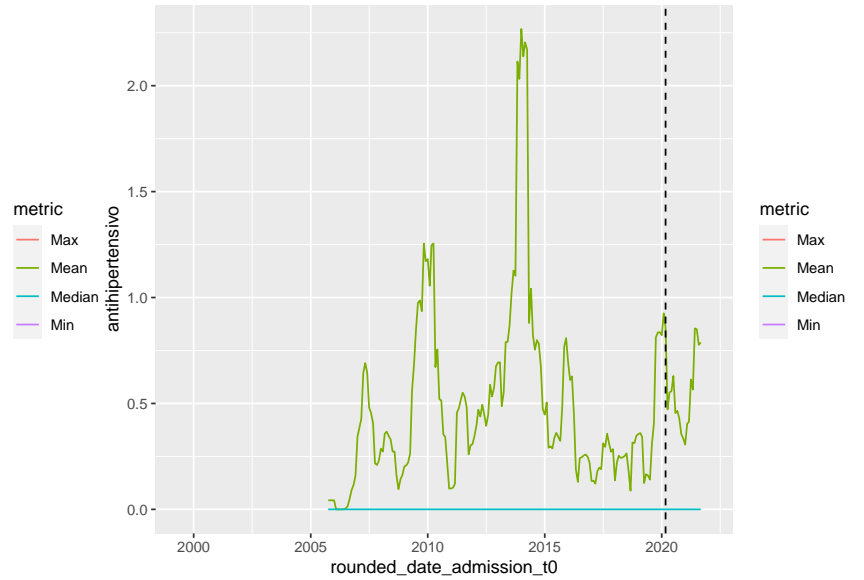
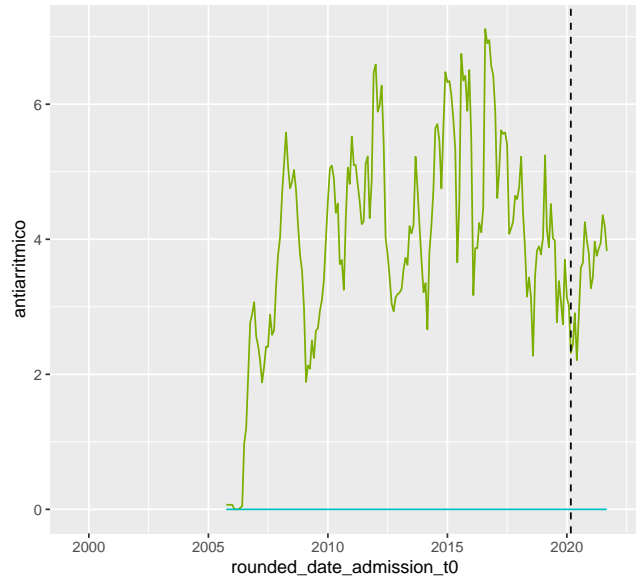
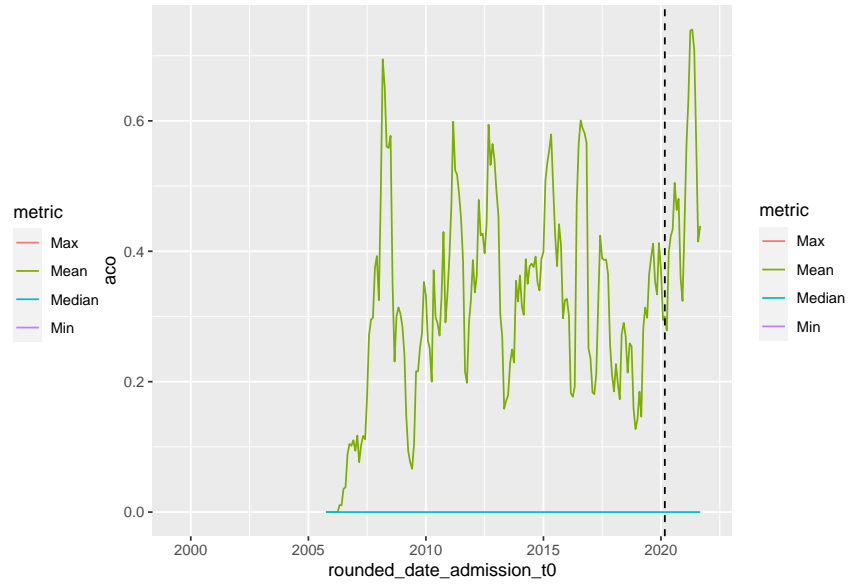
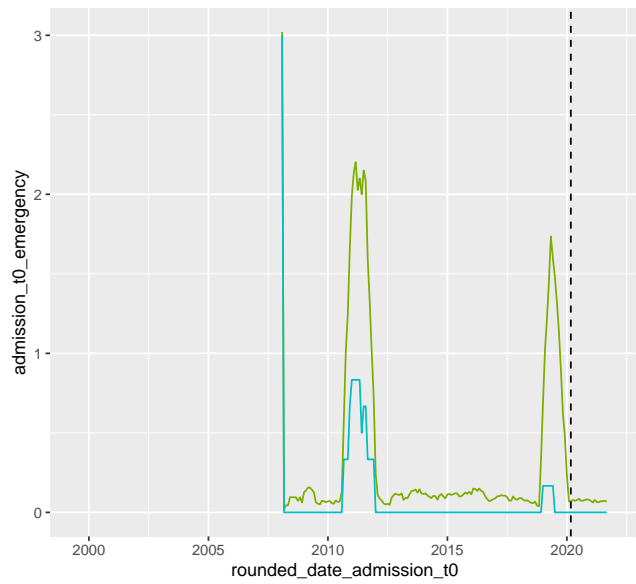
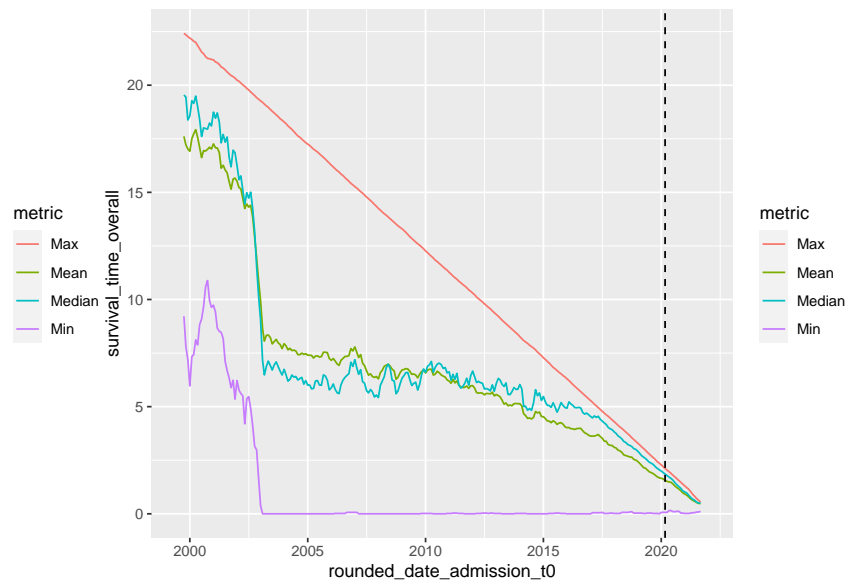
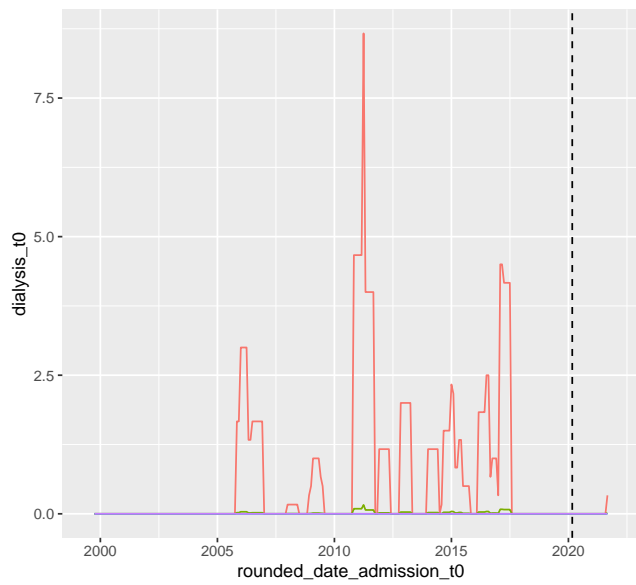
pivot_longer(!rounded_date_admission_t0, names_to = 'metric') %>%
ggplot(aes(x = rounded_date_admission_t0, y = value, color = metric)) +
  geom_line() +
  geom_vline(xintercept = lubridate::ymd('2020-03-01'), linetype = "dashed") +
  labs(y = column)-> p

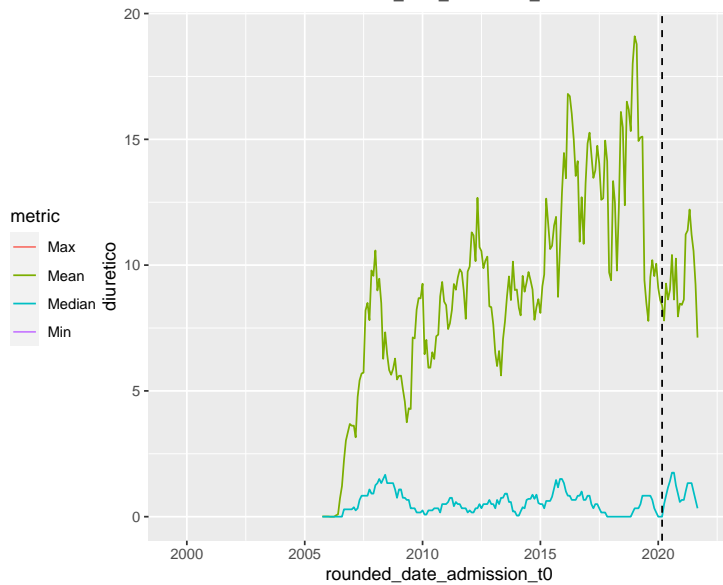
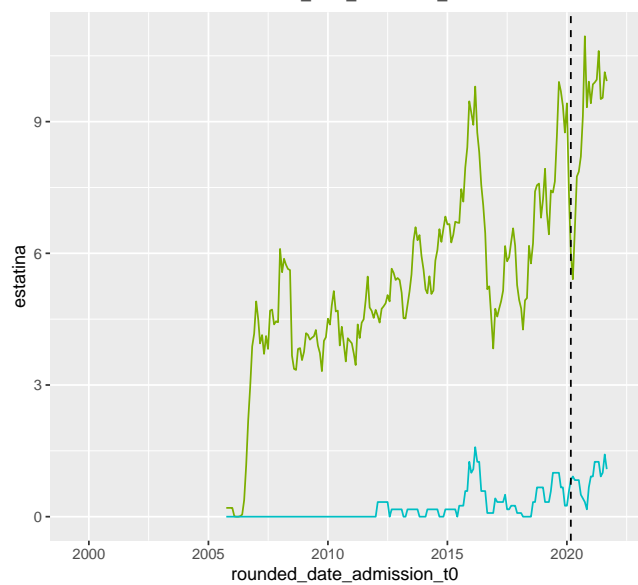
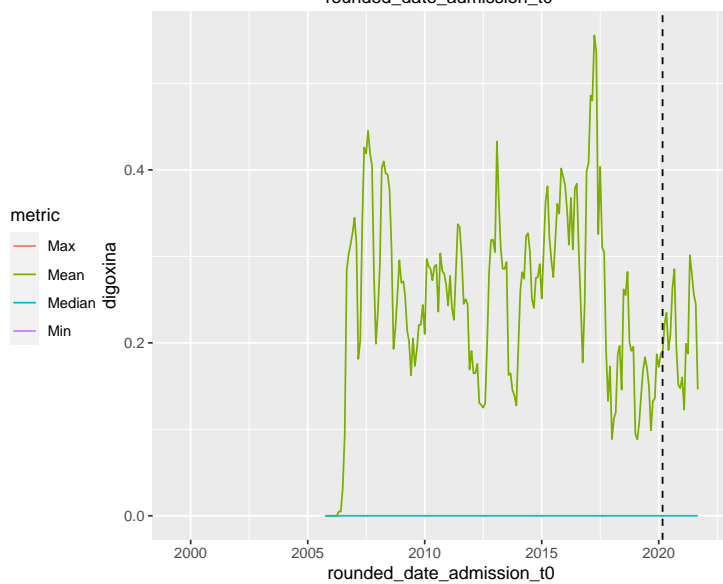
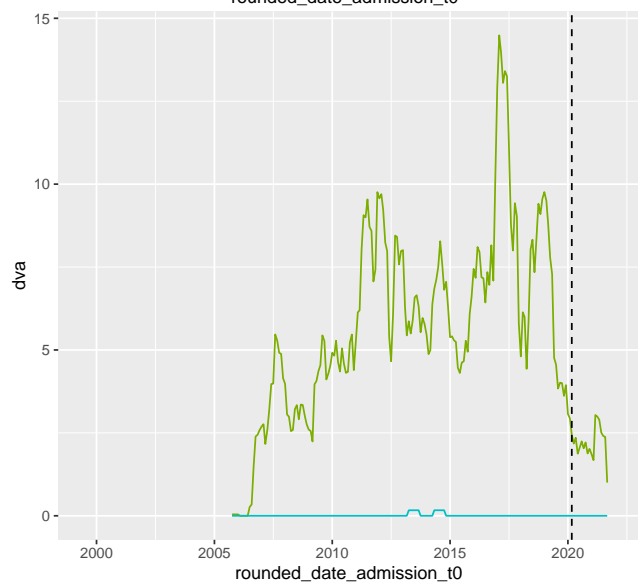
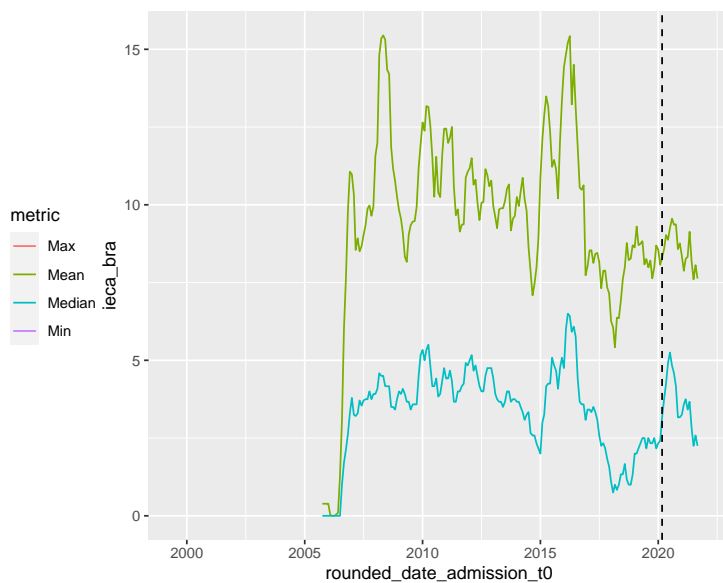
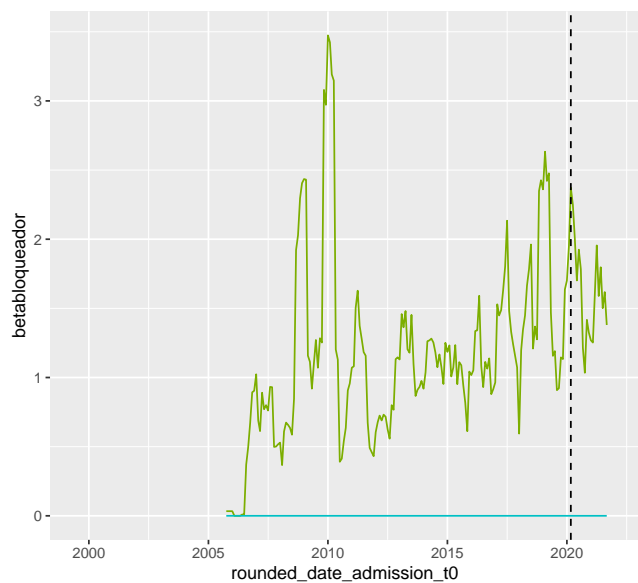
print(p)
}

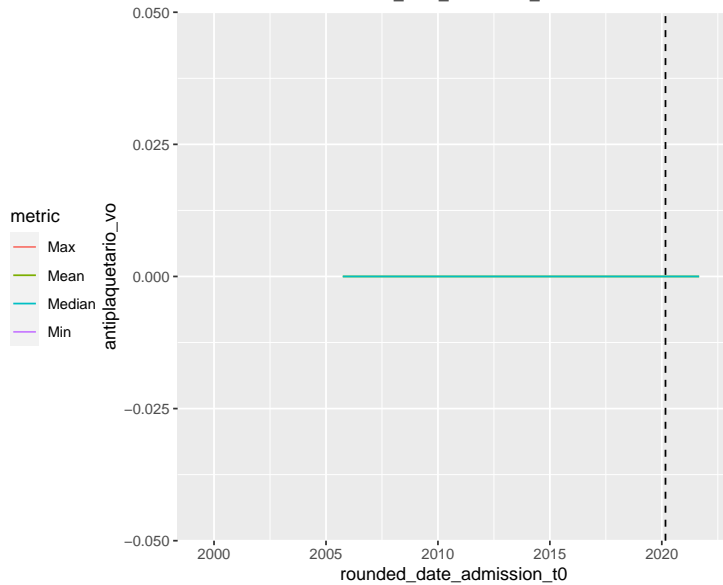
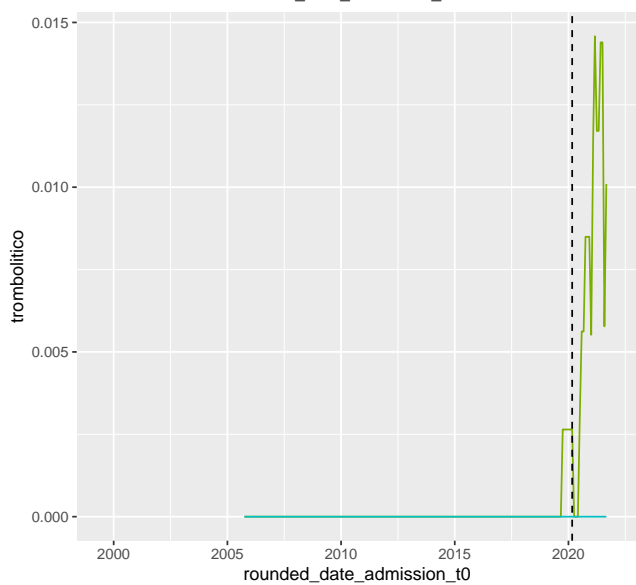
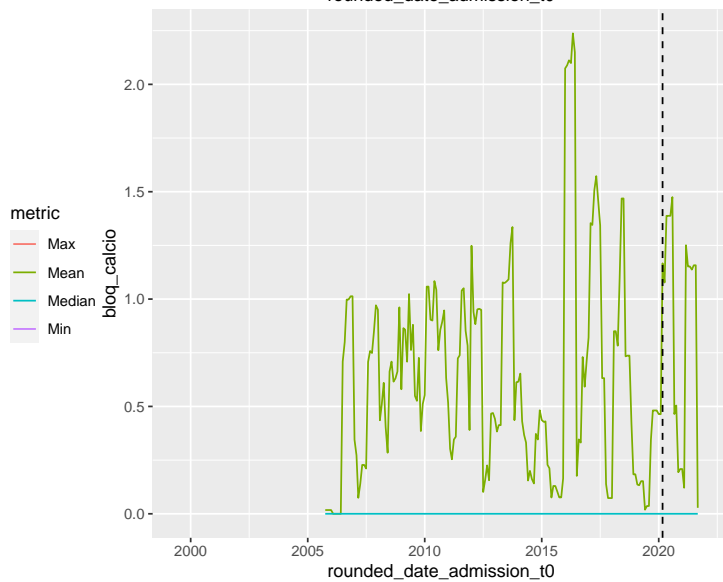
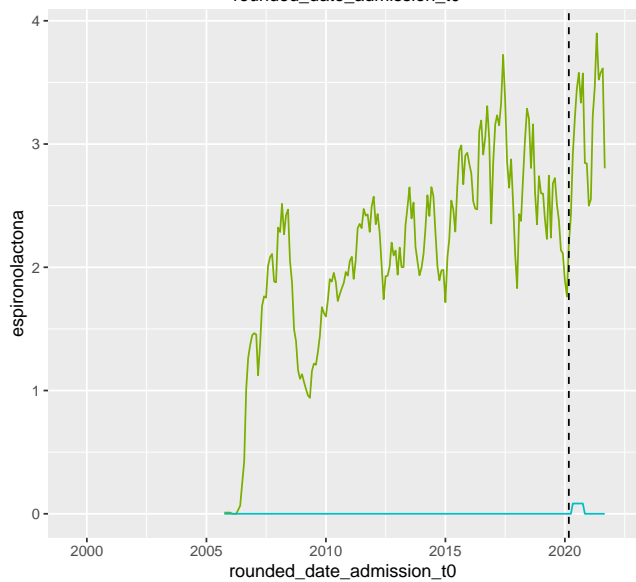
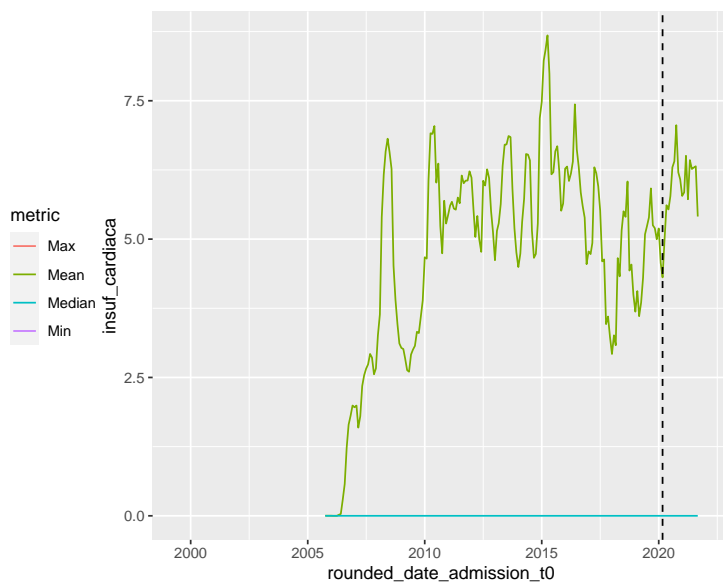
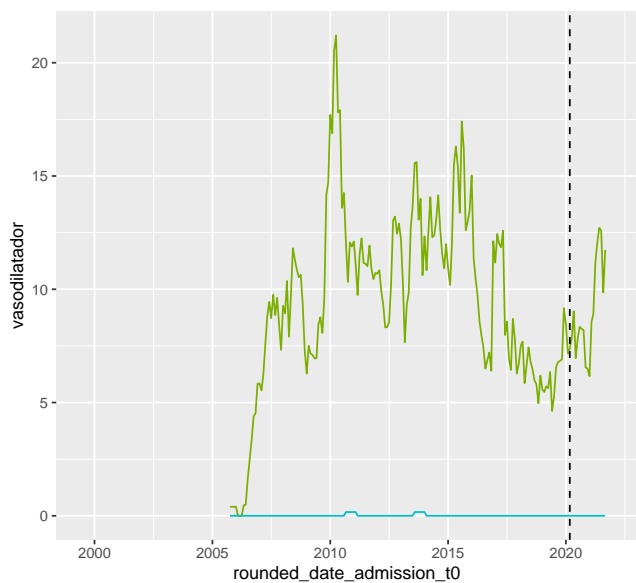
```

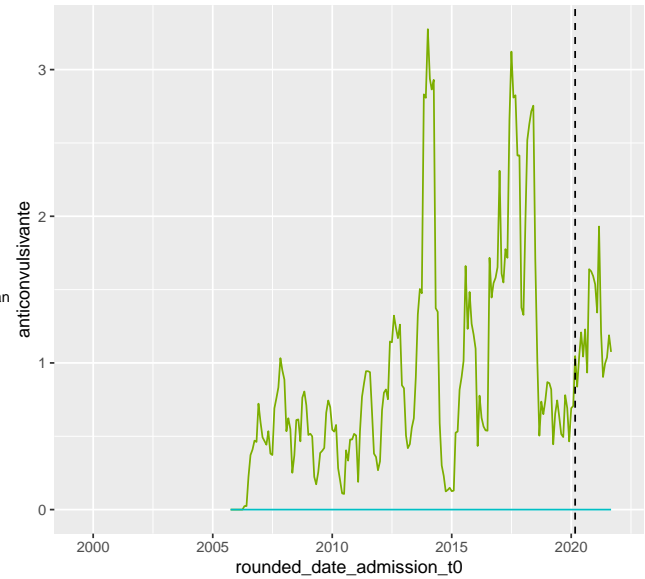
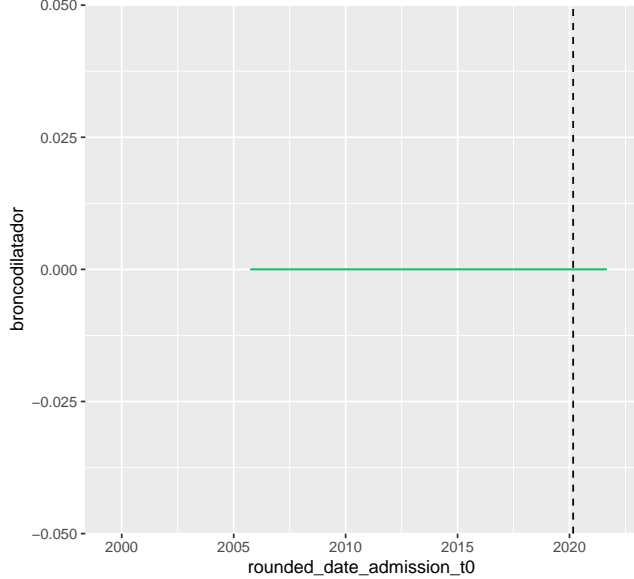
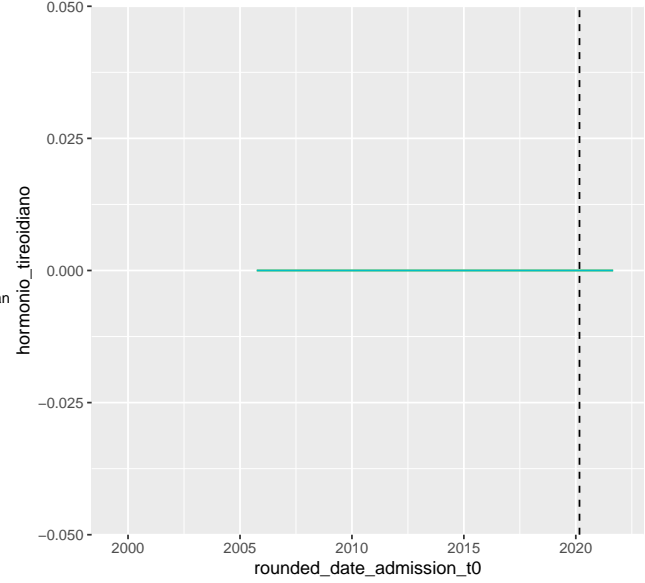
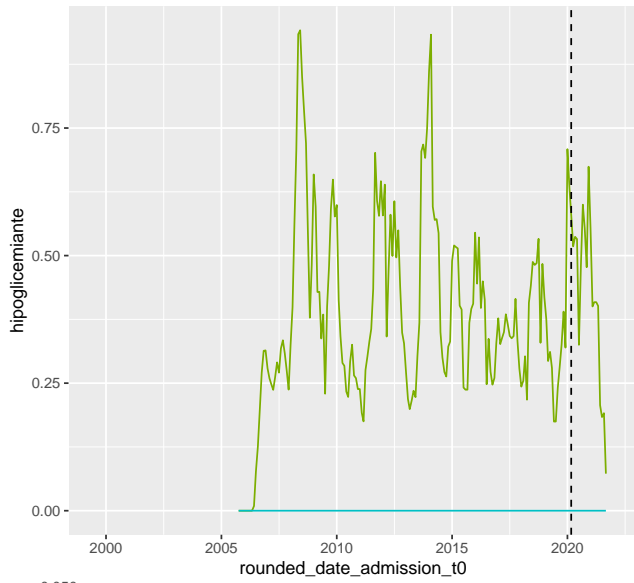
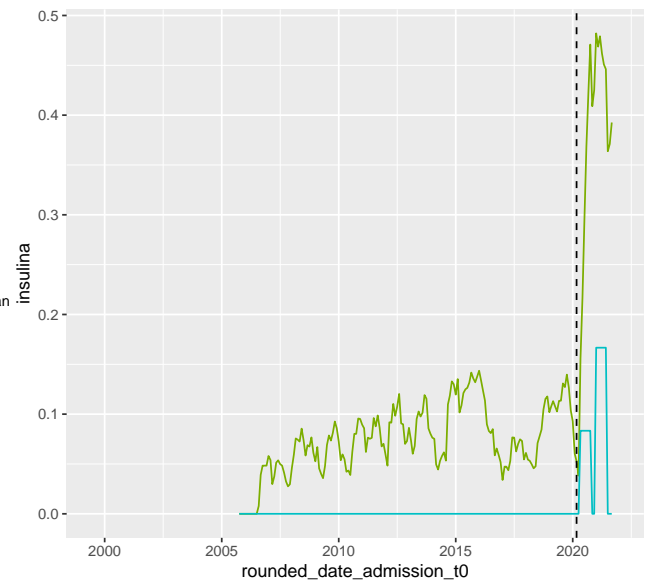
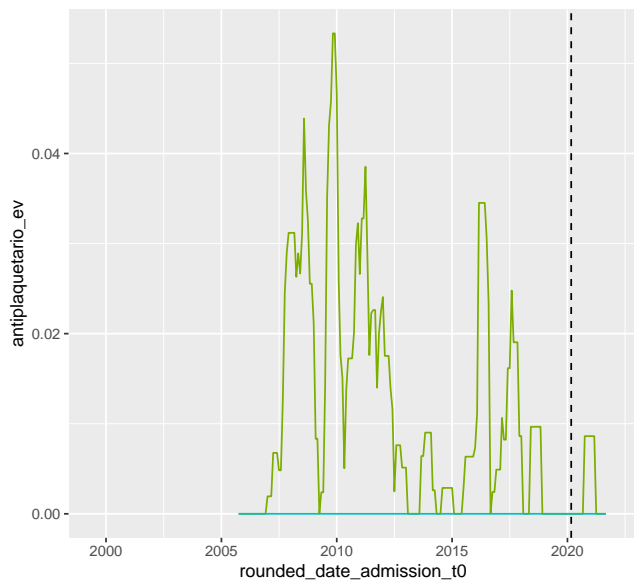


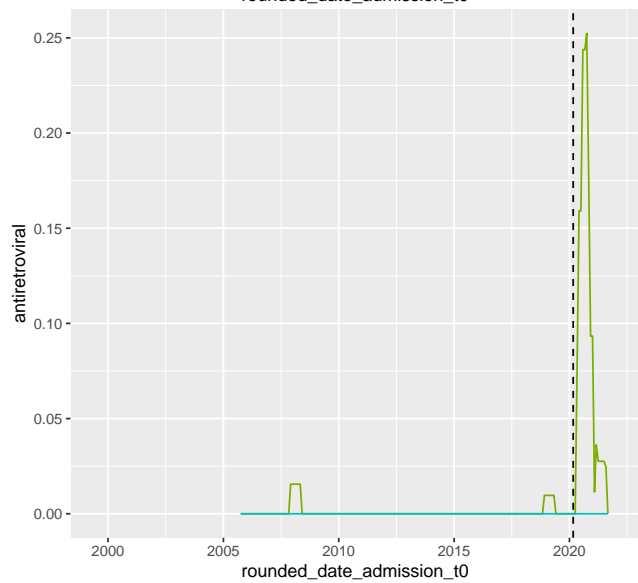
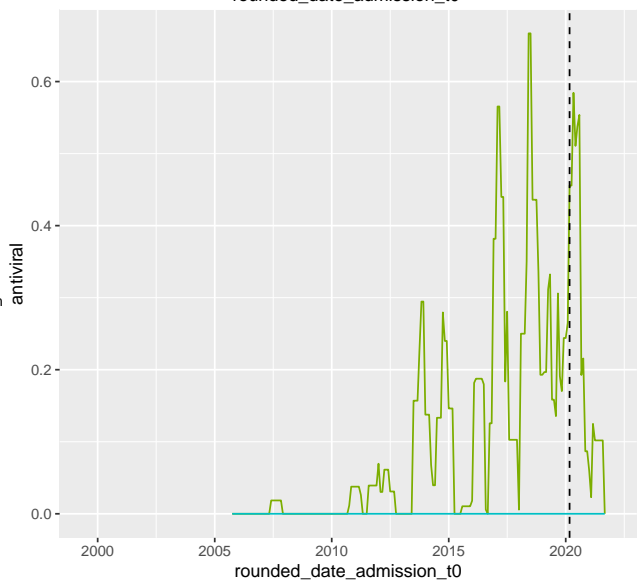
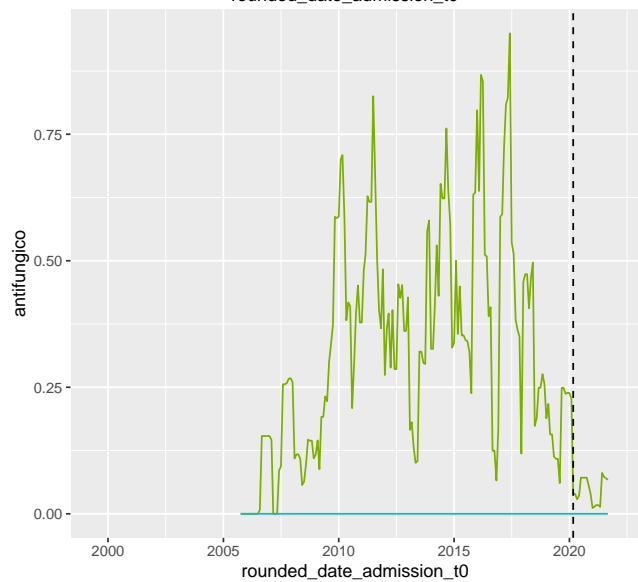
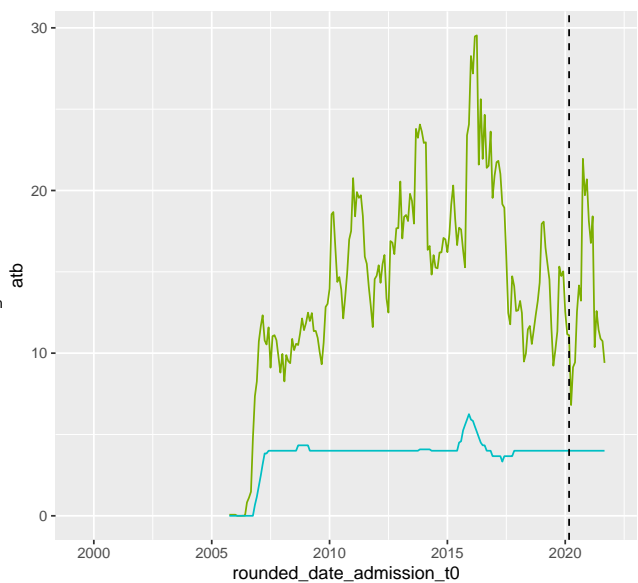
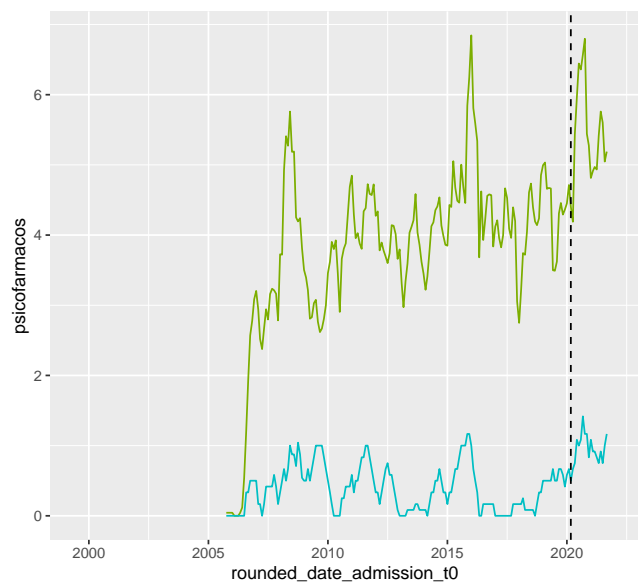


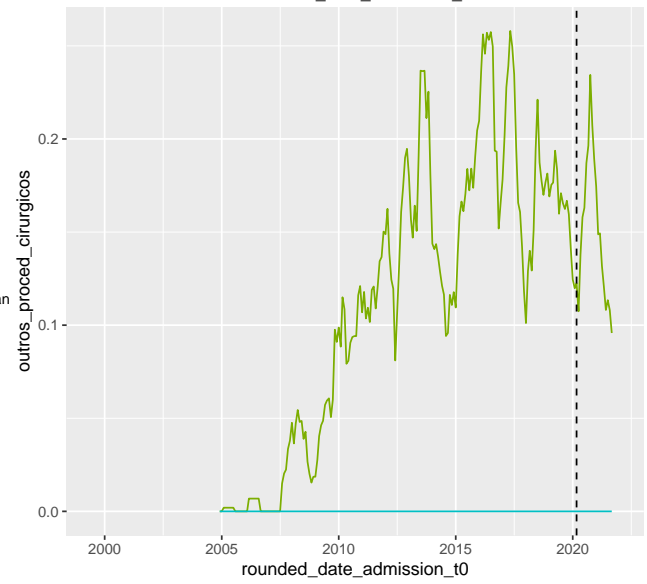
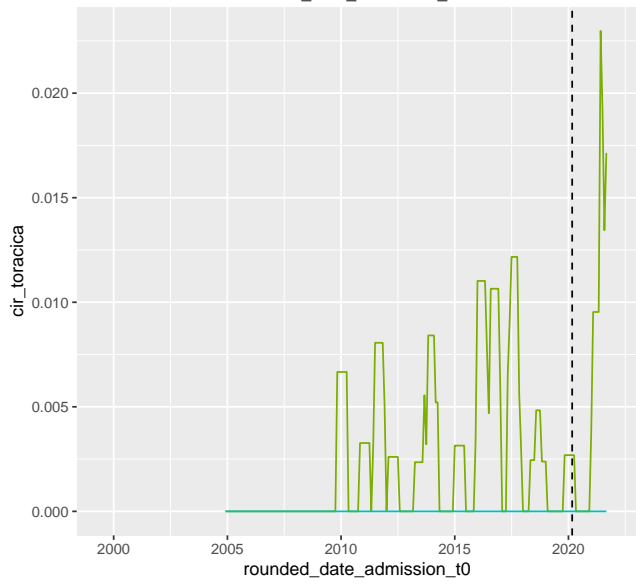
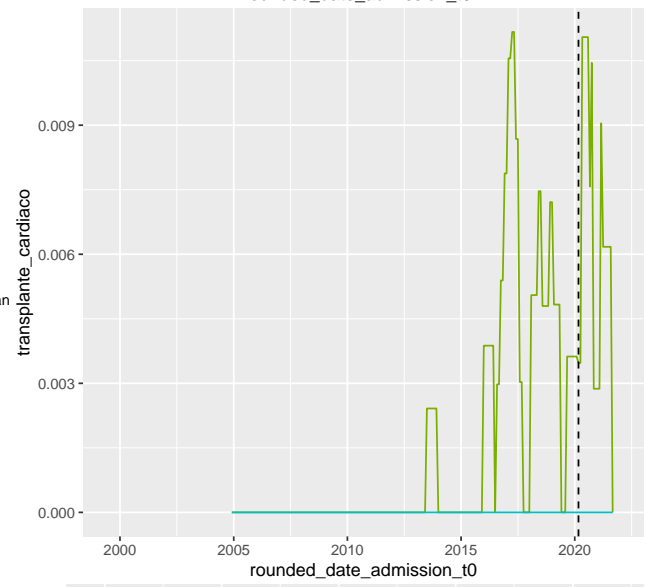
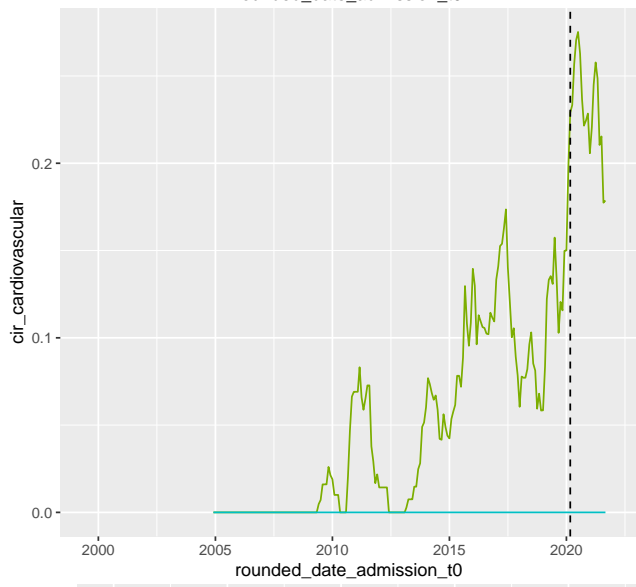
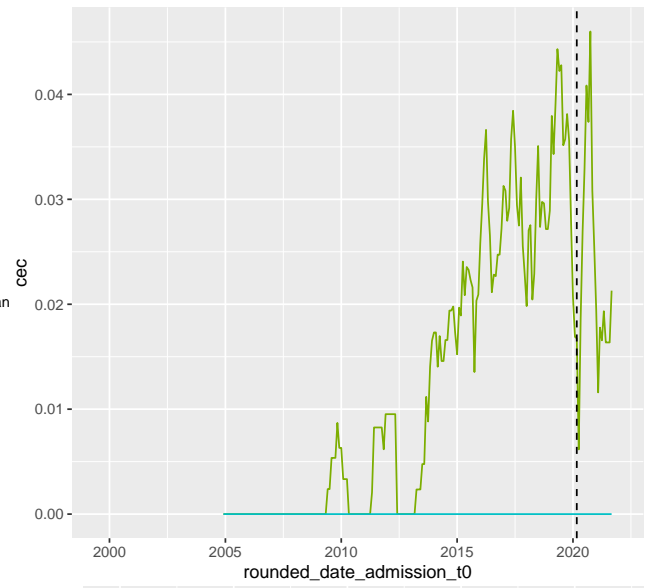
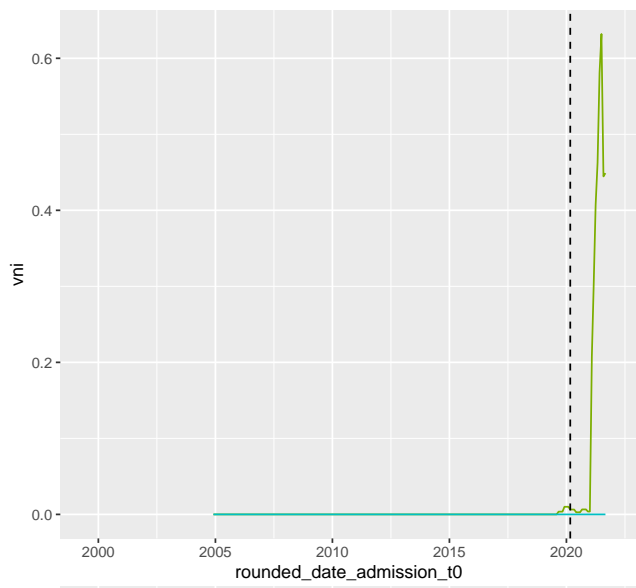


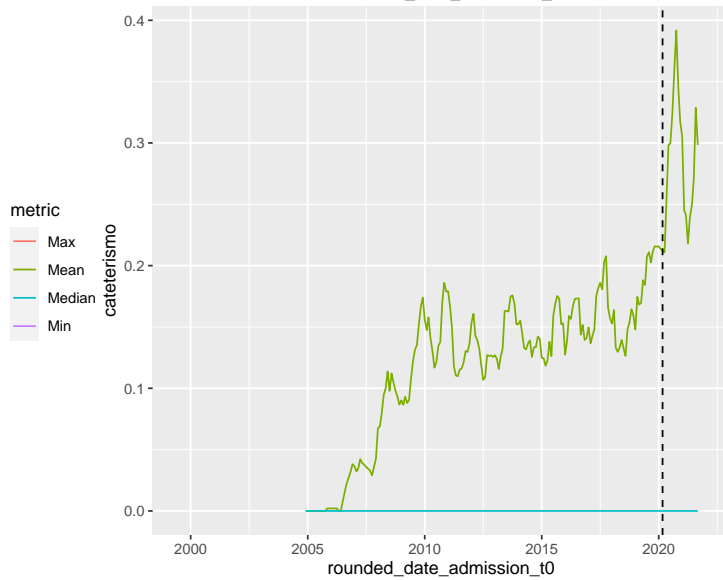
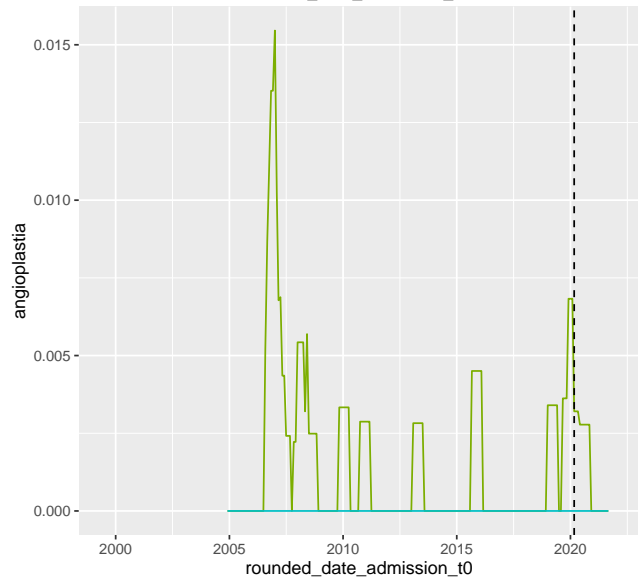
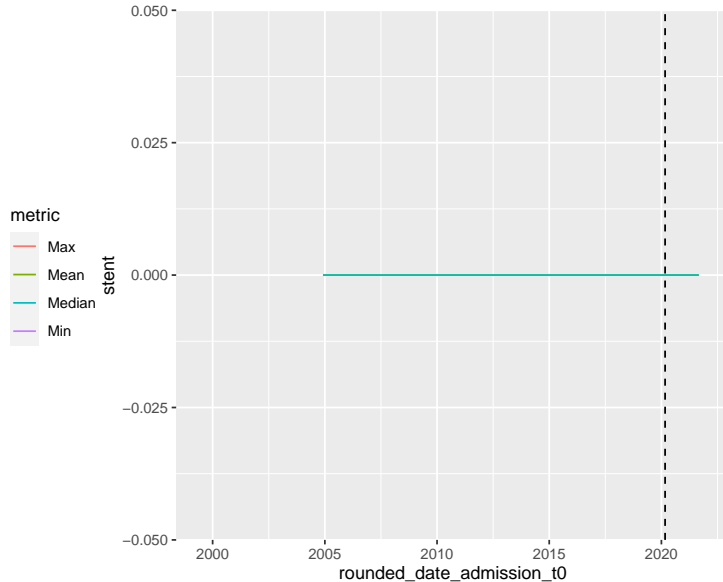
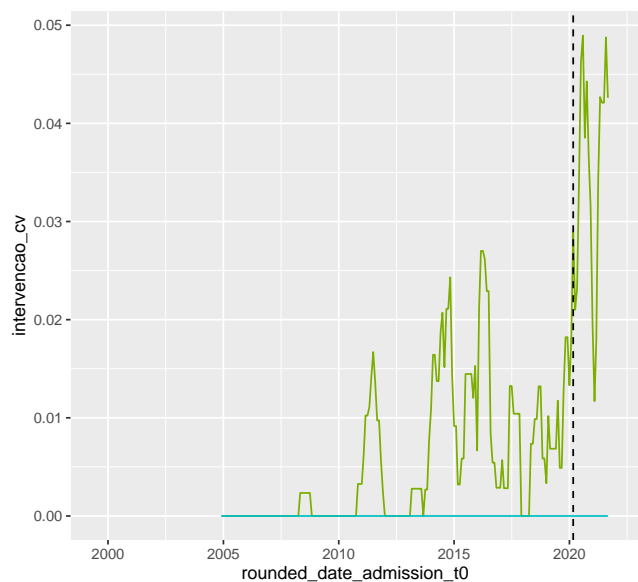
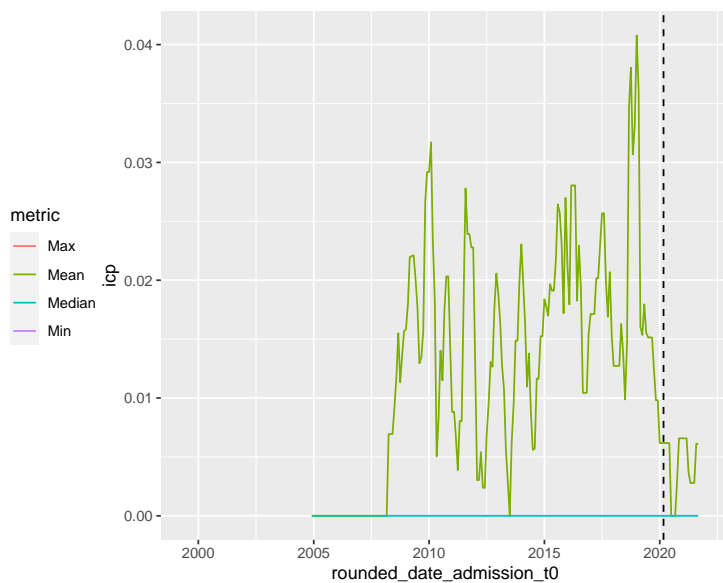
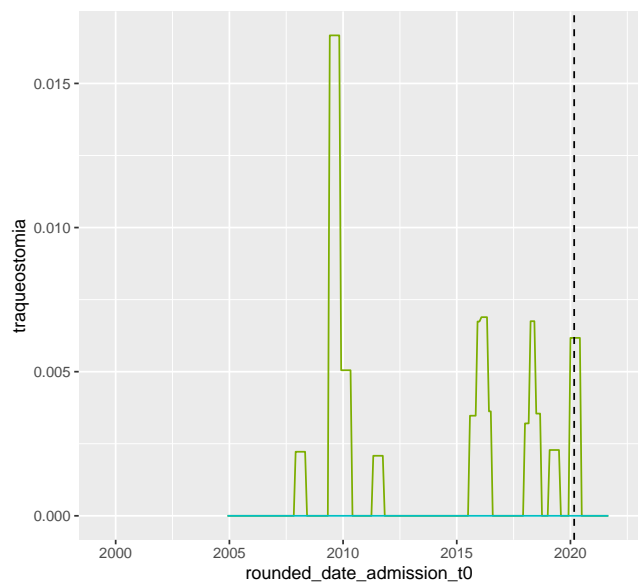


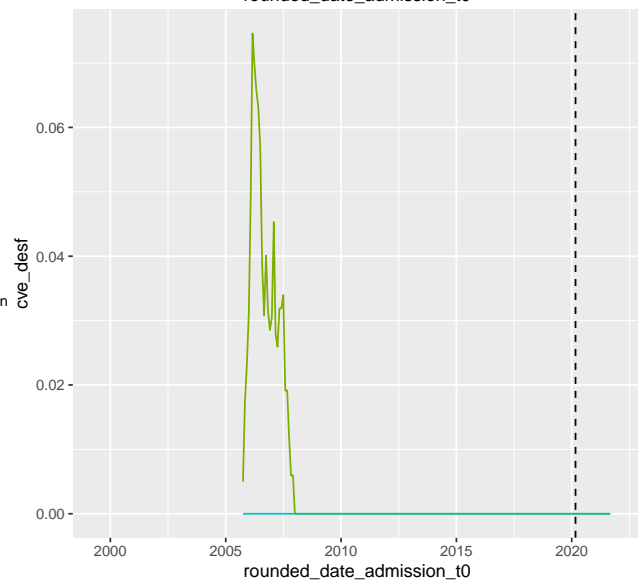
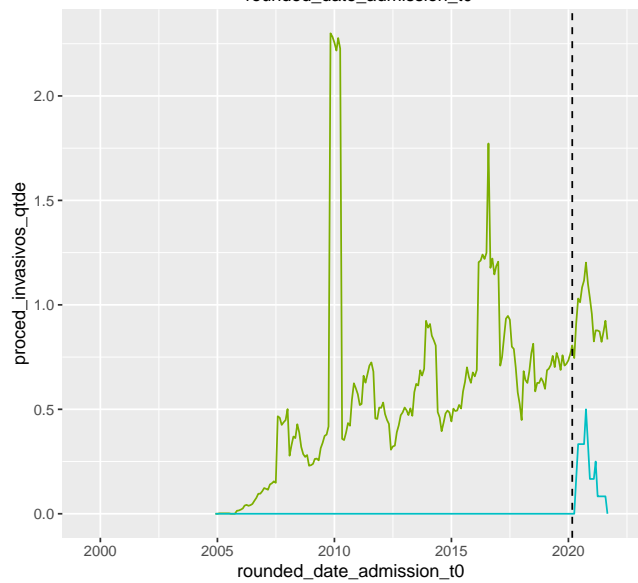
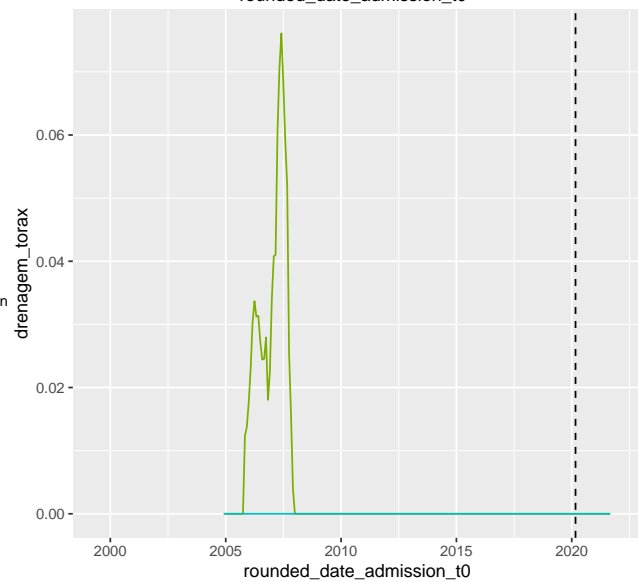
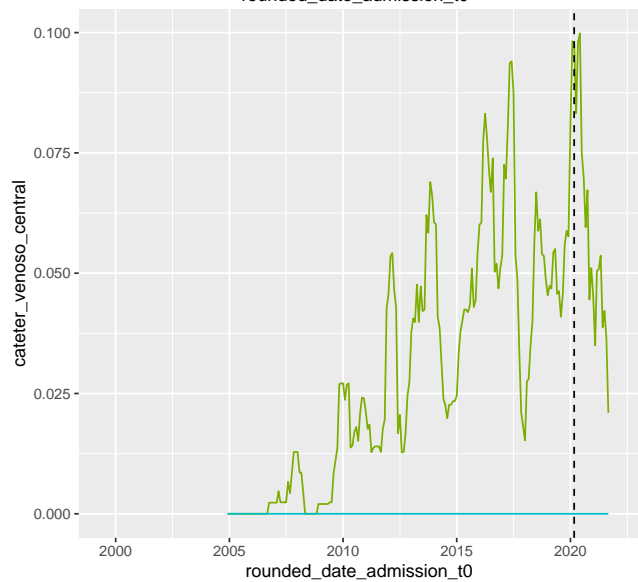
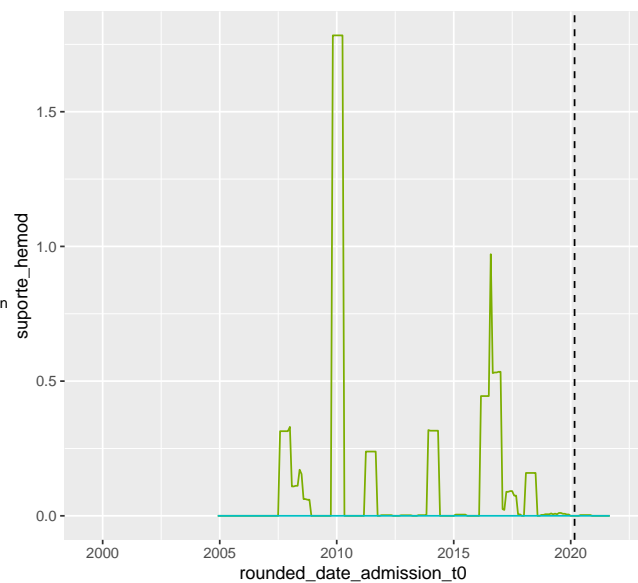
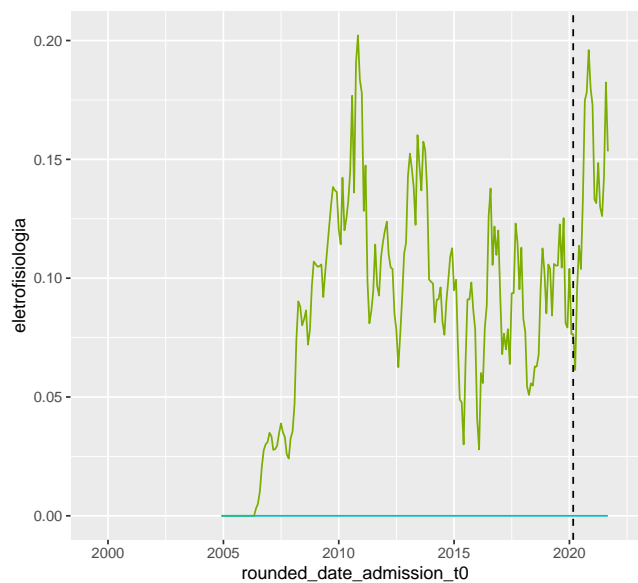


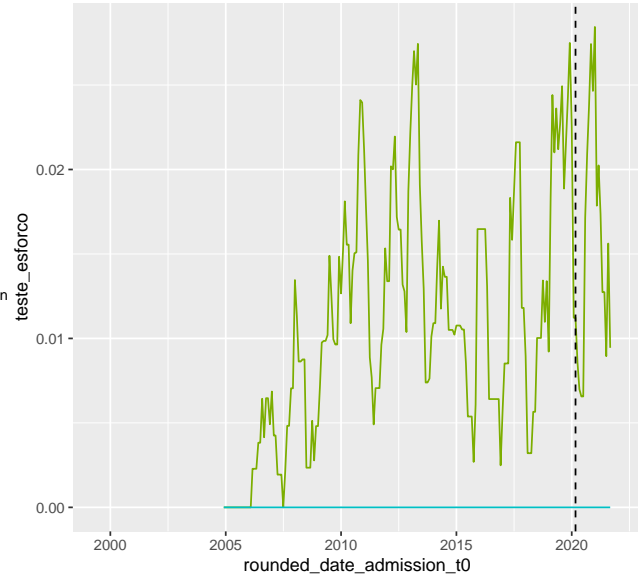
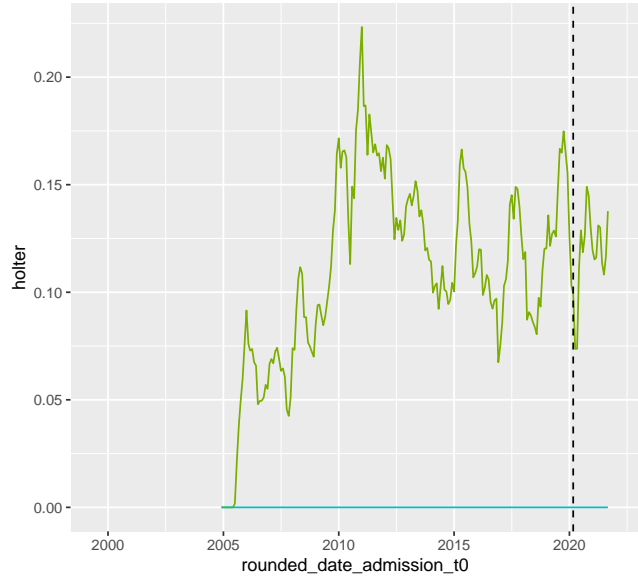
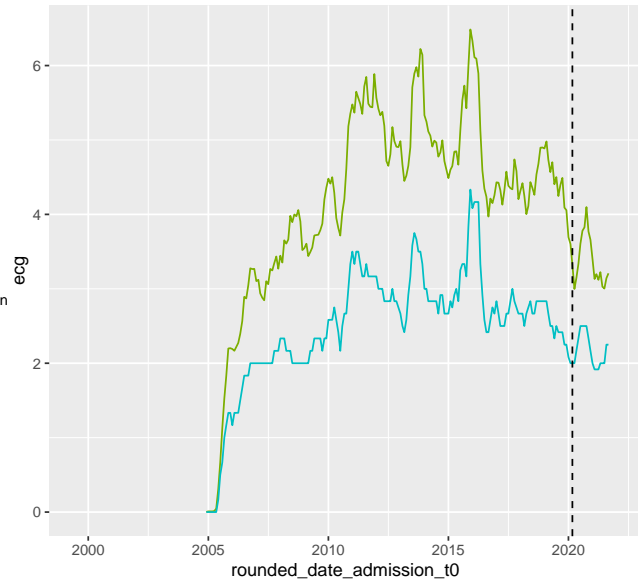
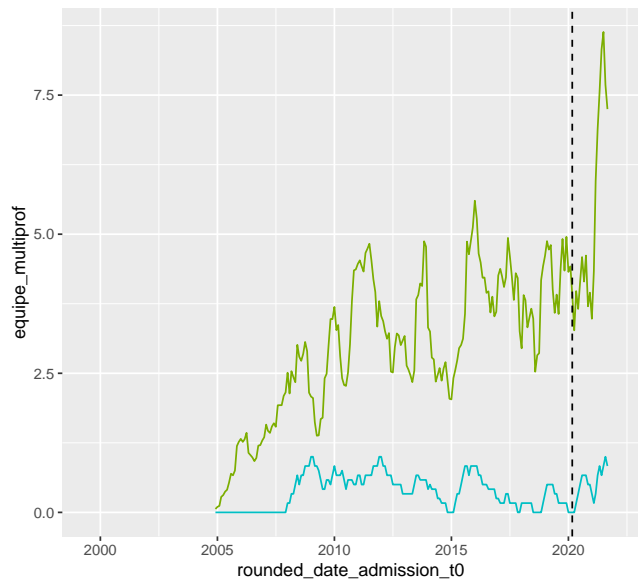
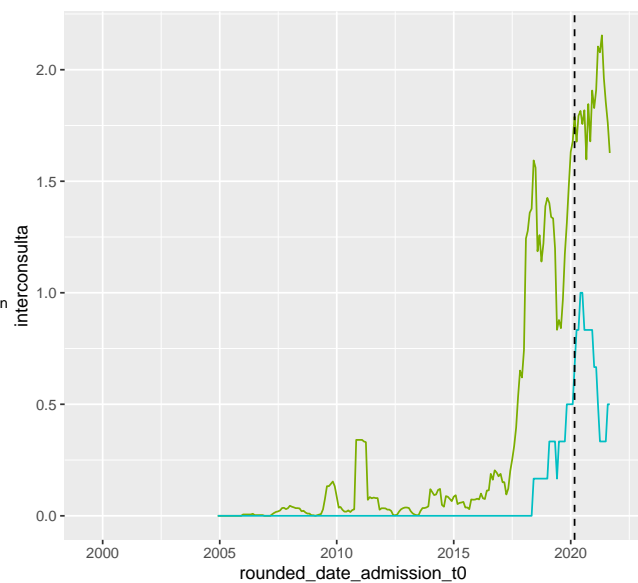
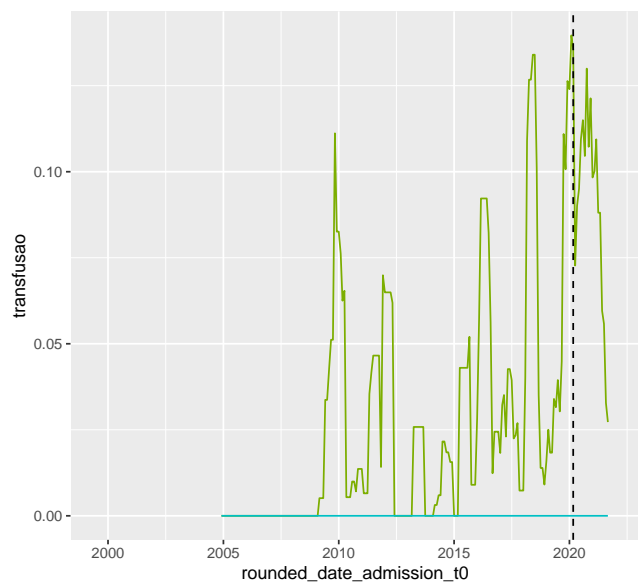


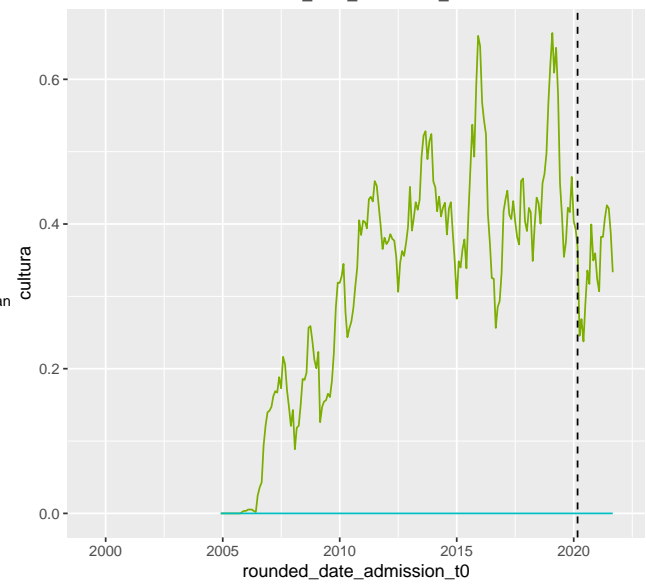
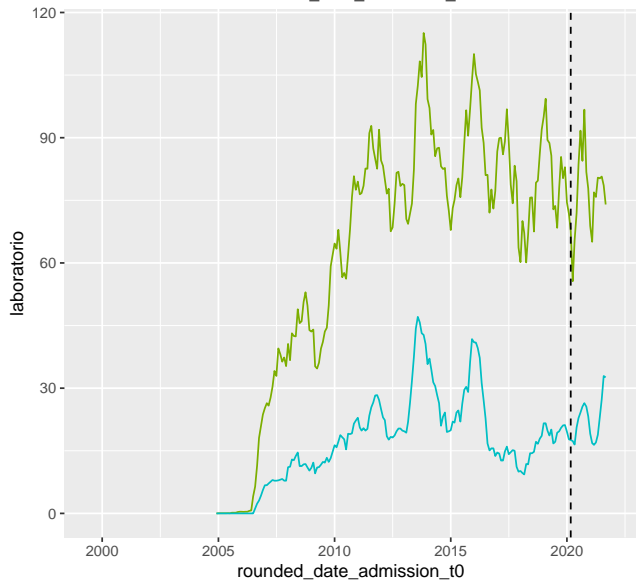
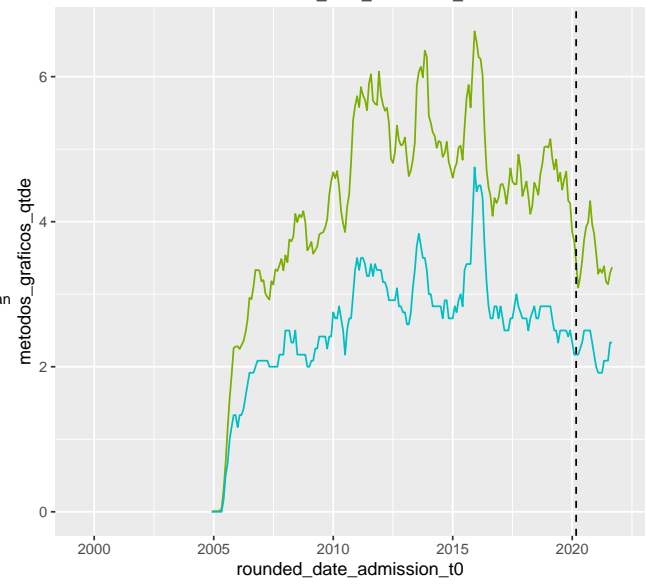
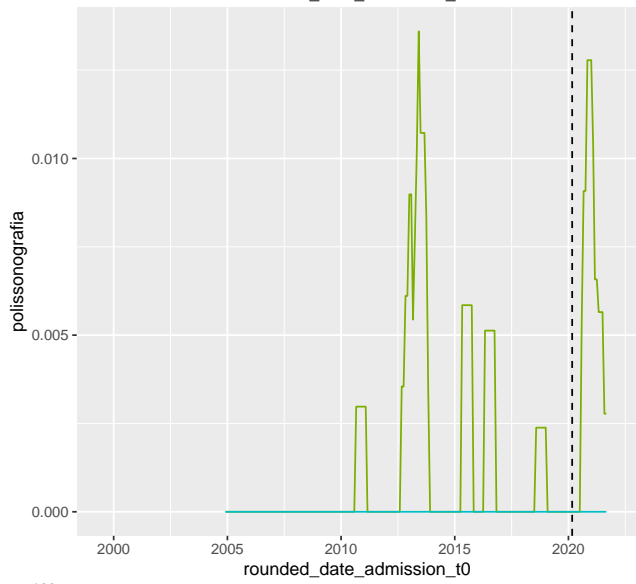
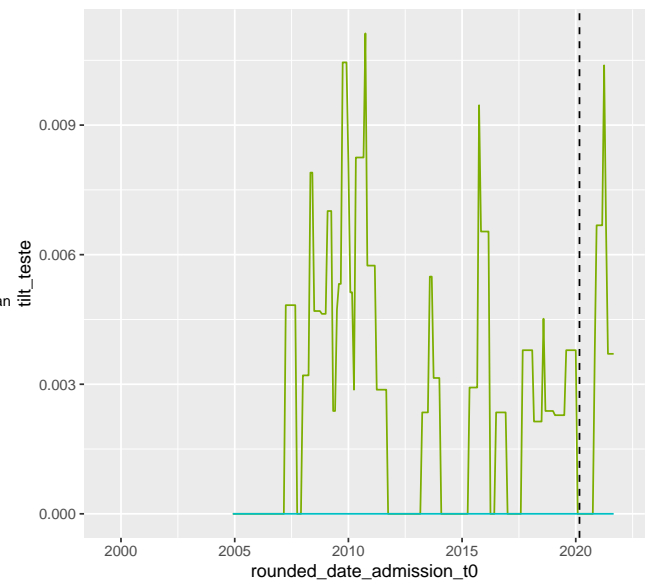
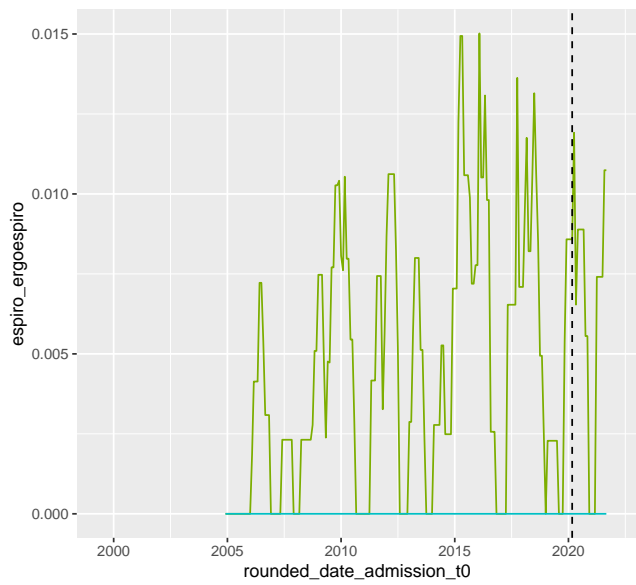


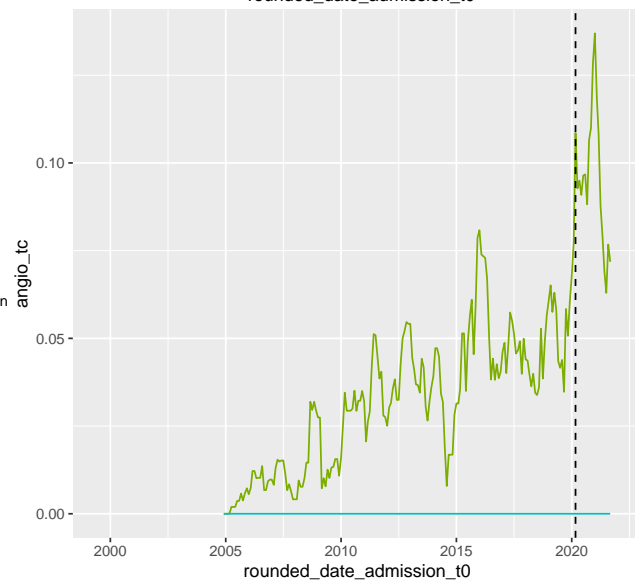
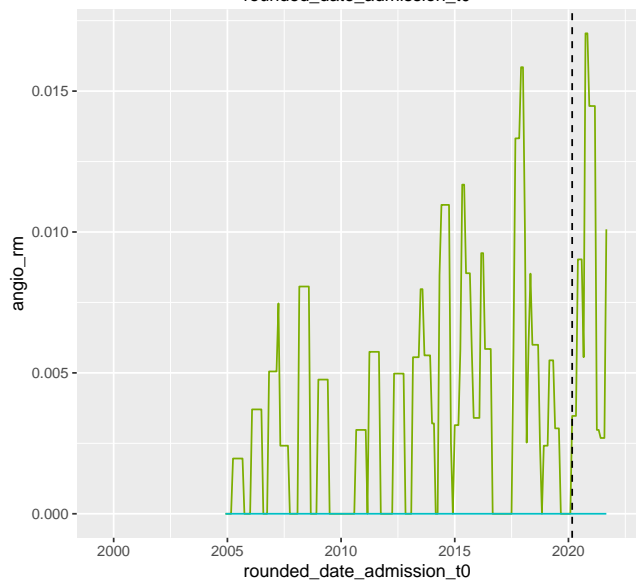
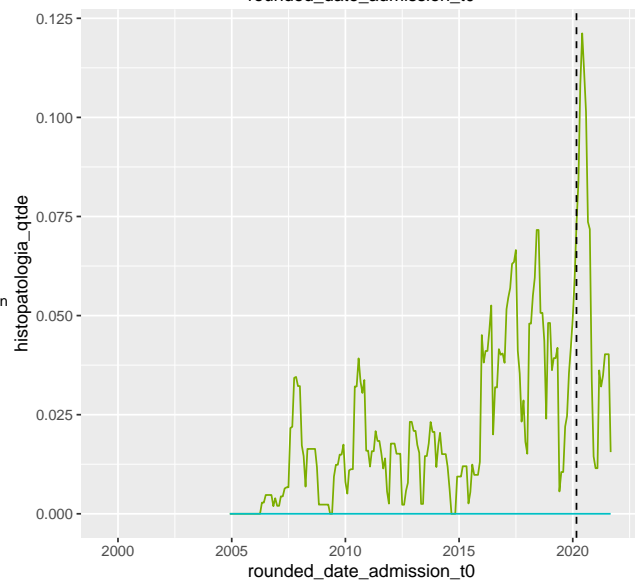
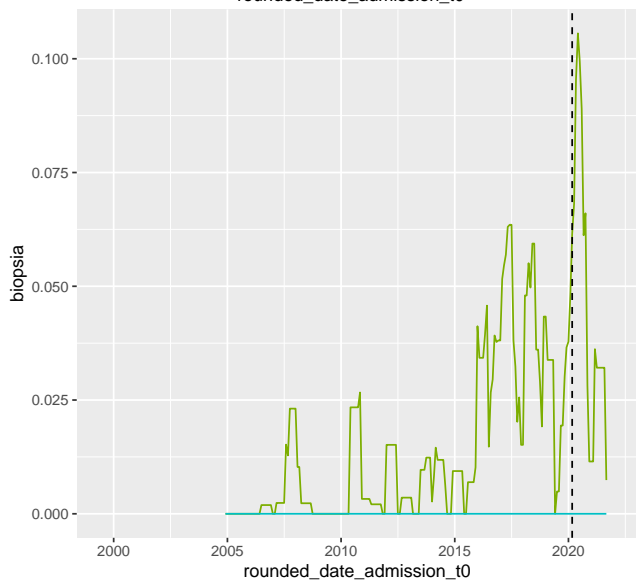
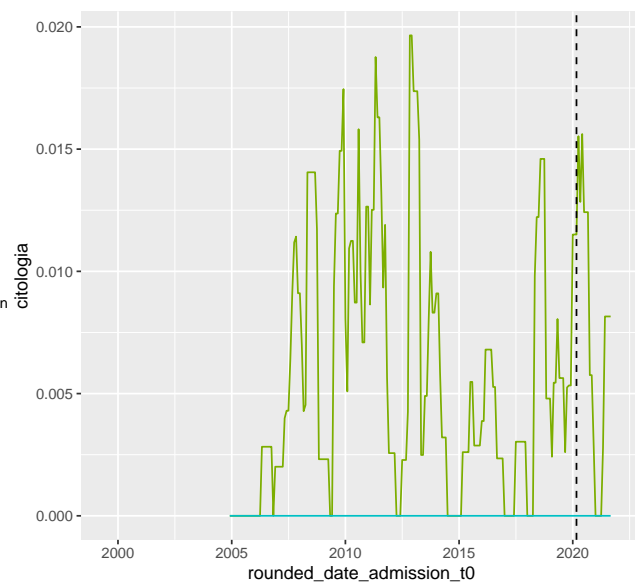
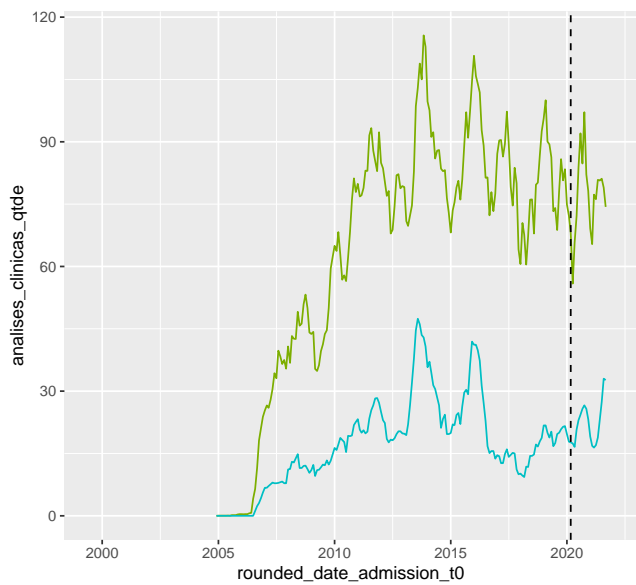


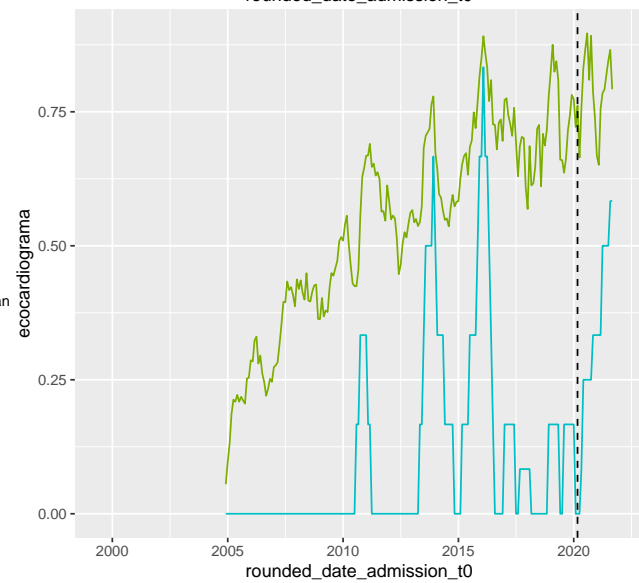
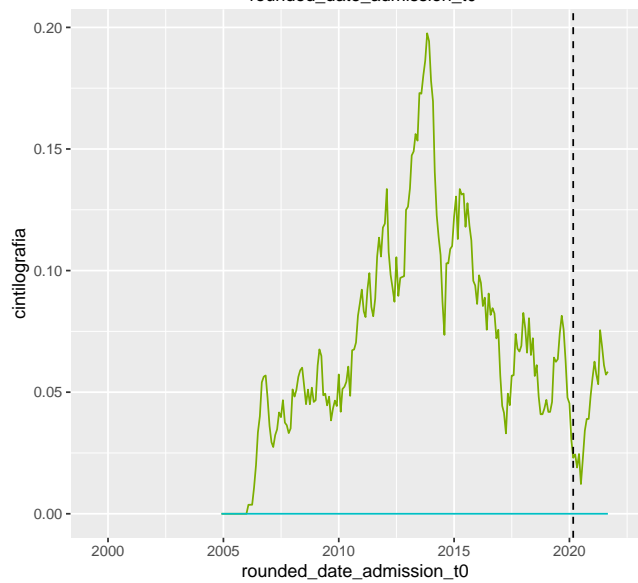
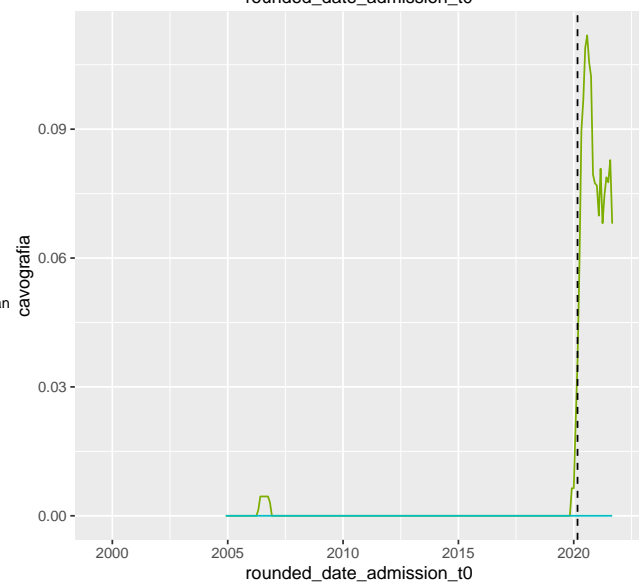
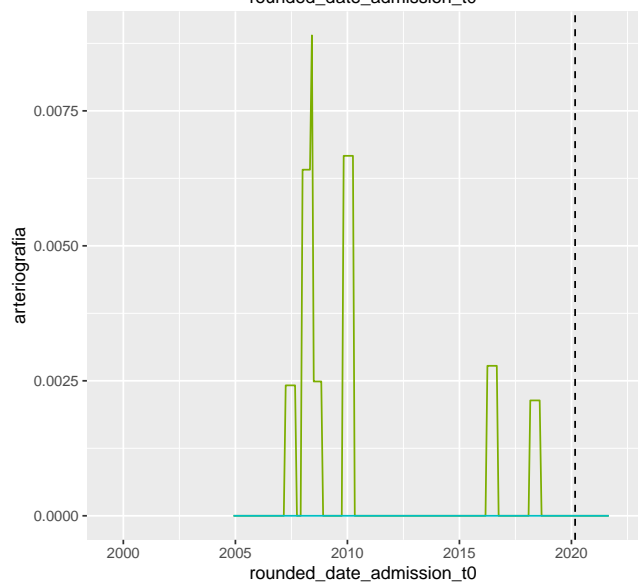
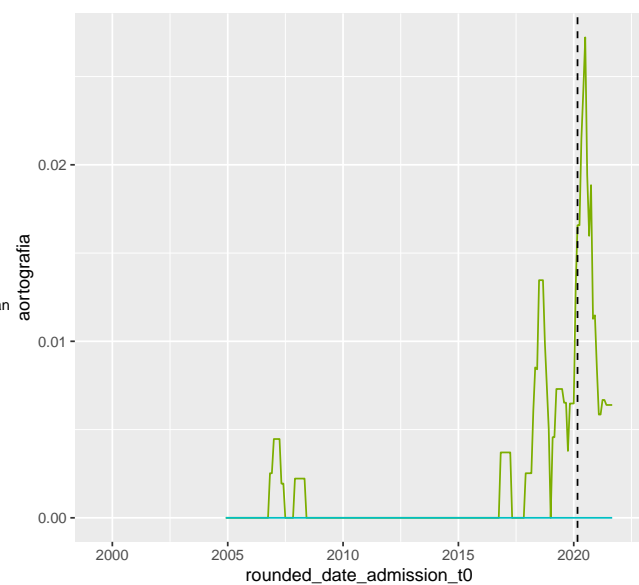
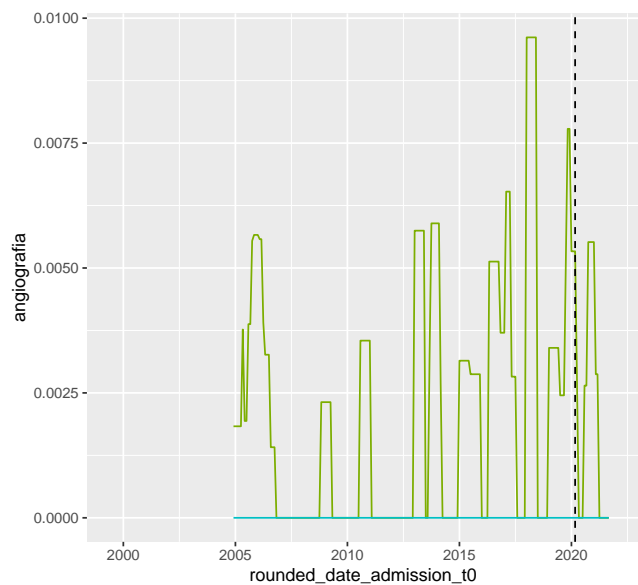


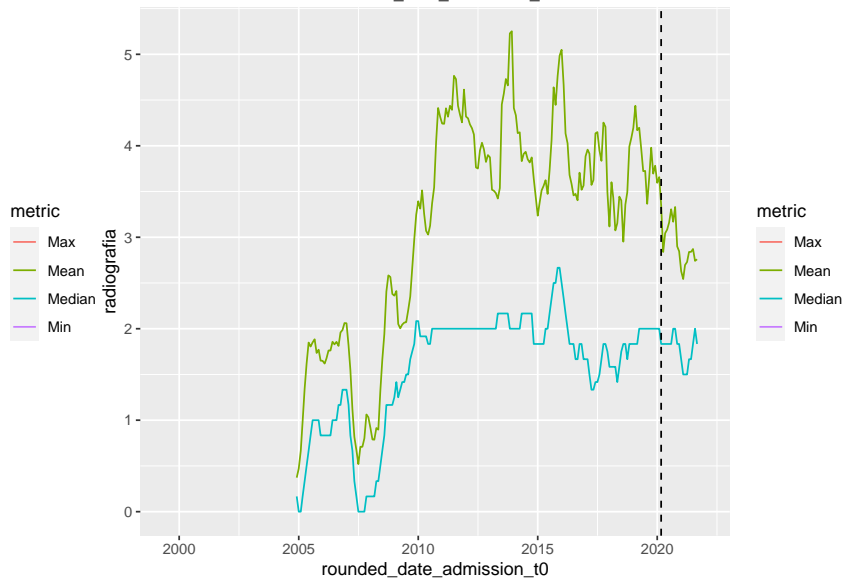
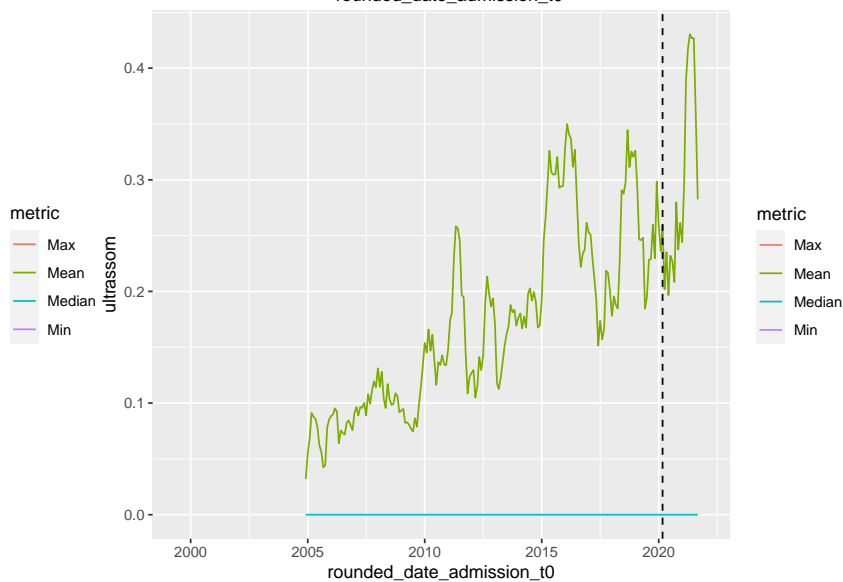
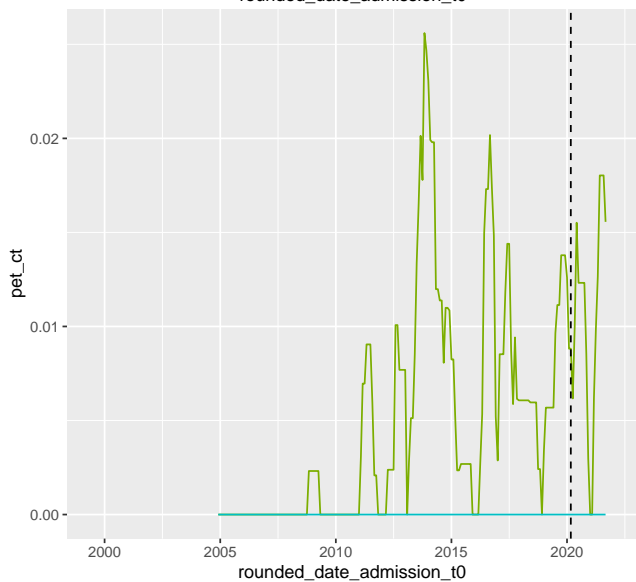
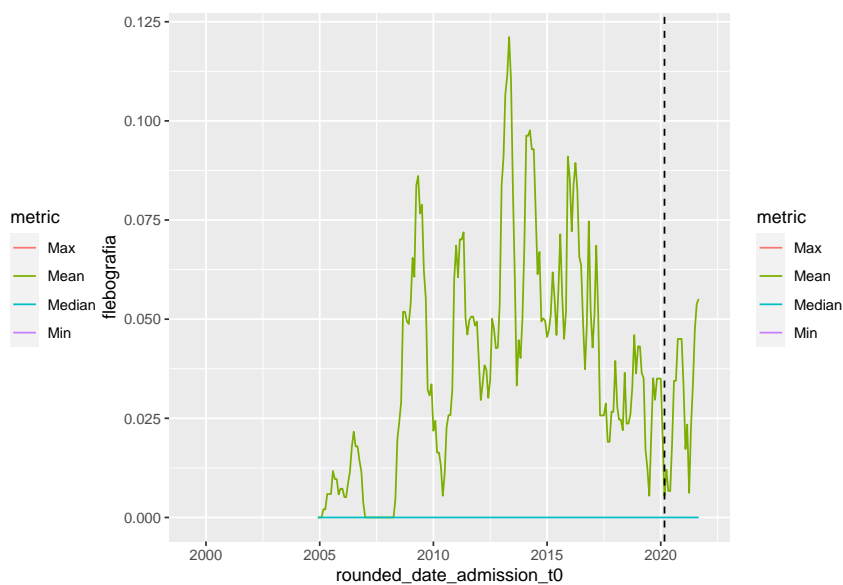
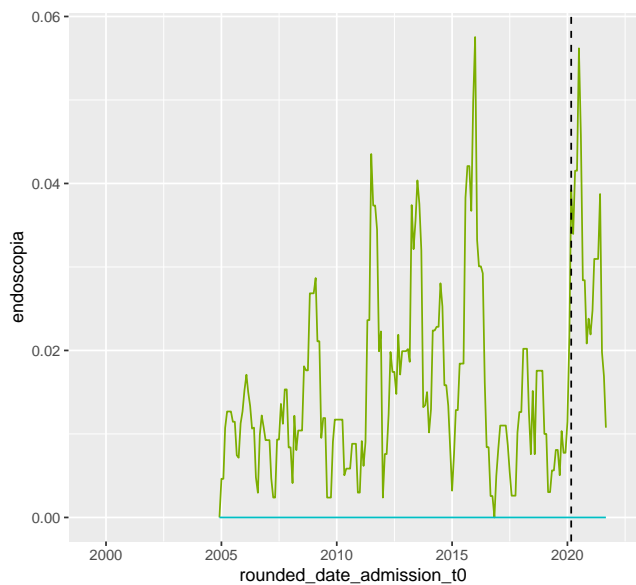


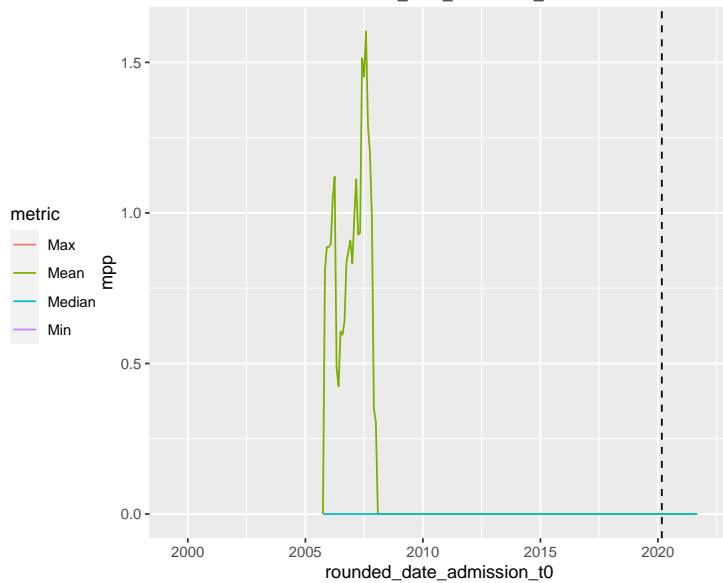
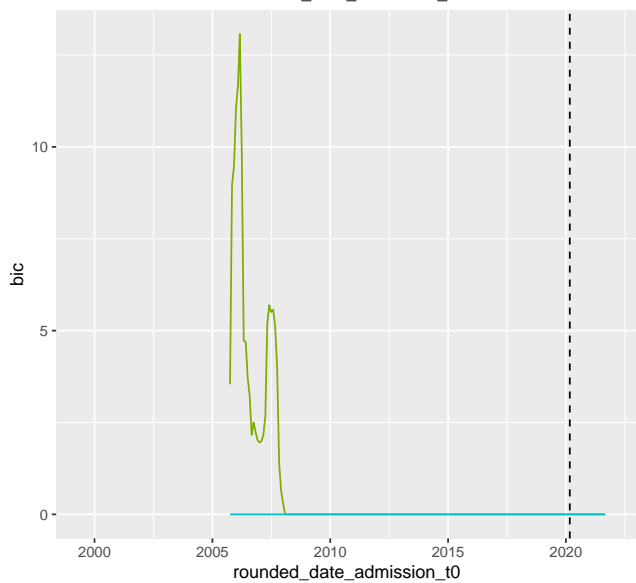
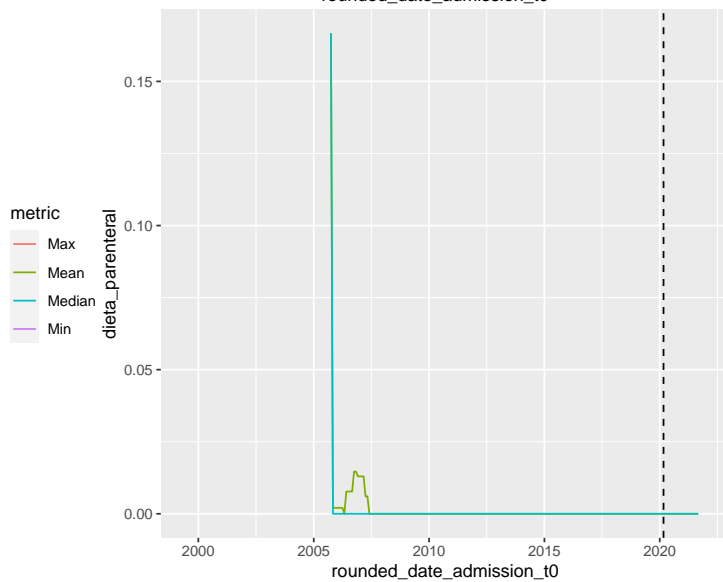
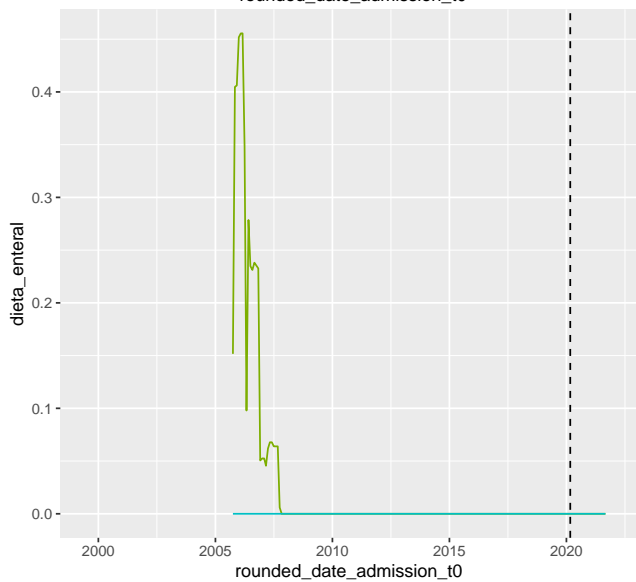
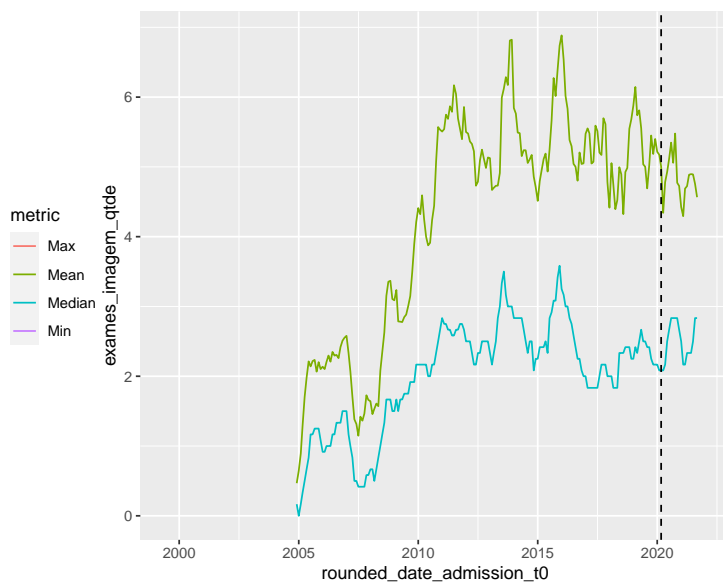
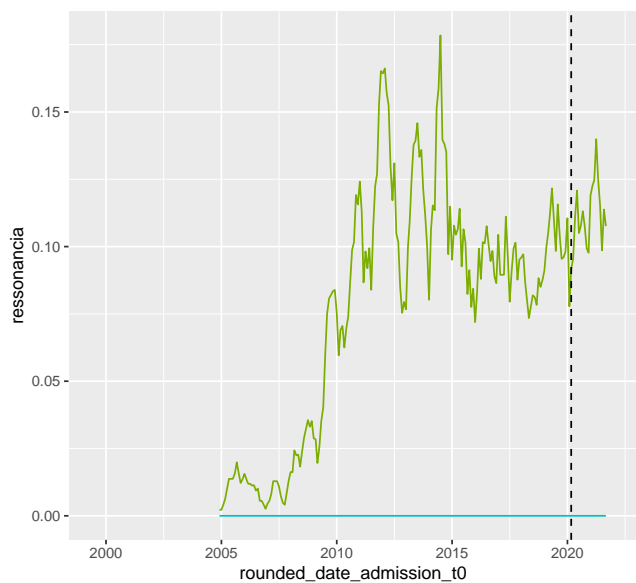


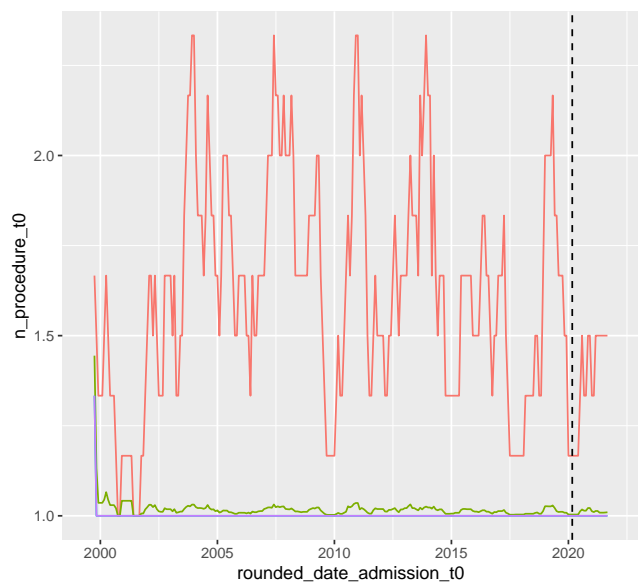






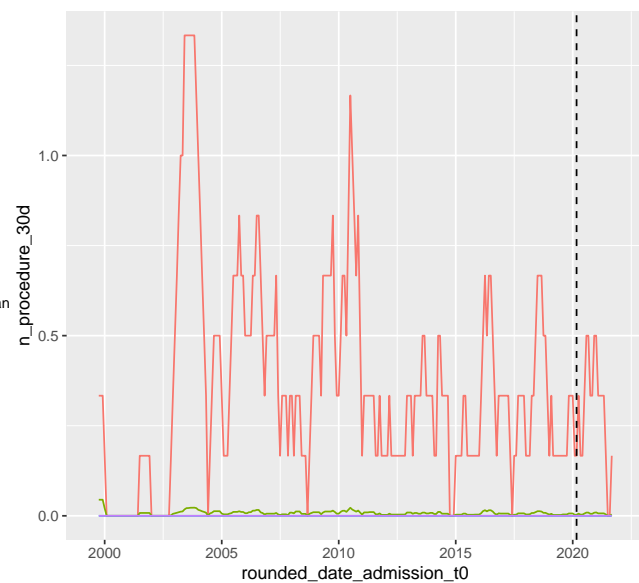






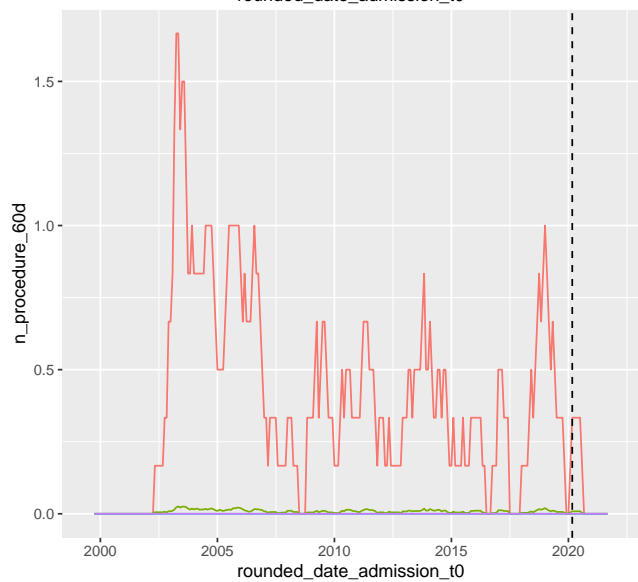
metric

- Max
- Mean
- Median
- Min



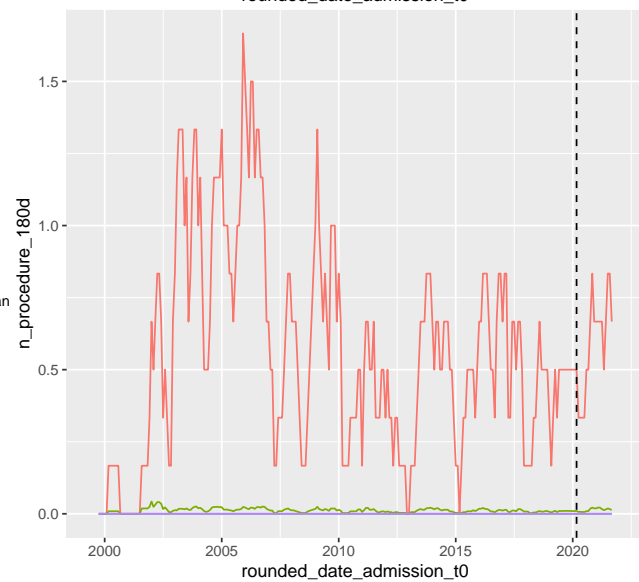
metric

- Max
- Mean
- Median
- Min



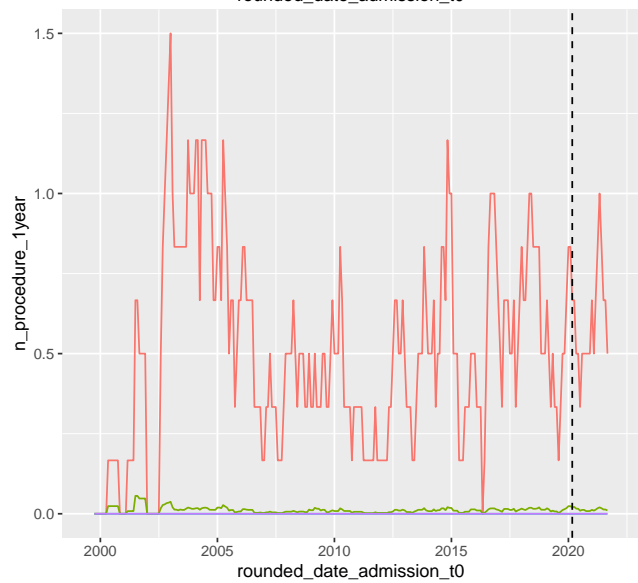
metric

- Max
- Mean
- Median
- Min



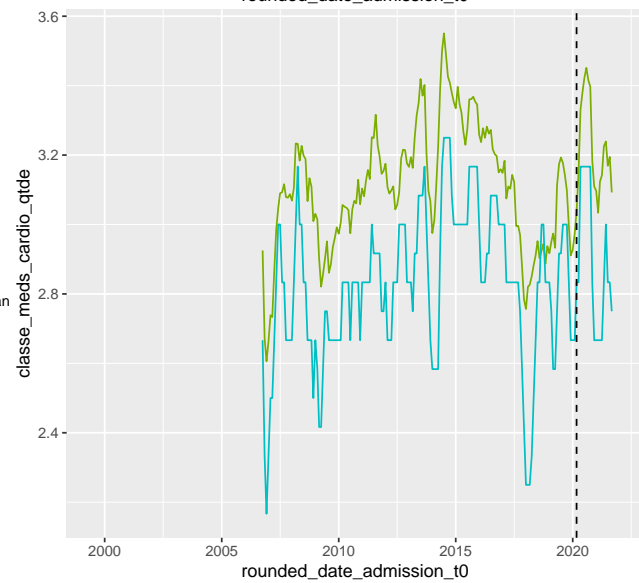
metric

- Max
- Mean
- Median
- Min



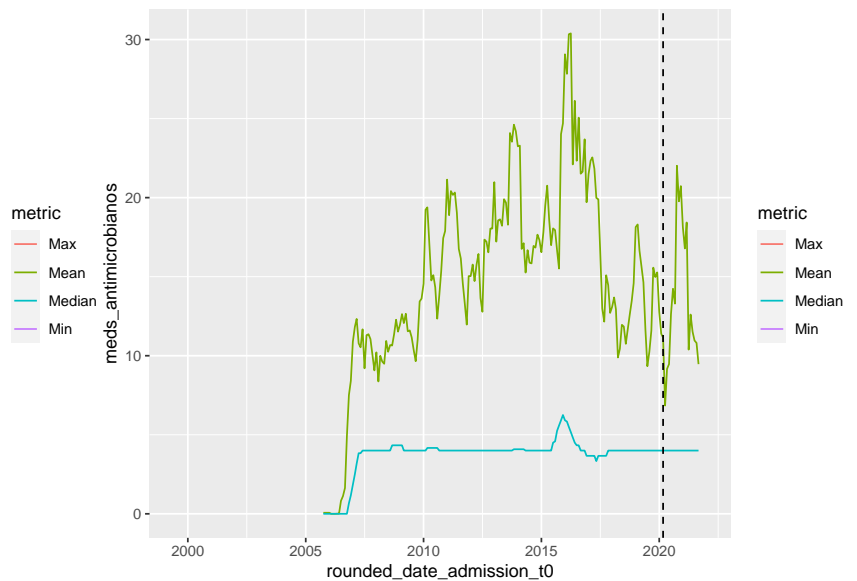
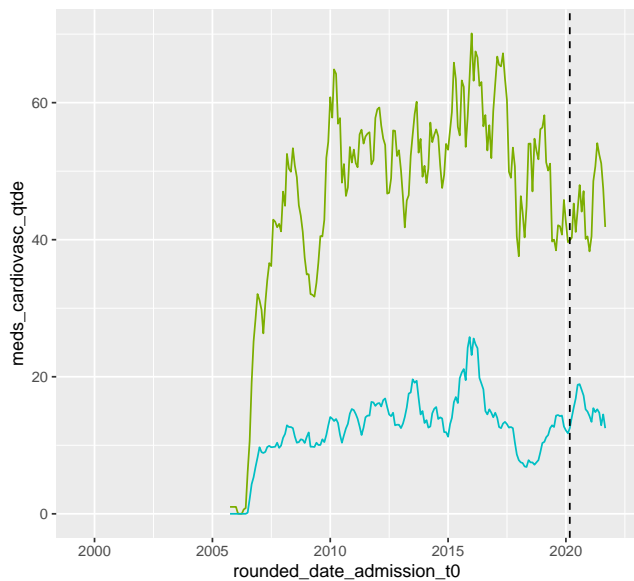
metric

- Max
- Mean
- Median
- Min



metric

- Max
- Mean
- Median
- Min



```
column <- 'death_intraop_2'

df[columns_list$outcome_columns] <- lapply(df[columns_list$outcome_columns], factor)

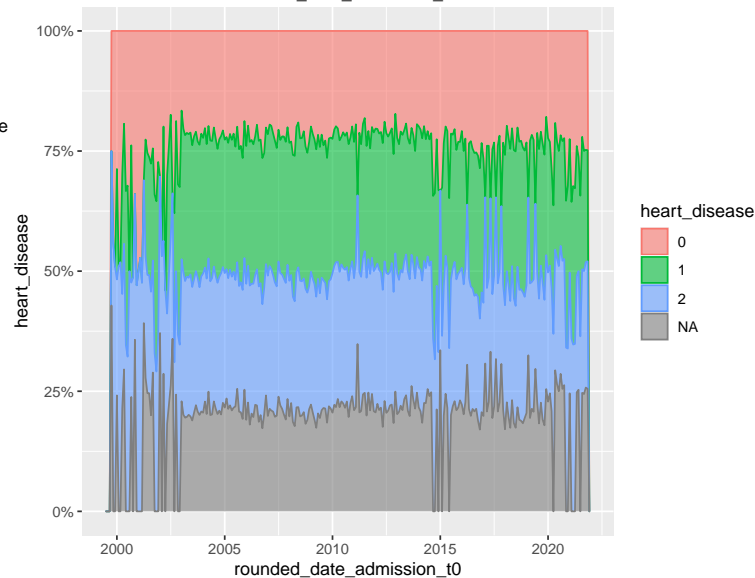
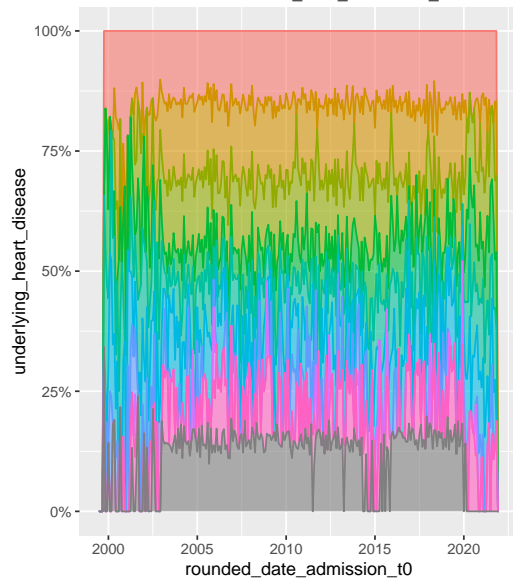
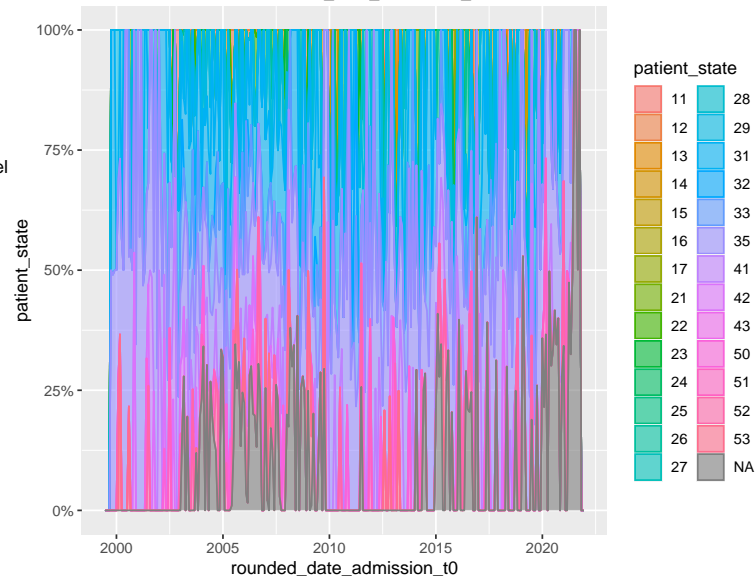
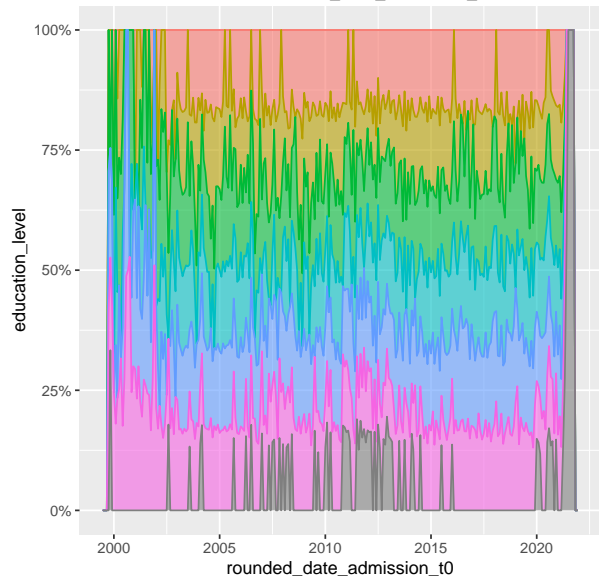
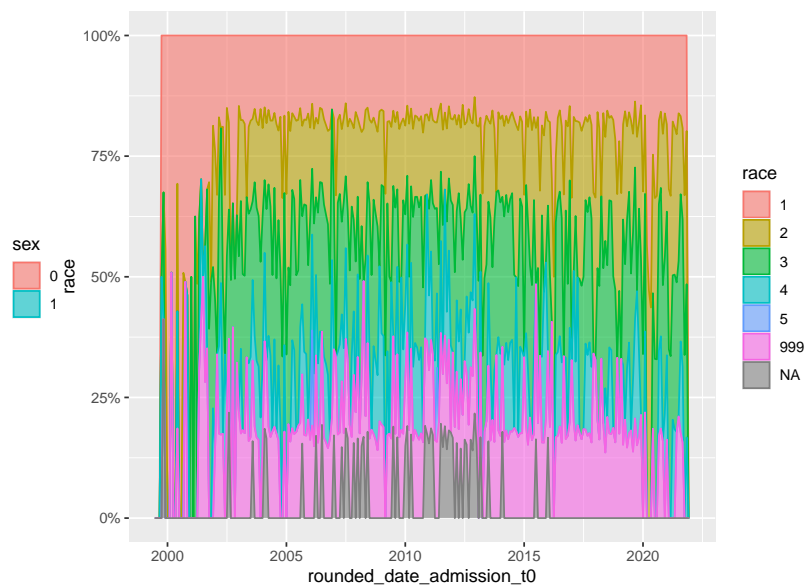
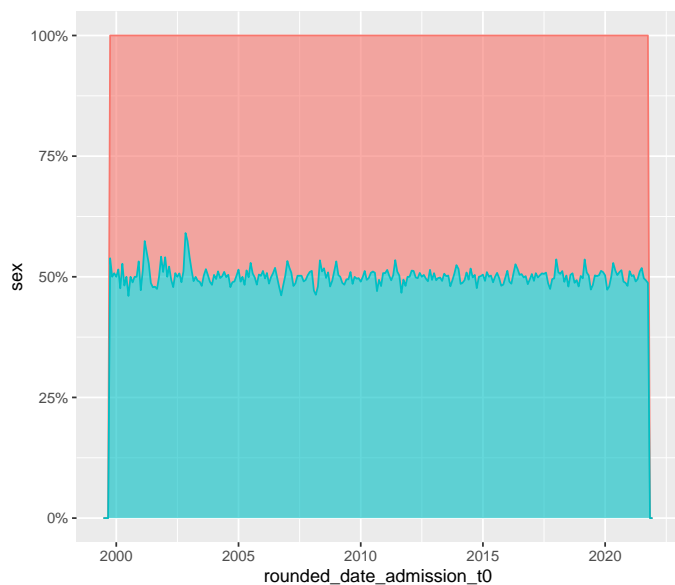
for (column in c(columns_list$categorical_columns, columns_list$outcome_columns)){
  if (mean(is.na(df[[column]])) > 0.5) next
  if (length(unique(df[[column]])) > 10) print(column)

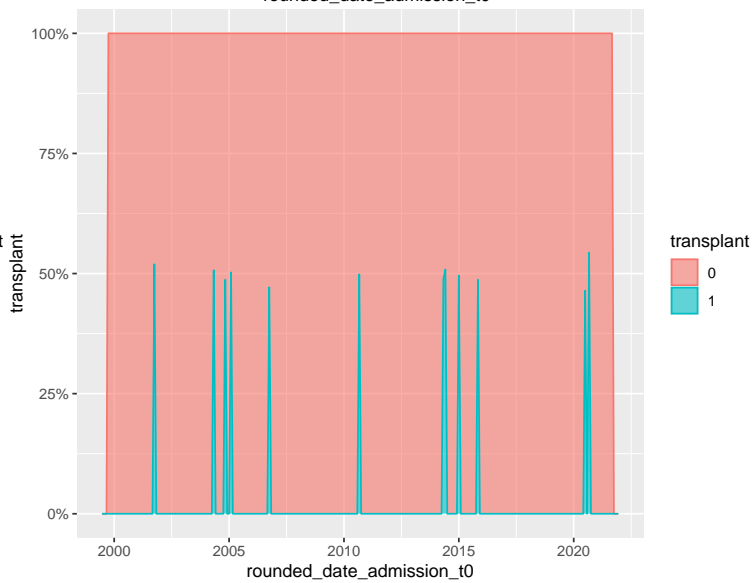
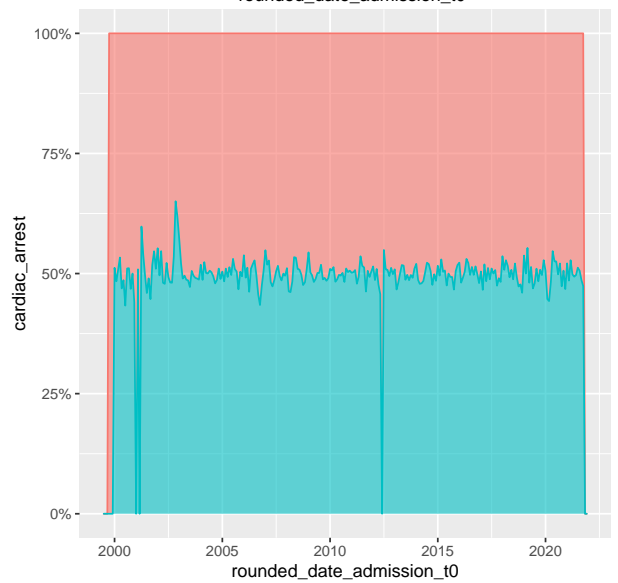
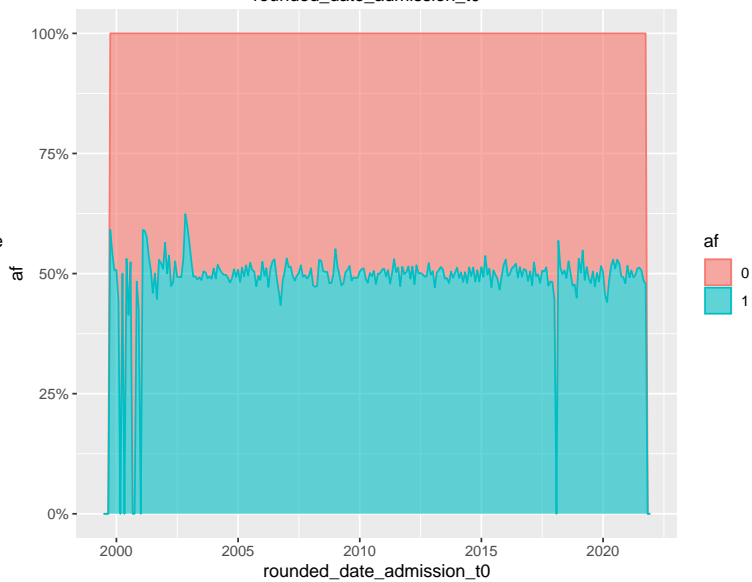
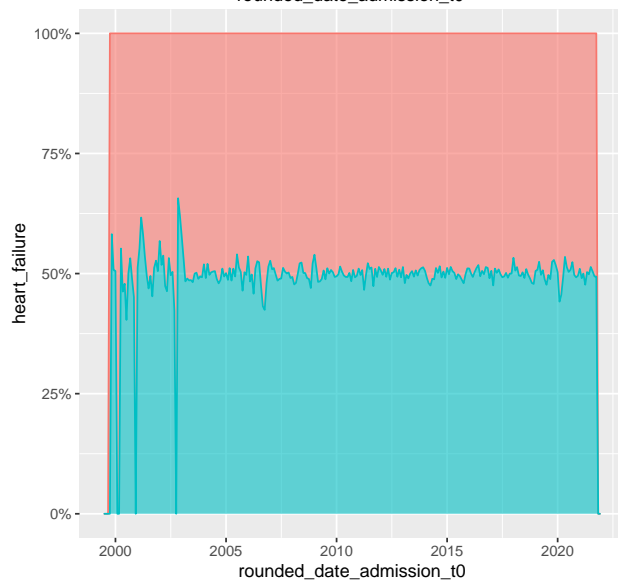
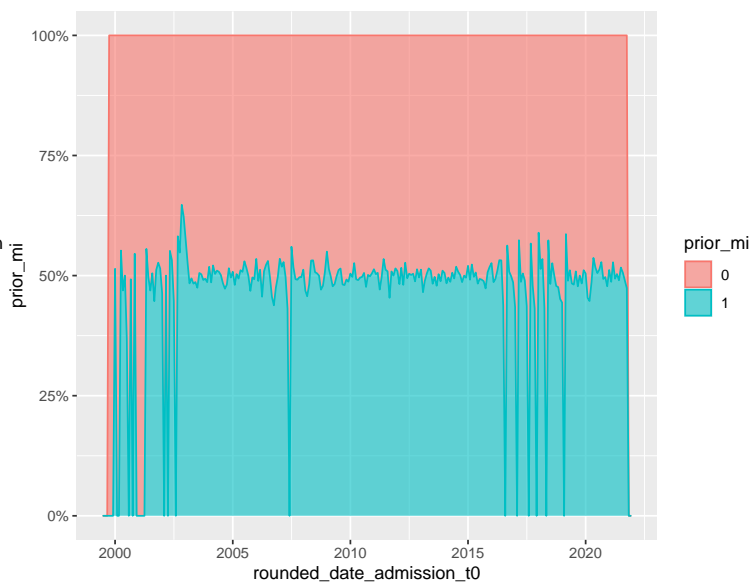
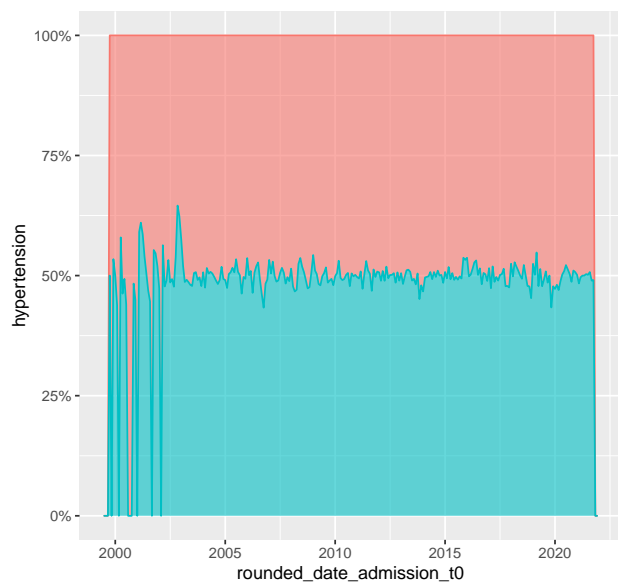
  df %>%
    mutate(rounded_date_admission_t0 = lubridate::floor_date(date_admission_t0,
                                                             'month')) %>%

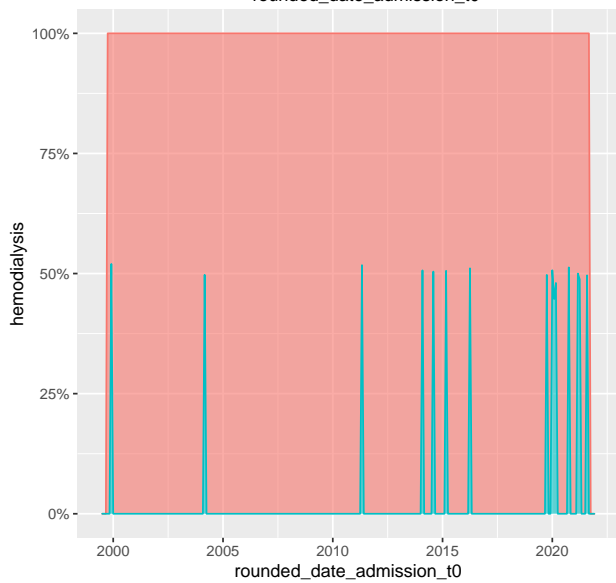
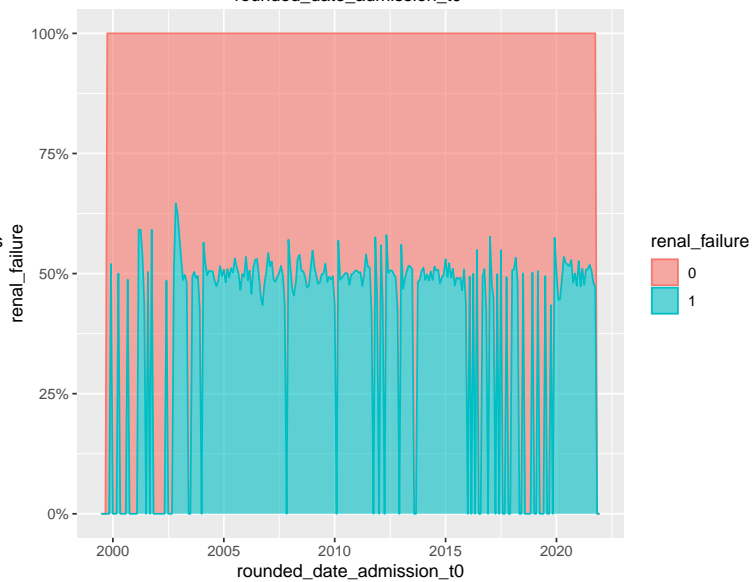
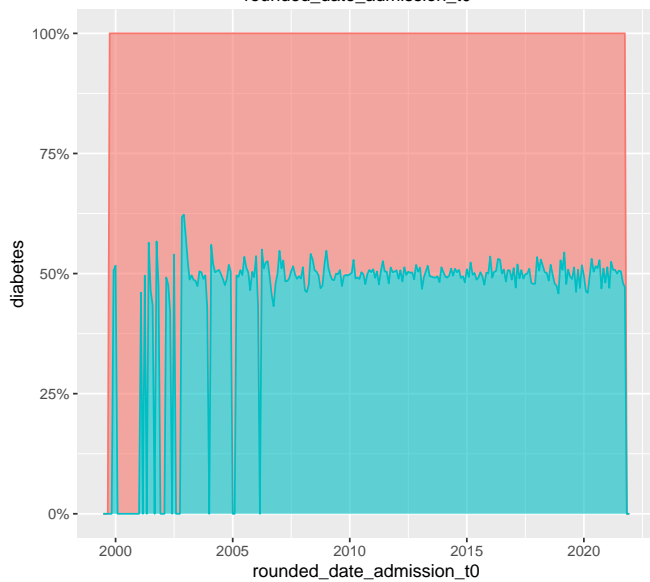
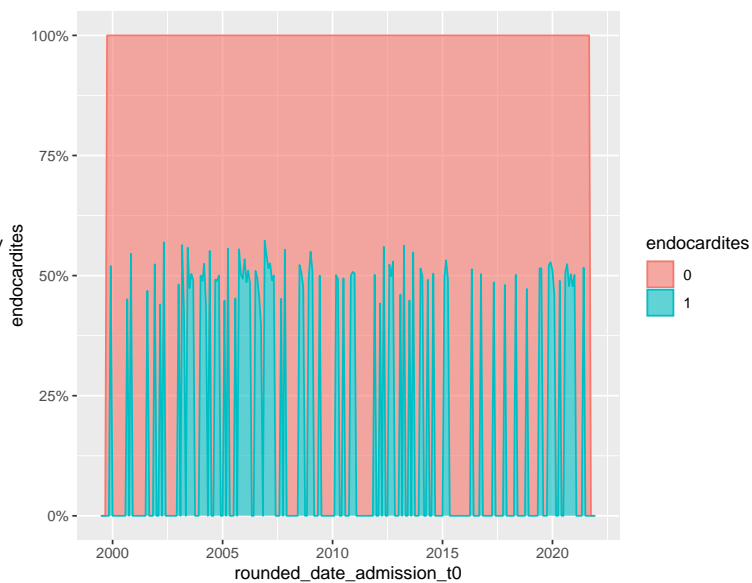
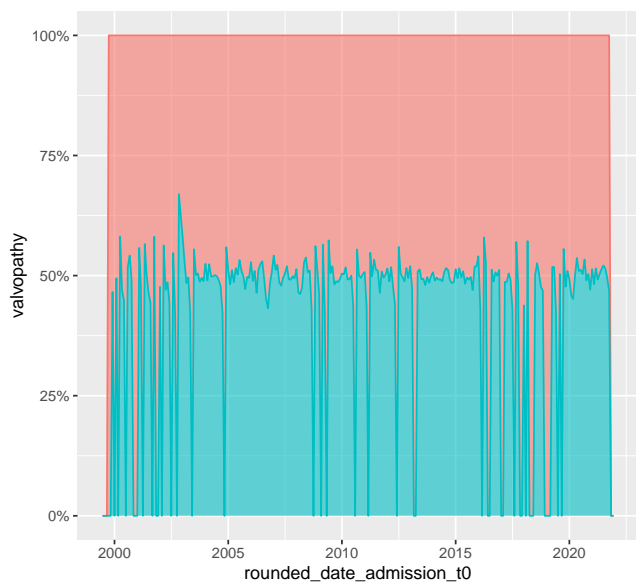
    group_by(rounded_date_admission_t0, !!sym(column)) %>%
    summarise(N = n()) %>%
    ungroup %>%
    arrange(rounded_date_admission_t0) %>%
    mutate(N = zoo::rollmean(N, k = k, fill = NA)) %>%
    group_by(rounded_date_admission_t0) %>%
    mutate(percentage = N / sum(N)) %>%
    ungroup %>%
    tidyr::complete(rounded_date_admission_t0, !!sym(column),
                     fill = list(N = 0, percentage = 0)) %>%
    ggplot(aes(x = rounded_date_admission_t0, y = percentage,
               color = !!sym(column), fill = !!sym(column))) +
    geom_area(alpha=0.6) +
    scale_y_continuous(labels = scales::percent) +
    labs(y = column)-> p

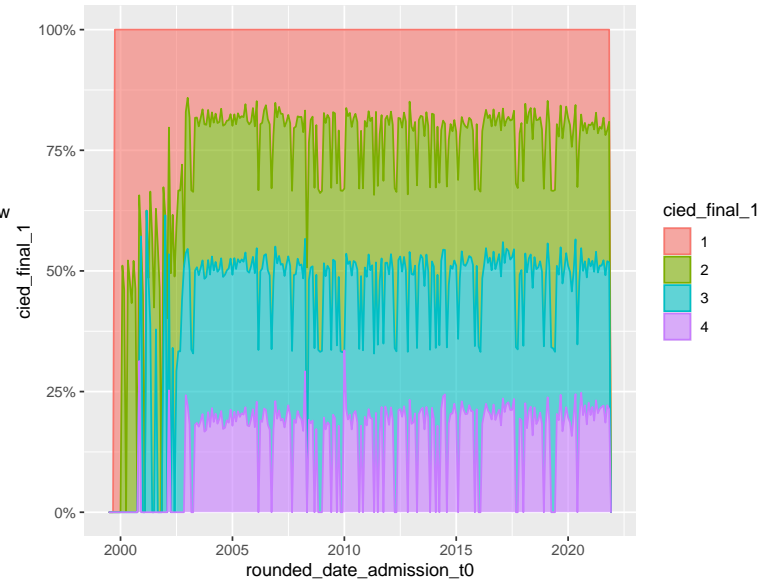
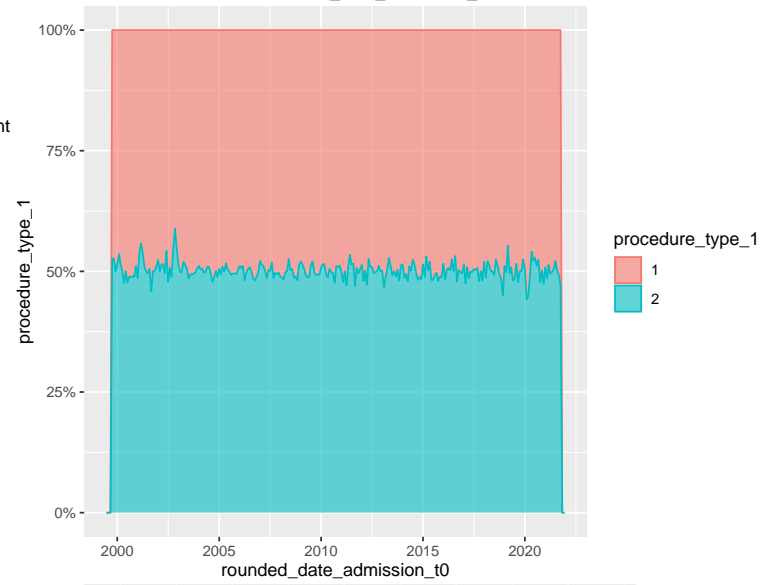
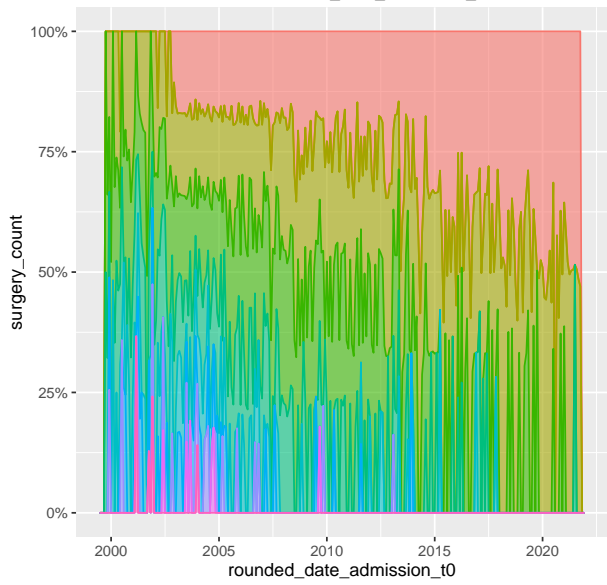
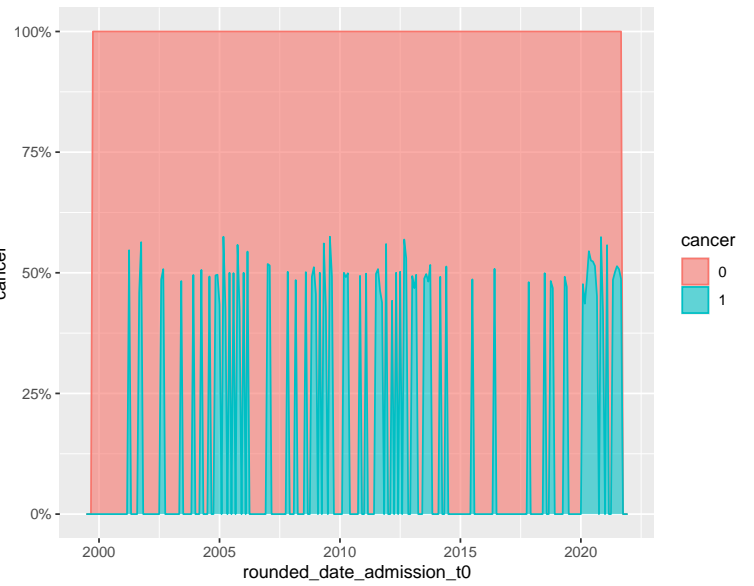
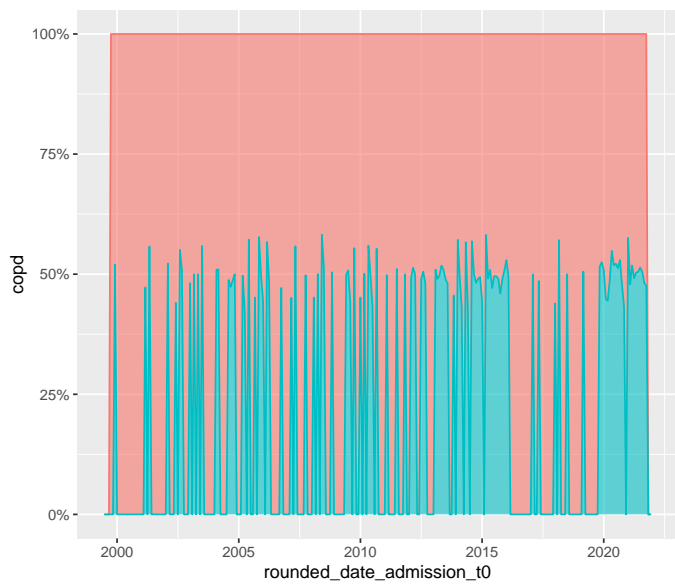
  print(p)
}

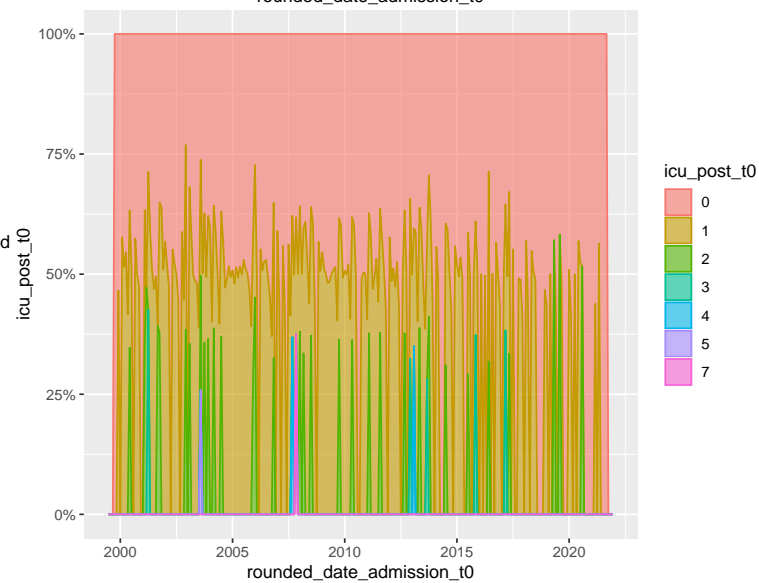
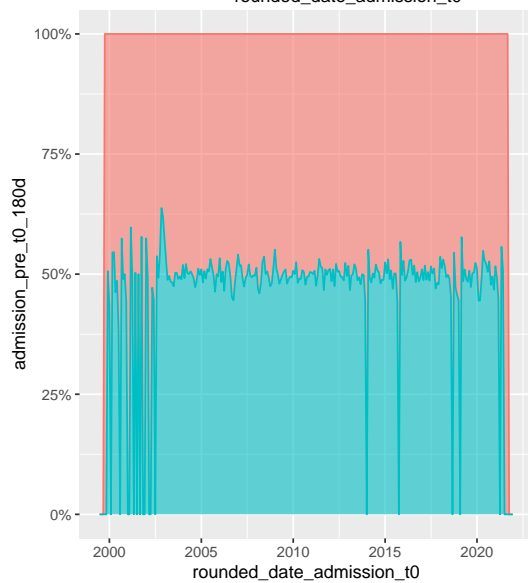
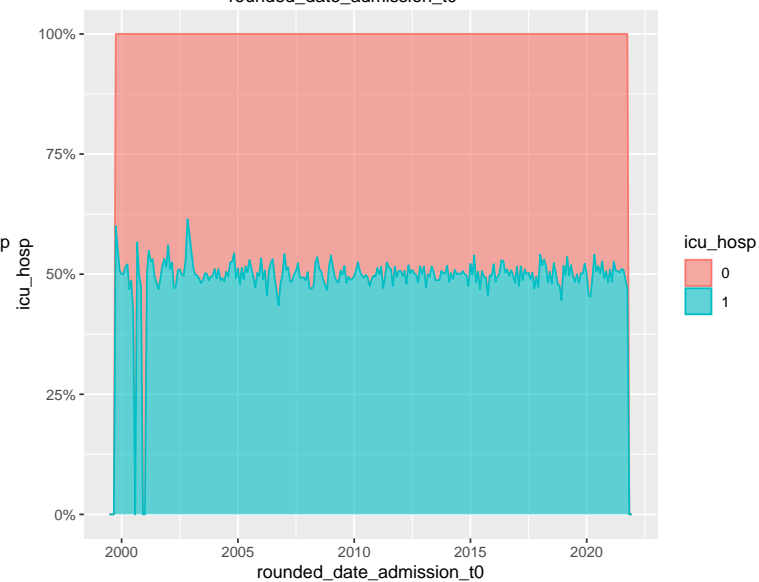
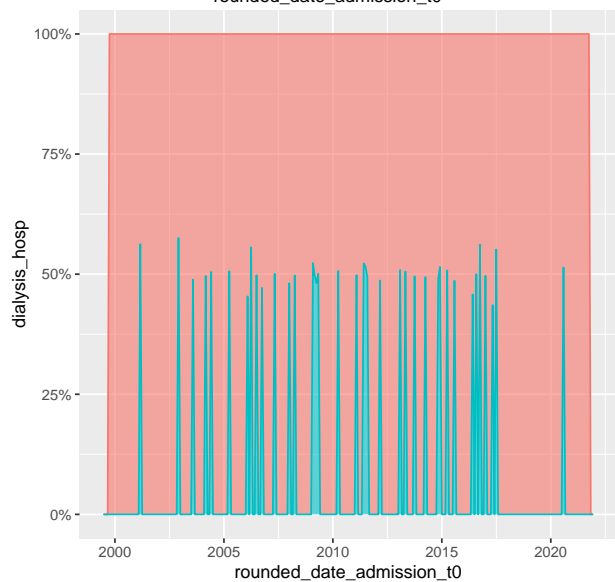
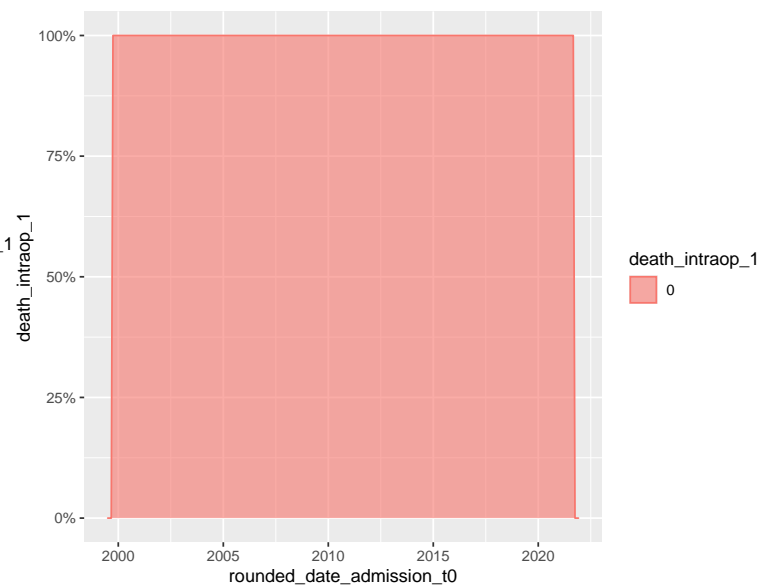
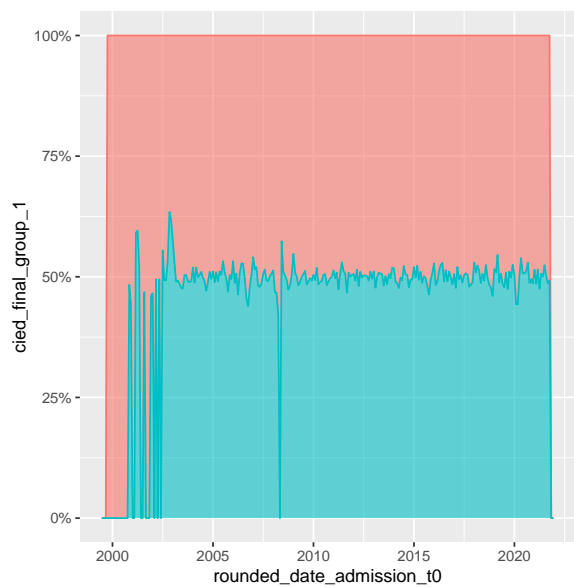
## [1] "patient_state"
```

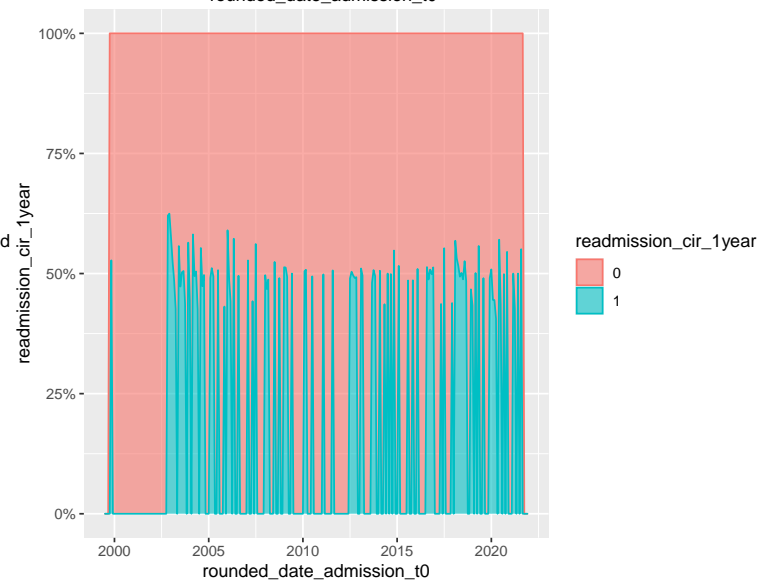
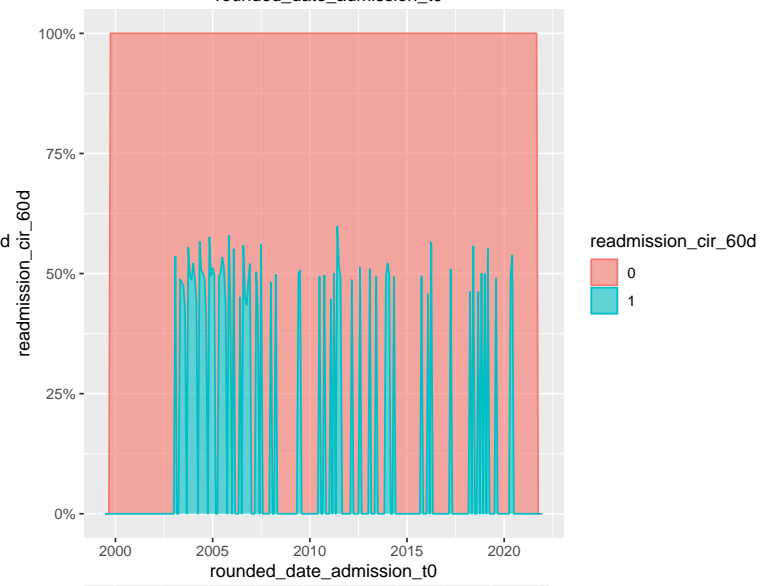
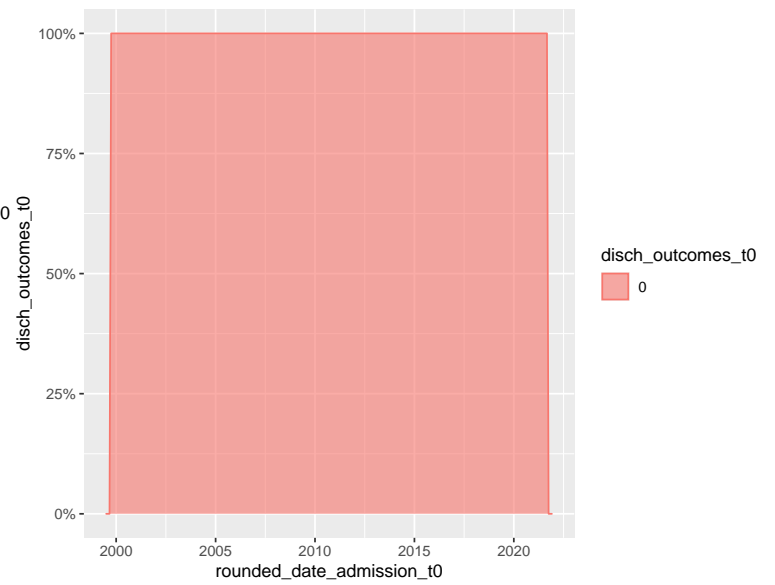
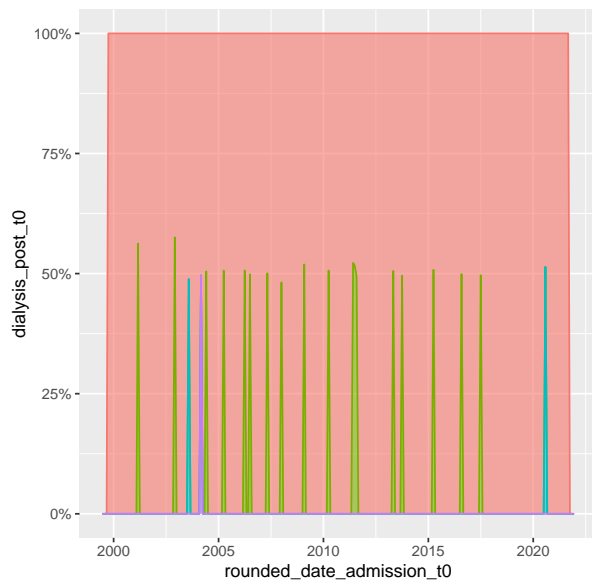


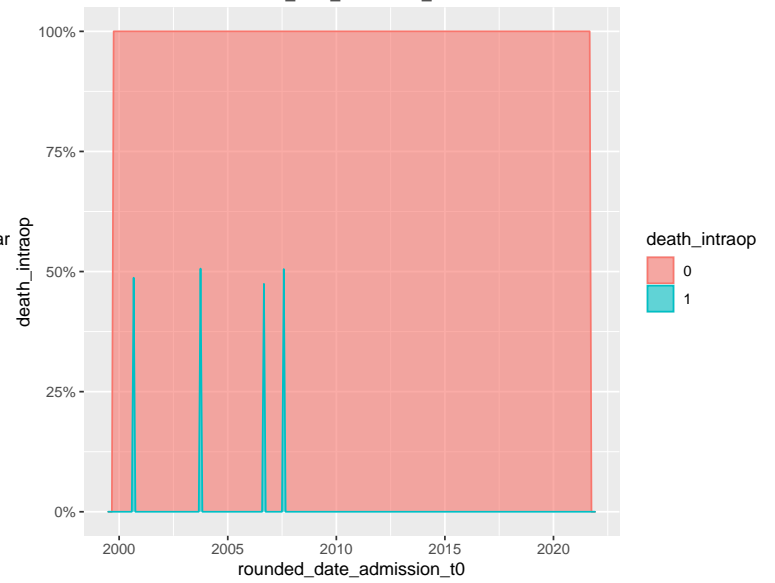
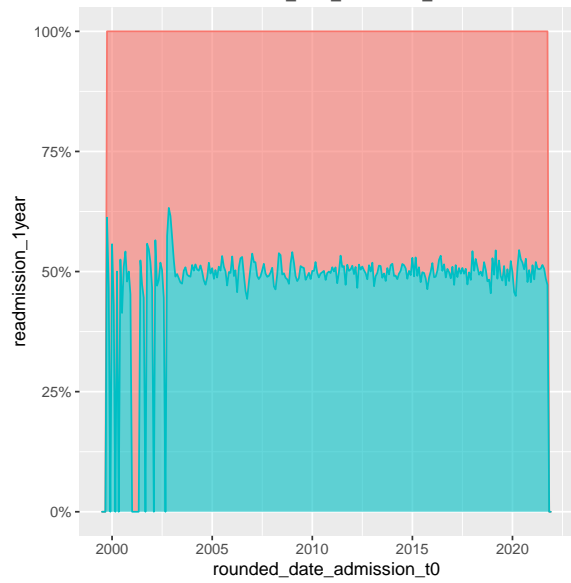
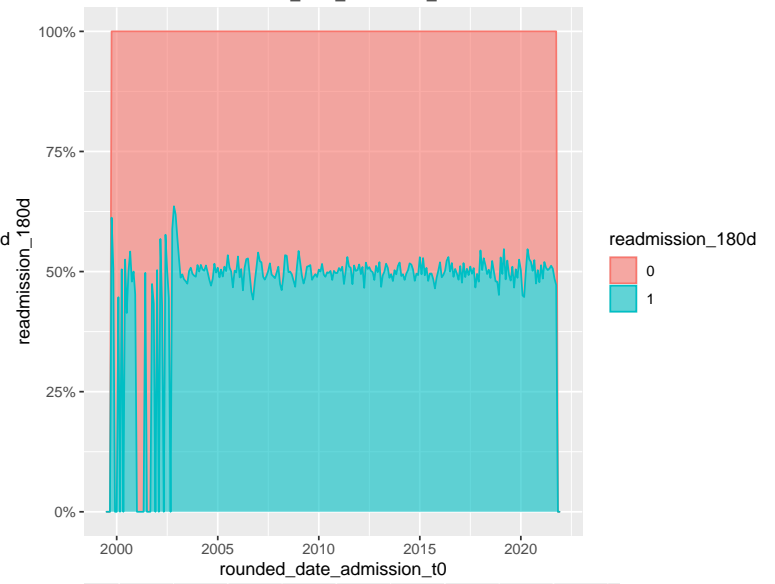
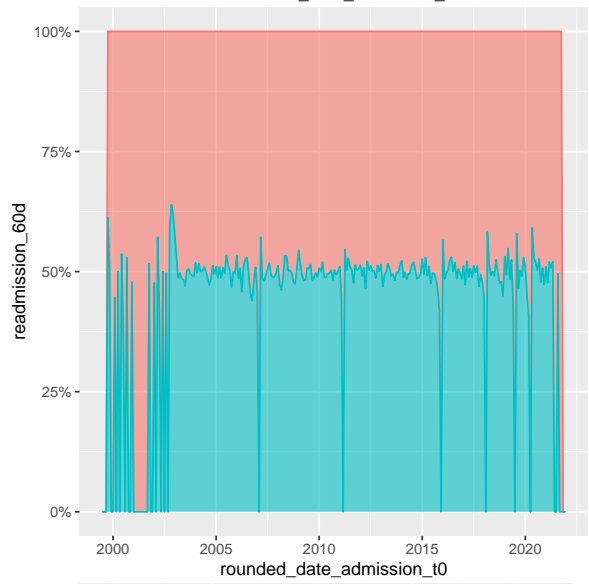
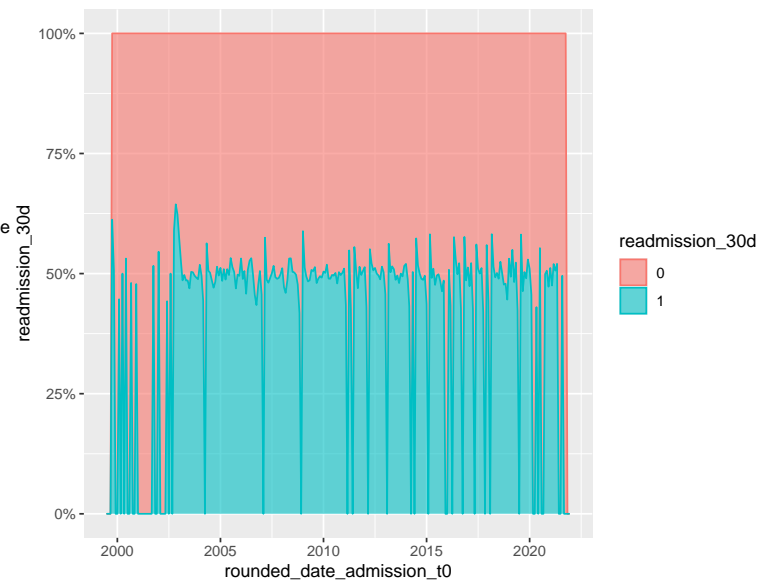
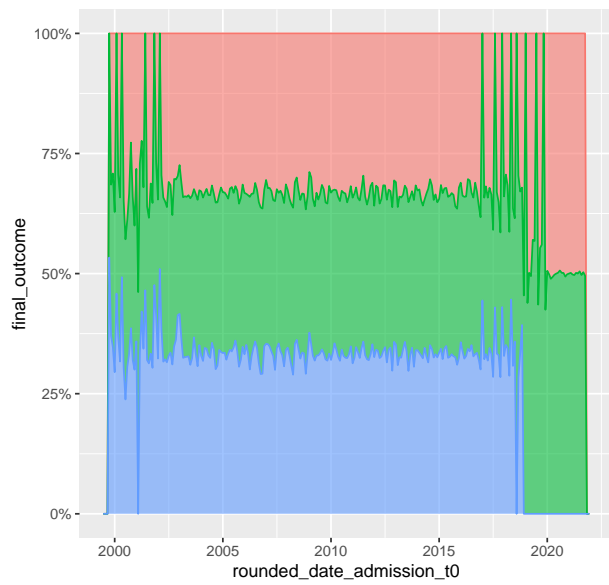


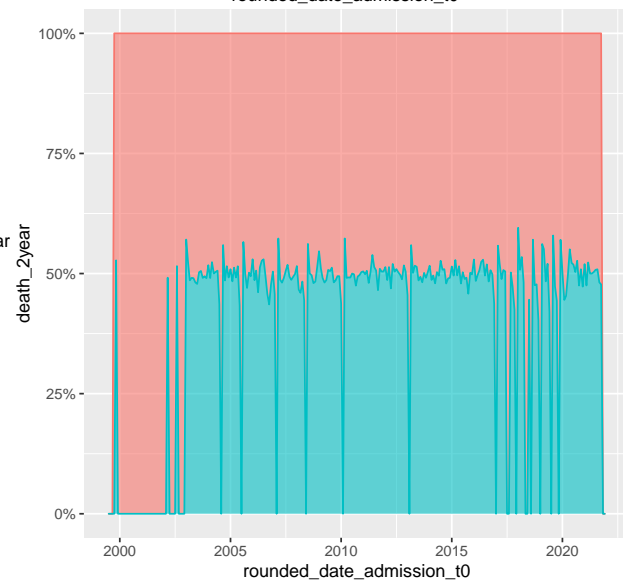
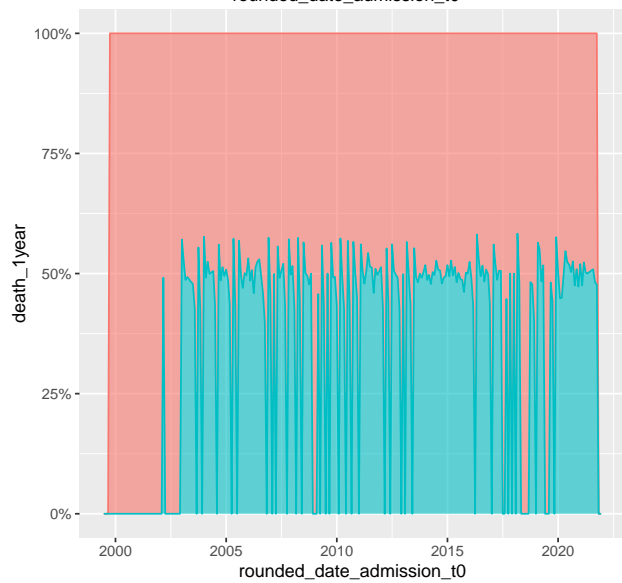
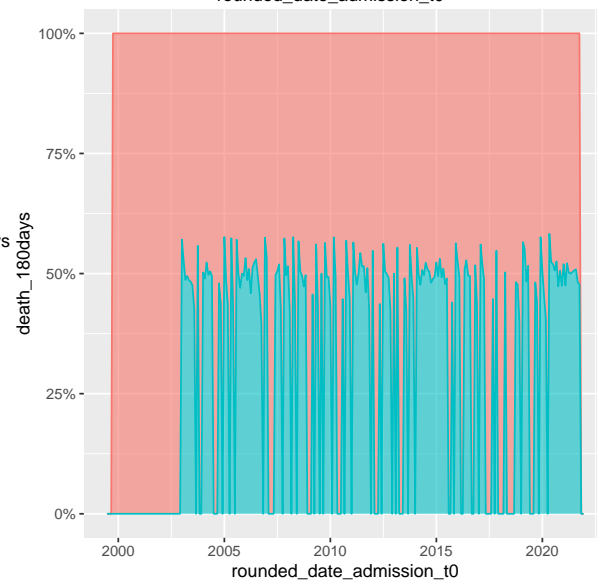
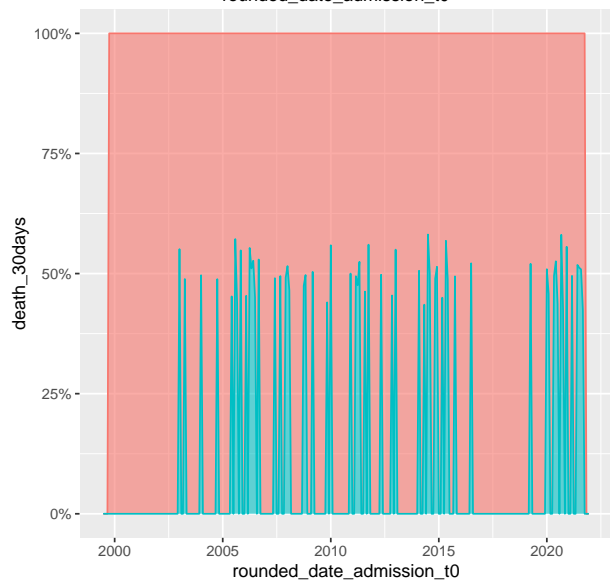
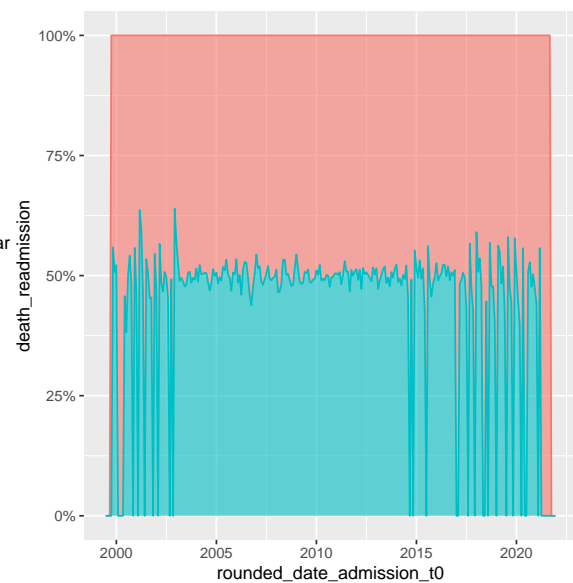
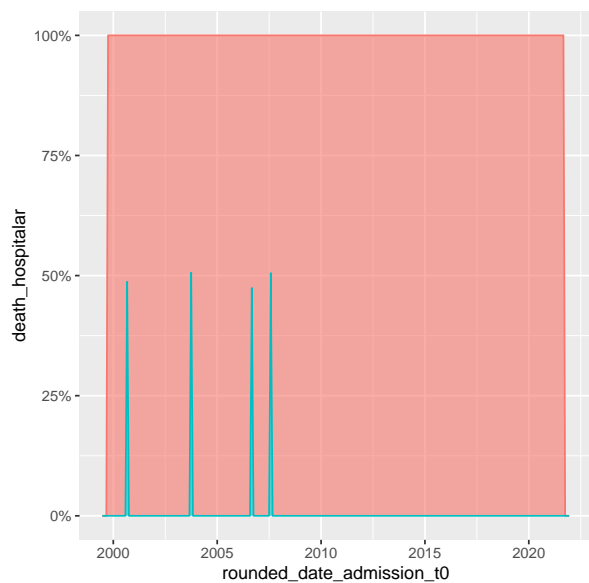


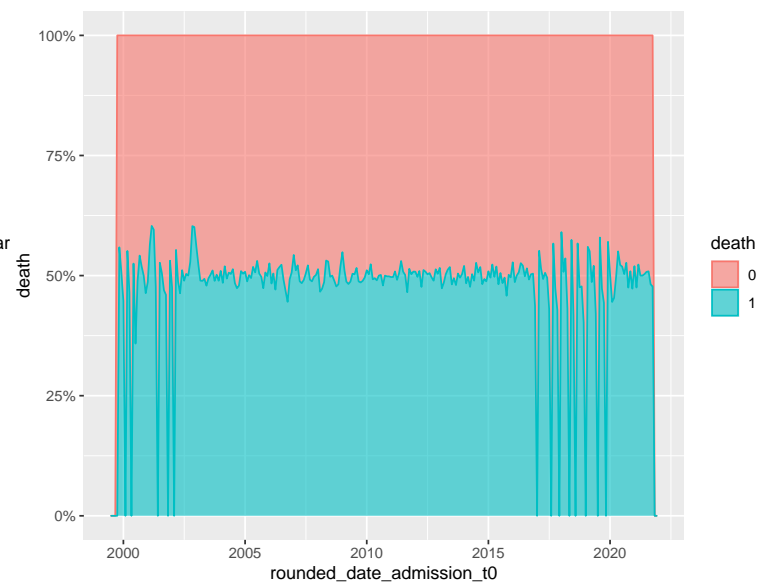
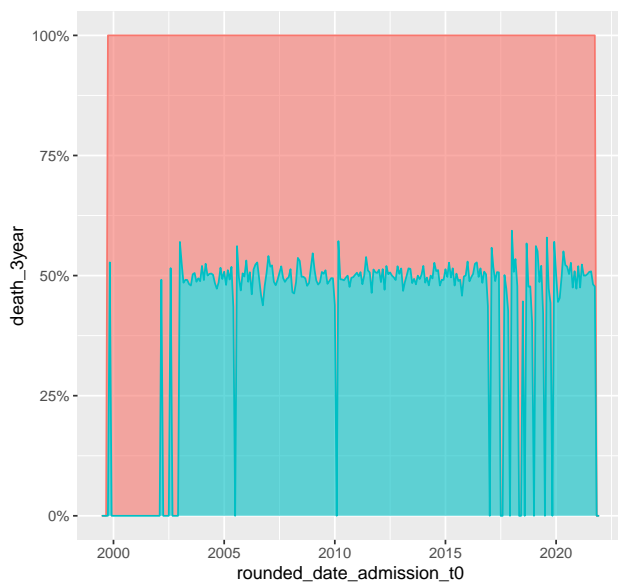












```
df %>%
  mutate(rounded_date_admission_t0 = lubridate::floor_date(date_admission_t0,
                                                         'month')) %>%
  filter(rounded_date_admission_t0 <= lubridate::ymd('2021-08-01'),
         rounded_date_admission_t0 <= lubridate::ymd('2021-01-01')) %>%
  .$aco %>%
  max(na.rm=T)
```

```
## [1] 98
```

```
column <- 'aco'
df %>%
  mutate(rounded_date_admission_t0 = lubridate::floor_date(date_admission_t0,
                                                         'month')) %>%
  group_by(rounded_date_admission_t0) %>%
  summarise('Mean' = mean(!sym(column), na.rm = T),
            'Min' = min(!sym(column), na.rm = T),
            'Median' = medianWithoutNA(!sym(column)),
            'Max' = max(!sym(column), na.rm = T)) %>%
  ungroup %>%
  arrange(rounded_date_admission_t0) %>%
  mutate(Mean = zoo::rollmean(Mean, k = k, fill = NA),
         Min = zoo::rollmean(Min, k = k, fill = NA),
         Median = zoo::rollmean(Median, k = k, fill = NA),
         Max = zoo::rollmean(Max, k = k, fill = NA)) %>%
  filter(rounded_date_admission_t0 <= lubridate::ymd('2021-12-01'),
         rounded_date_admission_t0 >= lubridate::ymd('2021-01-01'))
```

```
## # A tibble: 12 x 5
```

	rounded_date_admission_t0	Mean	Min	Median	Max
	<date>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	2021-01-01	0.441	NaN	0	NaN
## 2	2021-02-01	0.565	NaN	0	NaN
## 3	2021-03-01	0.630	NaN	0	NaN
## 4	2021-04-01	0.738	NaN	0	NaN
## 5	2021-05-01	0.740	NaN	0	NaN
## 6	2021-06-01	0.709	NaN	0	NaN
## 7	2021-07-01	0.569	NaN	0	NaN
## 8	2021-08-01	0.415	NaN	0	NaN
## 9	2021-09-01	0.439	NaN	0	NaN
## 10	2021-10-01	NA	NA	NA	NA
## 11	2021-11-01	NA	NA	NA	NA
## 12	2021-12-01	NA	NA	NA	NA