

Creating lagged RT-LBCI values

Lagged by 1, 2, 3 and 4 weeks

Vimaljeet Singh

2023-05-15

Read the RT-LBCI file

Extract year and month from the date and add it as a separate column because we will use them to group values together.

```
index <- read.csv("data/RT-LBCI.csv")

# Converting to date type
index$Date <- as.Date(index$Date)

# Extracting year and month
index$Year <- year(index$Date)
index$Month <- month(index$Date)

head(index)
```

##	Date	Montreal	Ottawa	Toronto	Calgary	Vancouver	Year	Month
## 1	2020-08-10	100.00	100.00	100.00	100.00	100.00	2020	8
## 2	2020-08-17	89.20	97.79	104.41	102.18	103.20	2020	8
## 3	2020-08-24	87.22	102.30	106.39	114.89	101.99	2020	8
## 4	2020-08-31	98.07	99.30	109.65	109.88	101.32	2020	8
## 5	2020-09-07	102.95	107.32	112.93	119.03	109.63	2020	9
## 6	2020-09-14	97.83	103.15	109.35	120.31	108.88	2020	9

Create required functions

- Function 1 creates lag in the input df by n intervals (in our case n = number of weeks we want to lag by)
- Function 2 works on the output of the first function and aggregates the lagged values by calculating the mean value by grouping them by year then month in 'cities' and rounds the mean values to 2 decimal places.
- Function 3 creates a date column by clubbing the month and year columns and making the date to be the first day of that month and drops the year and month column in the input df (which is the output from function 2)

```

# Function 1
create_lag <- function(data, n) {
  lagged_df <- data %>%
    arrange(Year, Month) %>%
    mutate(
      across(-Date, ~lag(., n))
    )

  lagged_df <- na.omit(lagged_df)
  row.names(lagged_df) <- NULL

  lagged_df$Year <- year(lagged_df$Date)
  lagged_df$Month <- month(lagged_df$Date)

  return(lagged_df)
}

# Function 2
aggregate_lagged_values <- function(data) {
  cities = c("Montreal", "Ottawa", "Toronto", "Calgary", "Vancouver")
  result <- data %>%
    group_by(Year, Month) %>%
    summarize(across(all_of(cities), ~ round(mean(., na.rm = TRUE), 2)))

  result <- as.data.frame(result) # Convert to data frame

  return(result)
}

# Function 3
add_date <- function(data) {
  # Convert Year and Month columns to numeric if they are stored as characters
  data$Year <- as.numeric(data$Year)
  data$Month <- as.numeric(data$Month)

  # Create a new Date column using the Year and Month values
  data$Date <- as.Date(paste(data$Year, data$Month, "01", sep = "-"))

  # Move the Date column to the first position
  data <- data[, c("Date", names(data)[-which(names(data) %in% c("Year", "Month"))])]

  # Remove the duplicate Date column
  data <- data[, -which(names(data) == "Date.1")]

  return(data)
}

```

Creating lagged variable aggregated by year and month and it has first date of each month

```
week1 <- create_lag(index, 1)
week2 <- create_lag(index, 2)
week3 <- create_lag(index, 3)
week4 <- create_lag(index, 4)

# Lagged_1week <- aggregate_lagged_values(week1)
# Lagged_2week <- aggregate_lagged_values(week2)
# Lagged_3week <- aggregate_lagged_values(week3)
# Lagged_4week <- aggregate_lagged_values(week4)

Lagged_1week <- add_date(aggregate_lagged_values(week1))
Lagged_2week <- add_date(aggregate_lagged_values(week2))
Lagged_3week <- add_date(aggregate_lagged_values(week3))
Lagged_4week <- add_date(aggregate_lagged_values(week4))
```

```
head(Lagged_1week)
```

##	Date	Montreal	Ottawa	Toronto	Calgary	Vancouver
## 1	2020-08-01	92.14	100.03	103.60	105.69	101.73
## 2	2020-09-01	100.07	101.30	109.30	113.60	105.27
## 3	2020-10-01	121.11	114.16	126.95	135.67	126.14
## 4	2020-11-01	135.90	128.19	127.17	142.12	138.22
## 5	2020-12-01	152.39	139.92	129.40	145.04	145.85
## 6	2021-01-01	164.84	129.79	141.80	141.40	179.55

```
head(Lagged_2week)
```

##	Date	Montreal	Ottawa	Toronto	Calgary	Vancouver
## 1	2020-08-01	94.60	98.90	102.20	101.09	101.60
## 2	2020-09-01	96.52	103.02	109.58	116.03	105.46
## 3	2020-10-01	117.49	108.66	123.48	128.20	117.68
## 4	2020-11-01	131.40	125.55	125.46	140.55	137.90
## 5	2020-12-01	147.39	135.41	127.56	145.75	140.84
## 6	2021-01-01	163.33	141.08	143.16	141.80	177.49

```
head(Lagged_3week)
```

##	Date	Montreal	Ottawa	Toronto	Calgary	Vancouver
## 1	2020-08-01	100.00	100.00	100.00	100.00	100.00
## 2	2020-09-01	94.36	101.68	108.34	111.50	104.03
## 3	2020-10-01	112.00	106.10	113.62	123.23	113.22
## 4	2020-11-01	125.21	117.95	127.10	138.91	135.26
## 5	2020-12-01	148.89	138.40	131.72	147.17	140.50
## 6	2021-01-01	157.77	145.31	138.18	142.01	167.14

```
head(Lagged_4week)
```

```
##           Date Montreal Ottawa Toronto Calgary Vancouver
## 1 2020-09-01    93.62  99.85  105.11  106.74    101.63
## 2 2020-10-01   107.23 103.39  111.25  117.86    108.22
## 3 2020-11-01   122.11 115.34  125.92  137.18    132.35
## 4 2020-12-01   147.69 137.22  131.47  148.71    140.83
## 5 2021-01-01   155.14 143.09  134.18  139.74    156.87
## 6 2021-02-01   163.44 119.11  139.36  143.25    173.70
```

```
# Change the file path if you want to save the dataframes as csv in your machine
# write.csv(Lagged_1week, "data/Lagged_1week.csv", row.names = FALSE)
# write.csv(Lagged_2week, "data/Lagged_2week.csv", row.names = FALSE)
# write.csv(Lagged_3week, "data/Lagged_3week.csv", row.names = FALSE)
# write.csv(Lagged_4week, "data/Lagged_4week.csv", row.names = FALSE)
```