COVID 19 Analysis

24 April 2023

Required Packages

Part 1 - Basic Exploration of US Data The New York Times (the Times) has aggregated reported COVID-19 data from state and local governments and health departments since 2020 and provides public access through a repository on GitHub. One of the data sets provided by the Times is county-level data for cumulative cases and deaths each day. This will be your primary data set for the first two parts of your analysis.

County-level COVID data from 2020, 2021, and 2022 has been imported below. Each row of data reports the cumulative number of cases and deaths for a specific county each day. A FIPS code, a standard geographic identifier, is also provided which you will use in Part 2 to construct a map visualization at the county level for a state.

Additionally, county-level population estimates reported by the US Census Bureau has been imported as well. You will use these estimates to caluclate statistics per 100,000 people.

```
# Import New York Times COVID-19 data
# Import Population Estimates from US Census Bureau

us_counties_2020 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
us_counties_2021 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
us_counties_2022 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
```

```
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
us_population_estimates <- read_csv("fips_population_estimates.csv")
```

```
## Parsed with column specification:
## cols(
##   STNAME = col_character(),
##   CTYNAME = col_character(),
##   fips = col_double(),
##   STATE = col_double(),
##   COUNTY = col_double(),
##   Year = col_double(),
##   Estimate = col_double()
## )
```

Question 1 Your first task is to combine and tidy the 2020, 2021, and 2022 COVID data sets and find the total deaths and cases for each day since March 15, 2020 (2020-03-15). The data sets provided from the NY Times also includes statistics from Puerto Rico, a US territory. You may remove these observations from the data as they will not be needed for your analysis. Once you have tidied the data, find the total COVID-19 cases and deaths since March 15, 2020. Write a sentence or two after the code block communicating your results. Use inline code to include the `max_date`, `us_total_cases`, and `us_total_deaths` variables. To write inline code use `r`.

```
# Combine and tidy the 2020, 2021, and 2022 COVID data sets.
# Hint: Review the rbind() documentation to combine the three data sets.
#
## YOUR CODE HERE ##
us_counties_20to22 <- bind_rows(us_counties_2020, us_counties_2021, us_counties_2022) %>%
  filter(!state %in% "Puerto Rico") %>%  #filter out Puerto Rico
  filter(!date < "2020-03-15") #fliter dates before March 15
#Data is otherwise tidy.

totals <- us_counties_20to22 %>%
  group_by(date) %>%
  summarise(`total deaths` = sum(deaths),
            `total cases` = sum(cases))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
max_date <- max(totals$date)
us_total_cases <- max(totals$`total cases`)
us_total_deaths <- max(totals$`total deaths`)
```

```
# Your output should look similar to the following tibble:
#
#   A tibble: 657 x 3
#       date          total_deaths   total_cases
#      <date>            <dbl>         <dbl>
#   1 2020-03-15           68          3595
#   2 2020-03-16           91          4502
#   3 2020-03-17          117          5901
```

```
#   4 2020-03-18          162          8345
#   5 2020-03-19          212         12387
#   6 2020-03-20          277         17998
#   7 2020-03-21          359         24507
#   8 2020-03-22          457         33050
#   9 2020-03-23          577         43474
#  10 2020-03-24          783         53899
# ... with 647 more rows
#
```

– Communicate your methodology, results, and interpretation here –

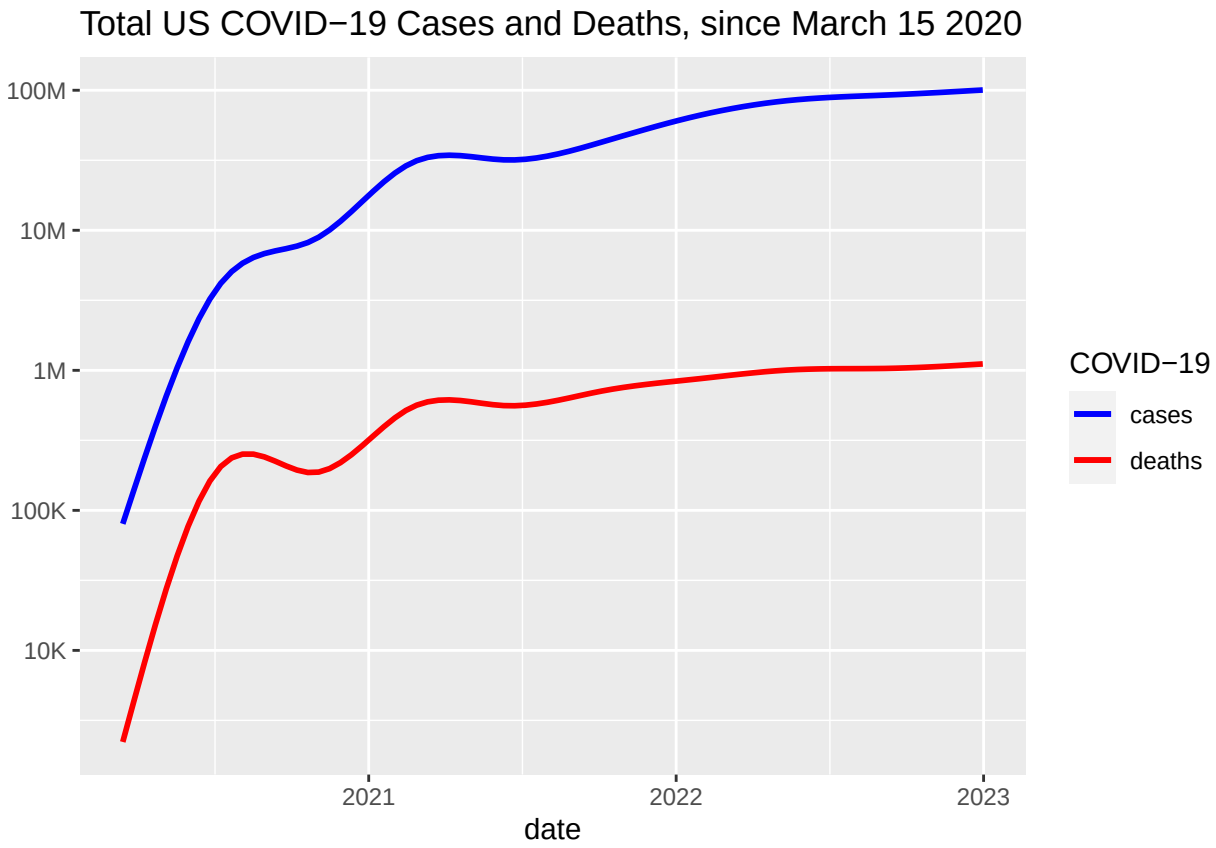
From March 15 2020, up until December 31, 2022, there had been 9.9374764×10^7 reported cases of COVID-19 in the United States (not including Puerto Rico), and 1.094296×10^6 deaths.

Question 2 Create a visualization for the total number of deaths and cases in the US since March 15, 2020. Before you create your visualization, review the types of plots you can create using the ggplot2 library and think about which plots would be effective in communicating your results. After you have created your visualization, write a few sentences describing your visualization. How could the plot be interpreted? Could it be misleading?

```
# Create a visualization for the total number of US cases and deaths since March 15, 2020.
#
## YOUR CODE HERE ##
us_counties_20to22 %>%
  group_by(date) %>%
  summarise(cases = sum(cases), deaths = sum(deaths)) %>%
  ggplot(aes(x = date)) +
  geom_smooth(aes(y = deaths, colour = "deaths"), se = FALSE) +
  geom_smooth(aes(y = cases, colour = "cases"), se = FALSE) +
  scale_color_manual(
    name = "COVID-19",
    values = c("cases" = "blue", "deaths" = "red")
    ) +
  scale_y_continuous(
    name = NULL,
    trans = "log10",
    labels = scales::label_number_si()
    ) +
  ggtitle("Total US COVID-19 Cases and Deaths, since March 15 2020")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



– Communicate your methodology, results, and interpretation here – The number of cases is vastly larger than the number of deaths. There are many options for visualising this. One would have been, for example, to report the rate of deaths per 10,000 cases. However, this wouldn't meet the task requirements of visualising both cases and deaths. So, I have used a log-scaled line graph.

The key drawback of this approach is that the Y-axis does not increase at a constant rate: it is exponential. This would make it difficult for some people to interpret.

The key strength of this approach, is that it is simple to visualise the 'plateau' of cases and deaths as time goes on. From mid-2021, the case and death totals increase at a much slower rate than early in the pandemic.

Question 3 While it is important to know the total deaths and cases throughout the COVID-19 pandemic, it is also important for local and state health officials to know the the number of new cases and deaths each day to understand how rapidly the virus is spreading. Using the table you created in Question 1, calculate the number of new deaths and cases each day and a seven-day average of new deaths and cases. Once you have organized your data, find the days that saw the largest number of new cases and deaths. Write a sentence or two after the code block communicating your results.

```
# Create a new table, based on the table from Question 1, and calculate the number of new deaths and ca
#
# Hint: Look at the documentation for lag() when computing the number of new deaths and cases and the s
#
#
## YOUR CODE HERE ##

#  total_deaths     >  the cumulative number of deaths up to and including the associated date
#  total_cases      >  the cumulative number of cases up to and including the associated date
#  delta_deaths_1   >  the number of new deaths since the previous day
```

```
#  delta_cases_1    >  the number of new cases since the previous day
#  delta_deaths_7   >  the average number of deaths in a seven-day period
#  delta_cases_7    >  the average number of cases in a seven-day period

totals <- us_counties_20to22 %>%
  group_by(date) %>%
  summarise(`total deaths` = sum(deaths),
            `total cases` = sum(cases)) %>%
  mutate(`delta_deaths_1` = `total deaths` - lag(`total deaths`),
         `delta_cases_1` = `total cases` - lag(x = `total cases`,
                                               n = 1, order_by = date),
         #using rollmean() from the zoo package for the seven0day rolling average
         `delta_deaths_7` = zoo::rollmean(`total deaths`, k = 7, fill = NA),
         `delta_cases_7` = zoo::rollmean(`total cases`, k = 7, fill = NA))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
totals
```

```
## # A tibble: 1,022 x 7
##    date       `total deaths` `total cases` delta_deaths_1 delta_cases_1
##    <date>              <dbl>         <dbl>          <dbl>         <dbl>
##  1 2020-03-15             68          3595             NA            NA
##  2 2020-03-16             91          4502             23           907
##  3 2020-03-17            117          5901             26          1399
##  4 2020-03-18            162          8345             45          2444
##  5 2020-03-19            212         12387             50          4042
##  6 2020-03-20            277         17998             65          5611
##  7 2020-03-21            359         24507             82          6509
##  8 2020-03-22            457         33050             98          8543
##  9 2020-03-23            577         43474            120         10424
## 10 2020-03-24            783         53899            206         10425
## # ... with 1,012 more rows, and 2 more variables: delta_deaths_7 <dbl>,
## #   delta_cases_7 <dbl>
```

```
us_total_cases <- max(totals$`total cases`) #maximum number of total cases
us_total_deaths <- max(totals$`total deaths`) #maximum number of total deaths

#maximum number of cases in a single day
max_new_cases <- totals %>%
  filter(delta_cases_1 == range(delta_cases_1, na.rm = TRUE)[2])
max_new_cases_date <- max_new_cases$date #select only the date

#maximum number of deaths in a single day
max_new_deaths <- totals %>%
  filter(delta_deaths_1 == range(delta_deaths_1, na.rm = TRUE)[2])
max_new_deaths_date <- max_new_deaths$date #select only the date
```

```
#  Your output should look similar to the following tibble:
#
#  date
#  total_deaths     >  the cumulative number of deaths up to and including the associated date
#  total_cases      >  the cumulative number of cases up to and including the associated date
#  delta_deaths_1   >  the number of new deaths since the previous day
#  delta_cases_1    >  the number of new cases since the previous day
```

```
#  delta_deaths_7   >  the average number of deaths in a seven-day period
#  delta_cases_7    >  the average number of cases in a seven-day period
#==
#  A tibble: 813 x 7
#    date           total_deaths   total_cases   delta_deaths_1    delta_cases_1  delta_deaths_7  delta
#    <date>            <dbl>          <dbl>         <dbl>             <dbl>        <dbl>           <
#  1 2020-03-15           68           3600            0                 0          NA
#  2 2020-03-16           91           4507           23               907          NA
#  3 2020-03-17          117           5906           26              1399          NA
#  4 2020-03-18          162           8350           45              2444          NA
#  5 2020-03-19          212          12393           50              4043          NA
#  6 2020-03-20          277          18012           65              5619          NA
#  7 2020-03-21          360          24528           83              6516          NA
#  8 2020-03-22          458          33073           98              8545        55.7              4
#  9 2020-03-23          579          43505          121             10432        69.7              5
# 10 2020-03-24          785          53938          206             10433        95.4              6
# ... with 803 more rows
```

– Communicate your methodology, results, and interpretation here –

The date with the largest increase in cases, was 2022-01-10, when 1.427097×10^6 were reported.

The date with the largest increase in deaths, was 2022-11-11, when 1.2715×10^4 were reported.

```
# Create a new table, based on the table from Question 3, and calculate the number of new deaths and ca

# Hint: To calculate per 100,000 people, first tidy the population estimates data and calculate the US
#
# Hint: look at the help documentation for grepl() and case_when() to divide the averages by the US pop
# For example, take the simple tibble, t_new:
#
#     x     y
#   <int> <chr>
#     1     a
#     2     b
#     3     a
#     4     b
#     5     a
#     6     b
#
#
# To add a column, z, that is dependent on the value in y, you could:
#
# t_new %>%
#   mutate(z = case_when(grepl("a", y) ~ "not b",
#                        grepl("b", y) ~ "not a"))
#

## YOUR CODE HERE ##
#  date
#  total_deaths     >  the cumulative number of deaths up to and including the associated date per 1000
#  total_cases      >  the cumulative number of cases up to and including the associated date per 10000
#  delta_deaths_1   >  the number of new deaths since the previous day per 100000 people
#  delta_cases_1    >  the number of new cases since the previous day per 100000 people
```

```
#  delta_deaths_7   >  the average number of deaths in a seven-day period per 100000 people
#  delta_cases_7    >  the average number of cases in a seven-day period per 100000 people

US_Pop <- us_population_estimates <- read_csv("fips_population_estimates.csv")
```

Question 4

```
## Parsed with column specification:
## cols(
##   STNAME = col_character(),
##   CTYNAME = col_character(),
##   fips = col_double(),
##   STATE = col_double(),
##   COUNTY = col_double(),
##   Year = col_double(),
##   Estimate = col_double()
## )
```

```
Pop_Estimate <- US_Pop %>%
  group_by(Year) %>%
  summarise("Estimate" = sum(Estimate))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Pop_2020 <- as.integer(Pop_Estimate[1,2])
Pop_2021 <- as.integer(Pop_Estimate[2,2])

Proportional_Totals <- us_counties_20to22 %>%
  group_by(date) %>%
  summarise("deaths" = sum(deaths),
            "cases" = sum(cases))%>%
  mutate(
    "Pop_Estimate" = case_when(
      grepl("2020", date) ~ Pop_2020,
      grepl("2021", date) ~ Pop_2021)
  ) %>%
  mutate(
    "total_deaths" = (deaths/Pop_Estimate)*100000,
    "total_cases" = (cases/Pop_Estimate)*100000
  ) %>%
  mutate(
    "delta_deaths_1" = total_deaths - lag(total_deaths, n = 1, order_by = date),
    "delta_cases_1" = total_cases - lag(x = total_cases, n = 1, order_by = date)
  ) %>%
  mutate(
    "delta_deaths_7" = zoo::rollmean(`delta_deaths_1`, k = 7, fill = NA),
    "delta_cases_7" = zoo::rollmean(`delta_cases_1`, k = 7, fill = NA)
  ) %>%
  select("date" , `total_deaths`:`delta_cases_7`)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Proportional_Totals
```

```
## # A tibble: 1,022 x 7
##    date       total_deaths total_cases delta_deaths_1 delta_cases_1
##    <date>            <dbl>       <dbl>          <dbl>         <dbl>
```

```
##  1 2020-03-15       0.0205        1.08       NA             NA
##  2 2020-03-16       0.0275        1.36        0.00694        0.274
##  3 2020-03-17       0.0353        1.78        0.00784        0.422
##  4 2020-03-18       0.0489        2.52        0.0136         0.737
##  5 2020-03-19       0.0640        3.74        0.0151         1.22
##  6 2020-03-20       0.0836        5.43        0.0196         1.69
##  7 2020-03-21       0.108         7.39        0.0247         1.96
##  8 2020-03-22       0.138         9.97        0.0296         2.58
##  9 2020-03-23       0.174        13.1         0.0362         3.14
## 10 2020-03-24       0.236        16.3         0.0621         3.14
## # ... with 1,012 more rows, and 2 more variables: delta_deaths_7 <dbl>,
## #   delta_cases_7 <dbl>
```

```
#  Your output should look similar to the following tibble:
#
#  date
#  total_deaths     >  the cumulative number of deaths up to and including the associated date
#  total_cases      >  the cumulative number of cases up to and including the associated date
#  delta_deaths_1   >  the number of new deaths since the previous day
#  delta_cases_1    >  the number of new cases since the previous day
#  delta_deaths_7   >  the average number of deaths in a seven-day period
#  delta_cases_7    >  the average number of cases in a seven-day period
#==
#  A tibble: 657 x 7
#       date         total_deaths   total_cases   delta_deaths_1   delta_cases_1 delta_deaths_7 delta_c
#      <date>            <dbl>         <dbl>          <dbl>           <dbl>           <dbl>         <db
#   1 2020-03-15         0.0205        1.08               0               0              NA           N
#   2 2020-03-16         0.0275        1.36         0.00694         0.274                NA           N
#   3 2020-03-17         0.0353        1.78         0.00784         0.422                NA           N
#   4 2020-03-18         0.0489        2.52          0.0136         0.737                NA           N
#   5 2020-03-19         0.0640        3.74          0.0151          1.22                NA           N
#   6 2020-03-20         0.0836        5.43          0.0196          1.69                NA           N
#   7 2020-03-21         0.108         7.39          0.0247          1.96                NA           N
#   8 2020-03-22         0.138         9.97          0.0296          2.58           0.0168         1.2
#   9 2020-03-23         0.174        13.1           0.0362          3.14           0.0209         1.6
#  10 2020-03-24         0.236        16.3           0.0621          3.14           0.0287         2.0
```

– Communicate your methodology, results, and interpretation here – I used a lot of sequential mutates, I know it makes my code messier, but I decided to leave it as is, because I found it useful to be able to isolate and troubleshoot each step.

In terms of interpretation, there's not much to add. The data is the same as Q3, just transformed to a proportion per 100000 people.

The flaw, is that the population estimate data is only for 2020 and 2021, so the proportions for 2022 were not able to be calculated.

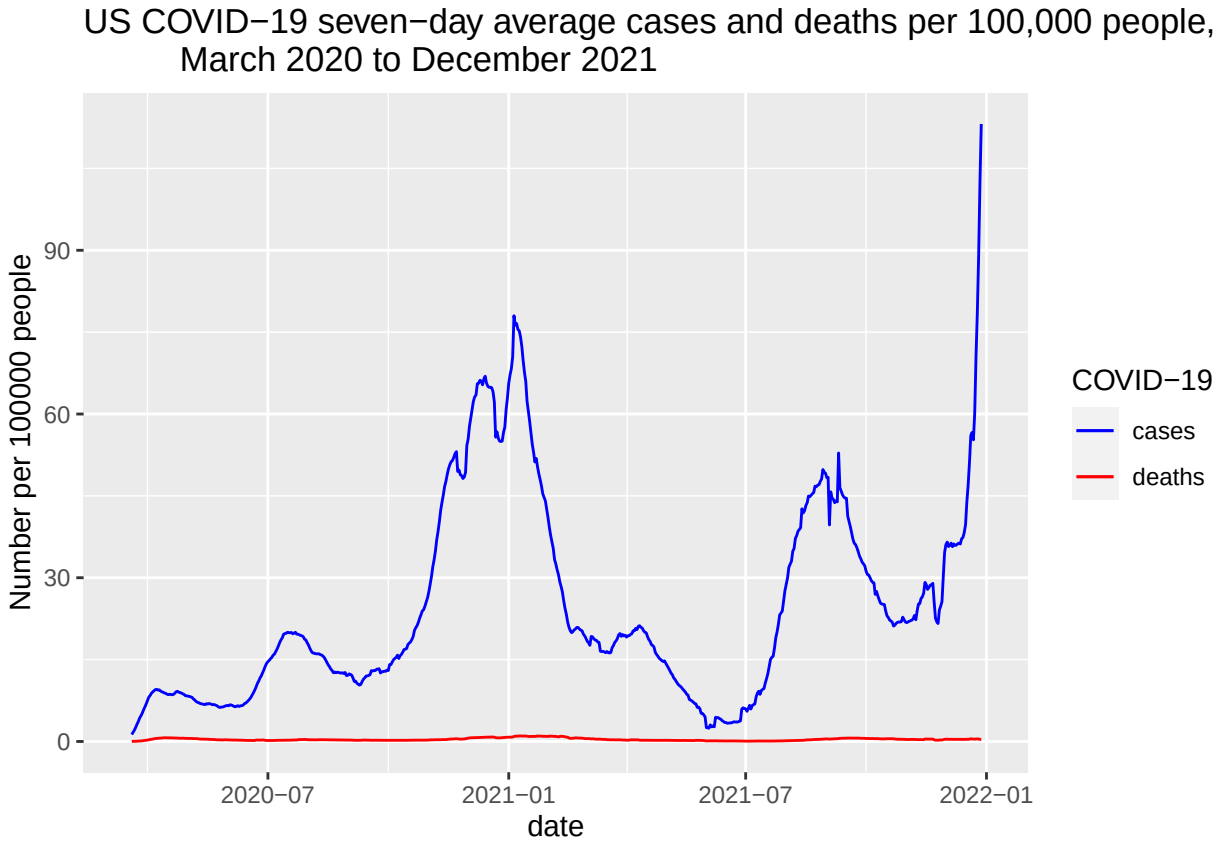
```
# Create a visualization to compare the seven-day average cases and deaths per 100,000 people.
Proportional_Totals %>%
  filter(!date > "2021-12-31") %>%  #no population estimates data for 2022 or 2023, so filter out.
  ggplot(aes(x = date)) +
  geom_line(aes(y = delta_cases_7, colour = "cases")) +
  geom_line(aes(y = delta_deaths_7, colour = "deaths")) +
  scale_color_manual(
    name = "COVID-19",
```

```
    values = c("cases" = "blue", "deaths" = "red")
  ) +
  scale_y_continuous(name = "Number per 100000 people") +
  ggtitle("US COVID-19 seven-day average cases and deaths per 100,000 people,
          March 2020 to December 2021")
```

Question 5

```
## Warning: Removed 7 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 7 row(s) containing missing values (geom_path).
```



– Communicate your methodology, results, and interpretation here –

Because of mission population estimates for 2022, this visualisation only covers 2020 and 2021. This is made explicit in the title.

I have used an un-scaled y-axis for this visualisation. It is thus a much clearer representation than for the previous question where I used a log transformation on the y-axis.

The most striking feature is the apparent low rate of death, which hides the absolute magnitude: there were more than 800,000 deaths in the period.