Code last run 2021-02-18.
Daily: Data as of January 29, 2021.
Neighbourhood: Data as of February 9, 2021.

Task 1: Daily cases

Data wrangling

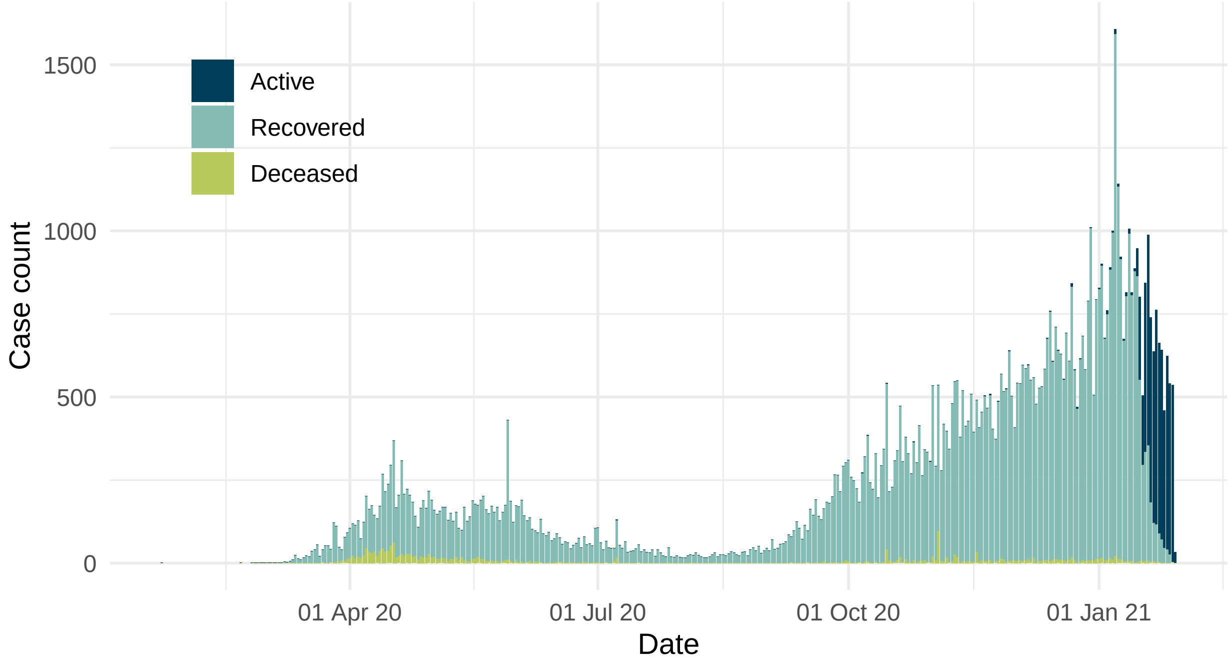
```
#To replace all NAs in numeric columns, only work on columns
reported <- reported_raw %>%
  mutate_if(is.numeric, replace_na, replace = 0)

#make sure the reported date into date format
reported$reported_date <- date(reported$reported_date)


#very important, increasing # of rows, decreasing # of cols
new_reported <- reported %>%
  pivot_longer(-c(reported_date),
               names_to = "status", values_to = "number")

#change the string to sentence case, which means the first letter is capitalized
new_reported$status <- str_to_sentence(new_reported$status)

#factor the variable "status"
new_reported$status <- factor(new_reported$status,
                              levels = c("Active", "Recovered", "Deceased"))
```

Data visualization

```
new_reported %>%
  ggplot(aes(x = reported_date, y = number, fill = status)) +
  scale_x_date(breaks = "3 months", date_labels = "%d %b %y")+
#stat = "identity" means I tell ggplot to skip aggregation, state that I will provide y values
  geom_bar(stat = "identity") +
# theme_minimal() means no background annotations
  theme_minimal() +
#add title, subtitle..caption explanation by using lab() in ggplot
  labs(title = "Cases reported by day in Toronto, Canada",
       subtitle = "Confirmed and probable cases",
       x = "Date",
       y = "Case count",
       caption = str_c("Created by: <Zishu Zhu> for STA303/1002, U of T\n",
"Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES\n",
format(Sys.time(), "Data as of %B %d, %Y"))) +
#no legend title and self-define the legend postion
  theme(legend.title=element_blank(), legend.position=c(0.15, 0.8)) +
#fill the bar chart by self-defined color, not automatically
  scale_fill_manual(values=c("#003F5C", "#86BCB6", "#B9CA5D"),
                    breaks=c("Active", "Recovered", "Deceased"))
```

Cases reported by day in Toronto, Canada

Confirmed and probable cases



Created by: <Zishu Zhu> for STA303/1002, U of T
Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES
Data as of February 18, 2021

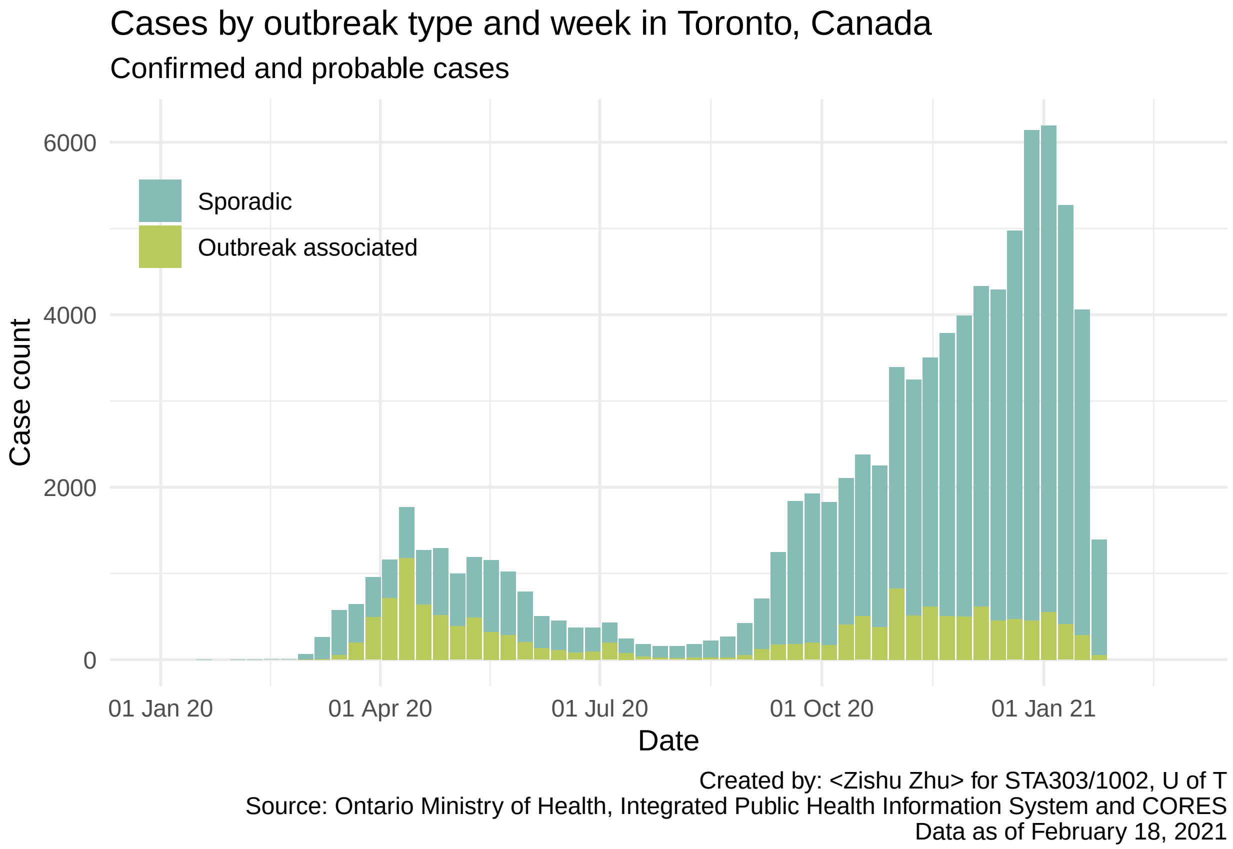
Task 2: Outbreak type

Data wrangling

```
#change the coloumn into date format
outbreak <- outbreak_raw
outbreak$episode_week <- date(outbreak$episode_week)


#rename level in variable outbreak_or_sporadic
outbreak$outbreak_or_sporadic <- str_replace_all(outbreak$outbreak_or_sporadic,
                                                 "OB Associated", "Outbreak associated")

#group the dataframe by episode week and calculate the sum of
#cases in each week by group_by() and sumarise()
outbreak %>% group_by(episode_week) -> temp
new_outbreak_total <- summarise(temp, total_case = sum(cases))

#merge every rows of outbreak and any matching rows in new_outbreak_total
outbreak <- left_join(outbreak, new_outbreak_total,
                      by = "episode_week")

#we can use factor() to store all the strings and integers as levels
outbreak$outbreak_or_sporadic <- factor(outbreak$outbreak_or_sporadic,
                                        levels = c("Sporadic", "Outbreak associated"))
```

Data visualization

```
#draw a bar-chart to compare sporadic and outbreak cases from day to day
outbreak %>%
  ggplot(aes(x = episode_week, y = cases, fill = outbreak_or_sporadic)) +
#change x to date format range from Jan1 2020 to the present day + 7 days
  scale_x_date(labels = scales::date_format("%d %b %y"), limits
= c(date("2020-01-01"), Sys.Date()+7))+
  geom_bar(stat = "identity") +
# theme_minimal() means no background annotations
  theme_minimal() +
  labs(title = "Cases by outbreak type and week in Toronto, Canada",
       subtitle = "Confirmed and probable cases",
       x = "Date",
       y = "Case count",
       caption = str_c("Created by: <Zishu Zhu> for STA303/1002, U of T\n",
"Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES\n",
format(Sys.time(), "Data as of %B %d, %Y"))) +
#no legend title and self-define the legend postion
  theme(legend.title=element_blank(), legend.position=c(0.15, 0.8)) +
#fill the bar chart by self-defined color, not automatically
  scale_fill_manual(values=c("#86BCB6", "#B9CA5D"), breaks=c("Sporadic", "Outbreak associated"))
```



Task 3: Neighbourhoods

Data wrangling: part 1

```
#find the row that indicate the % of low income people for each neighnourhood in toronto
nbhood_profile %>% filter(`_id` == 1143) -> income

#increase rows and decrese columns
income  = income %>% pivot_longer(-c(1:5), names_to = "neighbourhood_name",
                                  values_to = "percentage") %>%
#ignore all the non numeric characters
  mutate(percentage = parse_number(percentage))
#only keep the column 6 and column 7
income <- income[,6:7]
```

Data wrangling: part 2

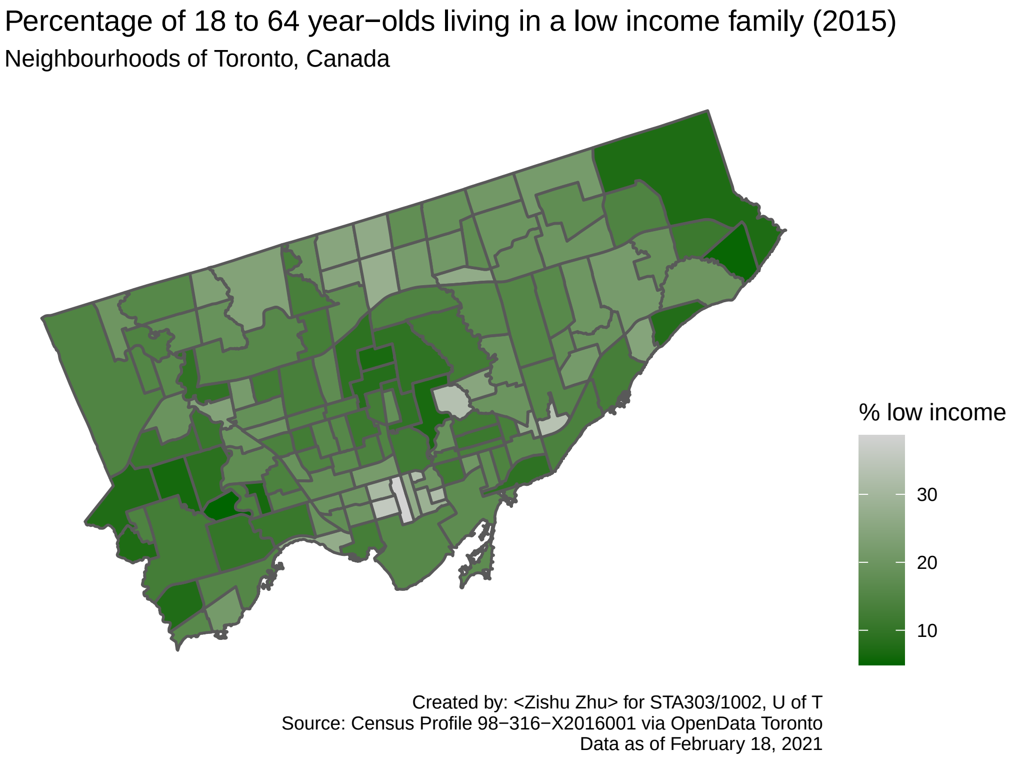
```
nbhoods_shape_raw %>%
#use str_remove to remove all the number in parentheses and the space
  mutate(neighbourhood_name = str_remove(AREA_NAME, "\\s\\(\\d+\\)$")) %>%
#keep the levels of neighbourhood name in nbhoods_shape_raw same as income
  mutate(neighbourhood_name = case_when(
    neighbourhood_name == "North St.James Town" ~ "North St. James Town",
    neighbourhood_name == "Cabbagetown-South St.James Town"
    ~"Cabbagetown-South St. James Town",
     neighbourhood_name == "Weston-Pellam Park" ~"Weston-Pelham Park",
#TRUE~means the other stays the same, no need to write all cases
    TRUE~neighbourhood_name
  )) -> nbhoods

#merge every rows of nbhoods and any matching rows in income by "neighbourhood_name"
comb <- left_join(nbhoods, income, by="neighbourhood_name")
combination <- left_join(comb, nbhood_raw, by="neighbourhood_name")
#rename column rate_per_100_000_people
nbhoods_all <- combination %>% rename(rate_per_100000 = rate_per_100_000_people)
```

Data wrangling: part 3

```
#remove NAs by na.rm
med_inc <- median(nbhoods_all$percentage, na.rm = TRUE)
med_rate <- median(nbhoods_all$rate_per_100000, na.rm = TRUE)
#mutate() create new variables
nbhoods_final <- nbhoods_all %>%
  mutate(
    nbhood_type = case_when(
      percentage >= med_inc & rate_per_100000 >= med_rate
      ~ "Higher low income rate, higher case rate",
     percentage >= med_inc & rate_per_100000 < med_rate
     ~ "Higher low income rate, lower case rate",
      percentage < med_inc & rate_per_100000 >= med_rate
     ~ "Lower low income rate, higher case rate",
      percentage < med_inc & rate_per_100000 < med_rate
     ~ "Lower low income rate, higher case rate"))
```

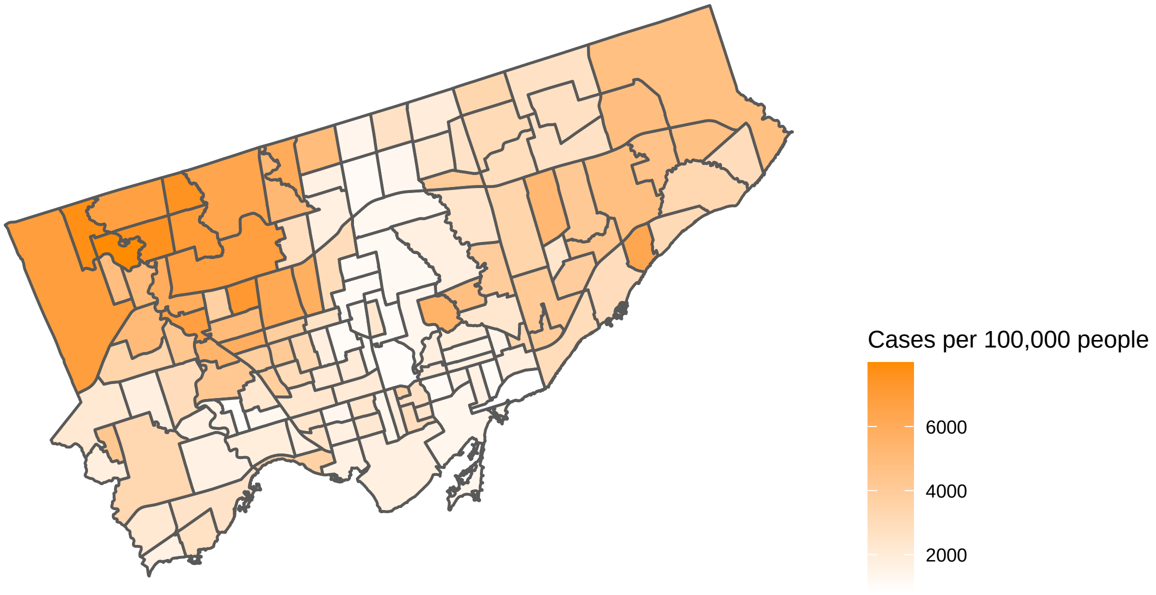
Data visualization

```r
ggplot() +
#we got shape data for mapping in second chunk, now we use the data to draw a toronto map
geom_sf(data = nbhoods_final, aes(fill = percentage)) +
theme_map() +
#change the legend position
theme(legend.position = "right")+
#make the color of legend gradient
  scale_fill_gradient(name= "% low income", low = "darkgreen", high = "lightgrey")+
#add title, subtitle..caption explanation by using lab() in ggplot
  labs(title = "Percentage of 18 to 64 year-olds living in a low income family (2015)",
       subtitle = "Neighbourhoods of Toronto, Canada",
       caption = str_c("Created by: <Zishu Zhu> for STA303/1002, U of T\n",
                       "Source: Census Profile 98-316-X2016001 via OpenData Toronto\n",
#\n tells r to start a new line, Sys.time() can return the present time
                       format(Sys.time(), "Data as of %B %d, %Y")))
```


Percentage of 18 to 64 year−olds living in a low income family (2015)
Neighbourhoods of Toronto, Canada
% low income
30
20
10
Created by: <Zishu Zhu> for STA303/1002, U of T
Source: Census Profile 98−316−X2016001 via OpenData Toronto
Data as of February 18, 2021

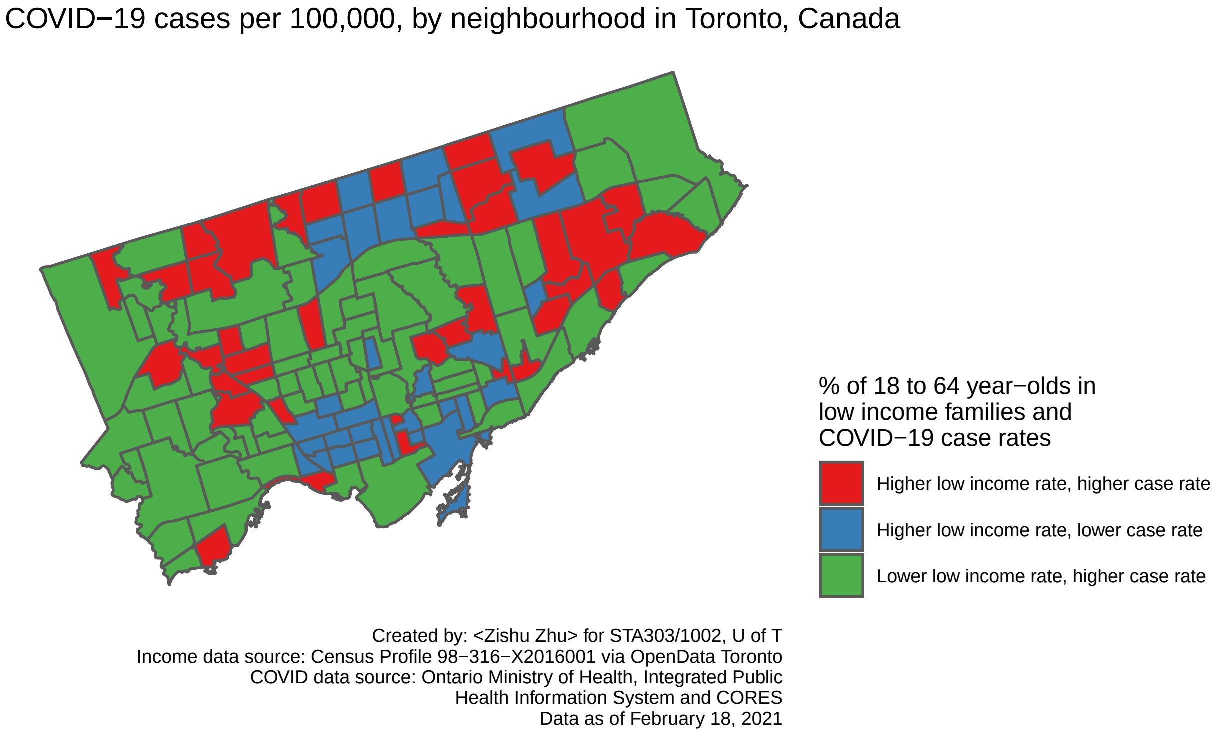
```
ggplot() +
#draw a toronto map, fill color by rate_per_100000
geom_sf(data = nbhoods_final, aes(fill = rate_per_100000)) +
theme_map() +
theme(legend.position = "right")+
#make the color of legend gradient
  scale_fill_gradient(name= "Cases per 100,000 people", low = "white", high = "darkorange")+
  labs(title = "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
       caption = str_c("Created by: <Zishu Zhu> for STA303/1002, U of T\n",
"Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES\n",
#\n tells r to start a new line, Sys.time() can return the present time
                       format(Sys.time(), "Data as of %B %d, %Y")))
```

COVID−19 cases per 100,000, by neighbourhood in Toronto, Canada



Created by: <Zishu Zhu> for STA303/1002, U of T
Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES
Data as of February 18, 2021

```
ggplot() +
geom_sf(data = nbhoods_final, aes(fill = nbhood_type)) +
theme_map() +
theme(legend.position = "right")+
#use color palette to color the toronto map by nbhood_type and change the legend name
  scale_fill_brewer(palette = "Set1",
  name = "% of 18 to 64 year-olds in\nlow income families and\nCOVID-19 case rates")+
  labs(title = "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
#caption = str_c() to add captions to explain the graph.
       caption = str_c("Created by: <Zishu Zhu> for STA303/1002, U of T\n",
        "Income data source: Census Profile 98-316-X2016001 via OpenData Toronto\n",
              "COVID data source: Ontario Ministry of Health, Integrated Public\n",
                       "Health Information System and CORES\n",
#\n tells r to start a new line, Sys.time() can return the present time
                       format(Sys.time(), "Data as of %B %d, %Y")))
```


COVID−19 cases per 100,000, by neighbourhood in Toronto, Canada

% of 18 to 64 year−olds in
low income families and
COVID−19 case rates
🟥 Higher low income rate, higher case rate
🟦 Higher low income rate, lower case rate
🟩 Lower low income rate, higher case rate

Created by: <Zishu Zhu> for STA303/1002, U of T
Income data source: Census Profile 98−316−X2016001 via OpenData Toronto
COVID data source: Ontario Ministry of Health, Integrated Public
Health Information System and CORES
Data as of February 18, 2021