Code last run 2021-02-16.
Daily: Data as of January 29, 2021.
Neighbourhood: Data as of January 28, 2021.

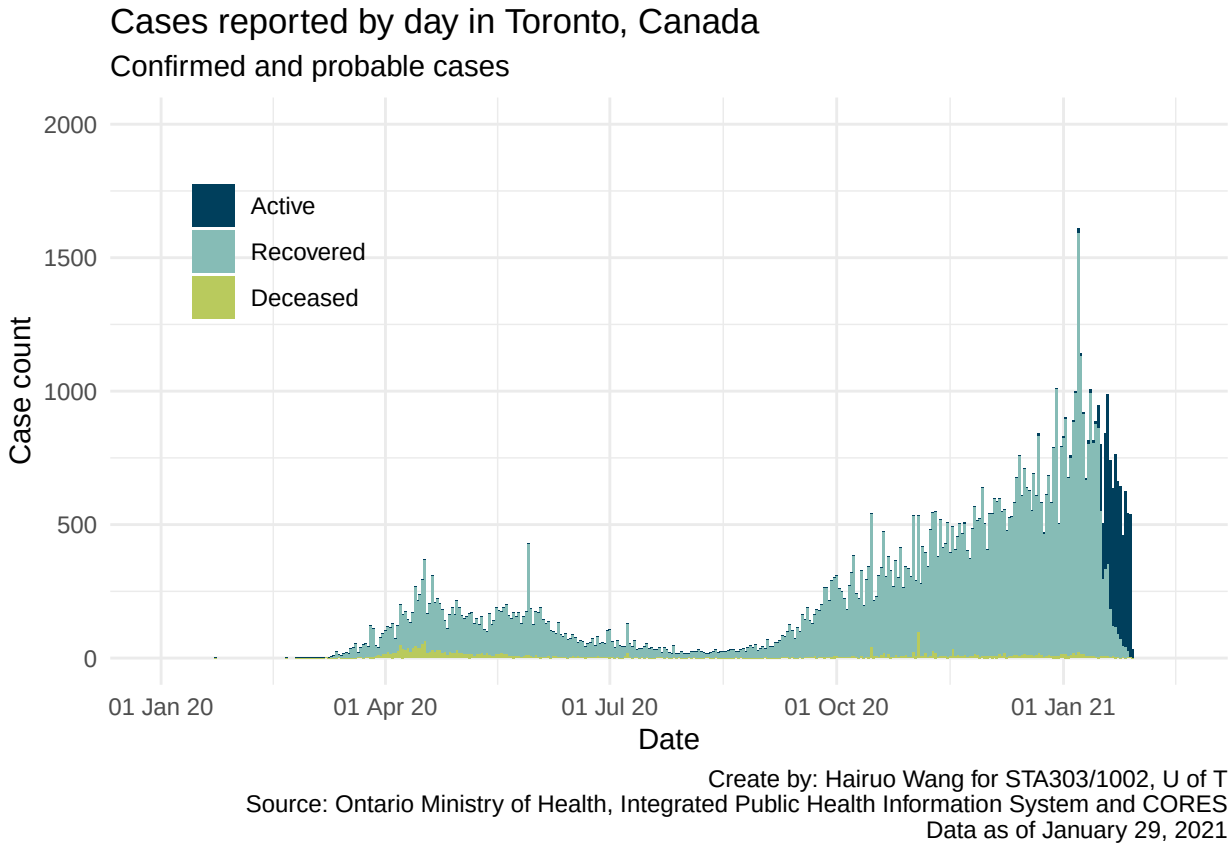
Task 1: Daily cases

Data wrangling

```
#create new dataset reported and start with reported_raw
reported = reported_raw %>%
  #replace all NA values with 0
  mutate_if(is.numeric, replace_na, replace=0) %>%
  #overwrite reported_date with date version
  mutate(reported_date=date(reported_date)) %>%
  #the data is not tidy, so alter it to be tidy.
  pivot_longer(c(recovered, active, deceased), names_to = "Status",
               values_to="Number_of_Cases") %>%
  #capitalized
  mutate(Status = str_to_sentence(Status)) %>%
  #reorder the status and they could appear in correct order in the legend
  mutate(Status = fct_relevel(Status, "Deceased", after=2))
```

Data visualization

```
#visualize reported dataset
reported %>%
  #Make x-axis, y-axis and fill
  ggplot(aes(x=reported_date, y=Number_of_Cases, fill = Status)) +
  #bar plot
  geom_bar(stat="identity")+
  #use minimal theme of the figure
  theme_minimal()+
  #add title, subtitle, number of x-axis, y-axis and caption of the figure
  labs(title = "Cases reported by day in Toronto, Canada",
       subtitle = "Confirmed and probable cases",
       x= "Date",
       y= "Case count",
       caption = str_c("Create by: Hairuo Wang for STA303/1002, U of T\n",
                       "Source: Ontario Ministry of Health, Integrated Public",
                       " Health Information System and CORES\n",
                       date_daily[1, 1])) +
  #set the limit and format of date on x-axis
  scale_x_date(labels = scales::date_format("%d %b %y"),
               limits =c(date("2020-01-01"), Sys.Date()))+
  #legend without title and set the position
  theme(legend.title = element_blank(), legend.position = c(0.15, 0.75))+
  #set the color of fill
  scale_fill_manual(values = c("#003F5C", "#86BCB6", "#B9CA5D"))+
  #set the limit of value on y-axis
  scale_y_continuous(limits = c(0, 2000))
```


Cases reported by day in Toronto, Canada
Confirmed and probable cases
Active
Recovered
Deceased
Case count
2000
1500
1000
500
0
01 Jan 20
01 Apr 20
01 Jul 20
01 Oct 20
01 Jan 21
Date
Create by: Hairuo Wang for STA303/1002, U of T
Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES
Data as of January 29, 2021

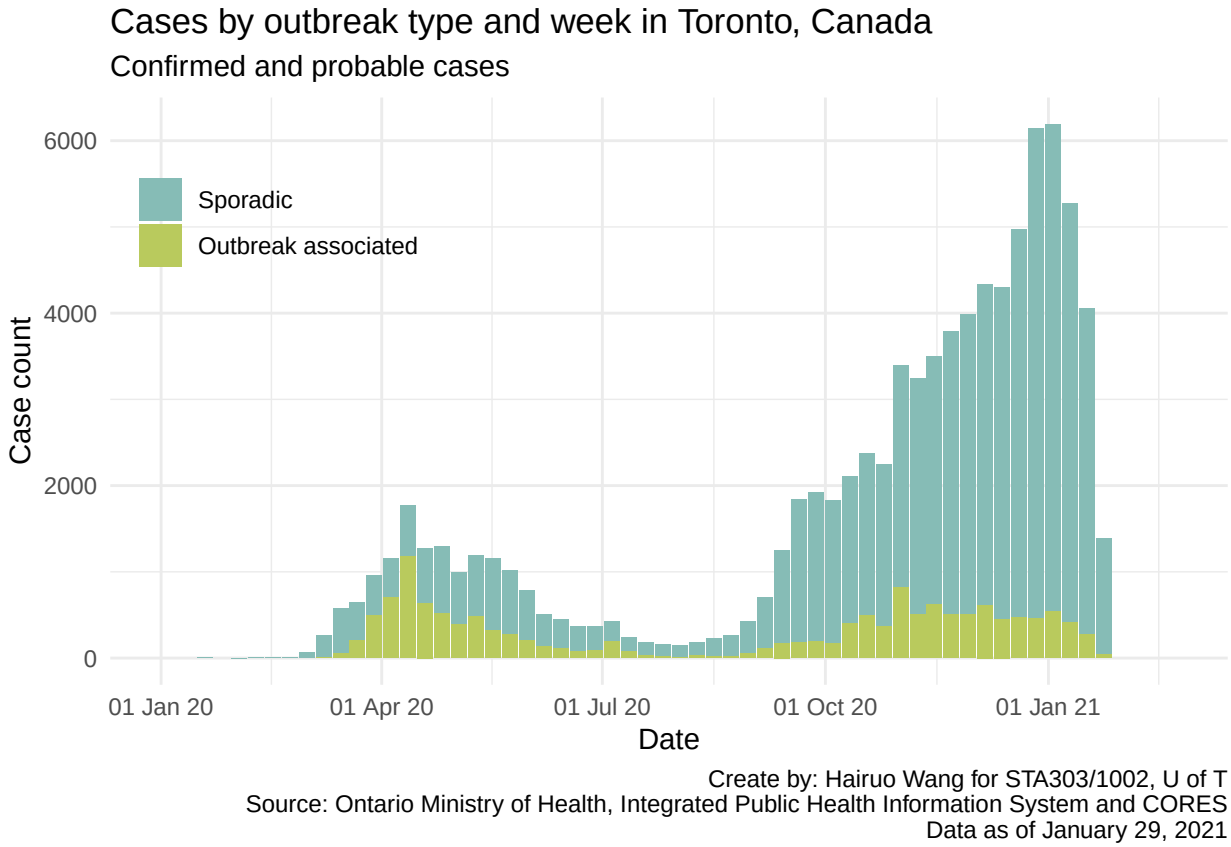
Task 2: Outbreak type

Data wrangling

```
#create new dataset outbreak and start with outbreak_raw
outbreak = outbreak_raw %>%
  #overwrite episode_week with date version
  mutate(episode_week=date(episode_week)) %>%
  #replace the name in order to get the same result with the given figure
  mutate(outbreak_or_sporadic = str_replace_all(outbreak_or_sporadic, "OB",
                                                "outbreak")) %>%
  #capitalized word in order to get the same result with the given figure
  mutate(outbreak_or_sporadic = str_to_sentence(outbreak_or_sporadic)) %>%
  #reorder the values
  mutate(outbreak_or_sporadic = fct_rev(outbreak_or_sporadic)) %>%
  #create a new variable which is number of cases in episode week
  group_by(episode_week) %>%
  mutate(total_case = sum(cases))
```

Data visualization

```
#visualize outbreak dataset
outbreak %>%
  #make x-axis, y-axis and fill
  ggplot(aes(x=episode_week, y=cases, fill = outbreak_or_sporadic)) +
  #bar plot
  geom_bar(stat="identity") +
  #minimal theme
  theme_minimal()+
  #add title, subtitle, number of x-axis, y-axis and caption
  labs(title = "Cases by outbreak type and week in Toronto, Canada",
       subtitle = "Confirmed and probable cases",
       x = "Date",
       y = "Case count",
       caption = str_c("Create by: Hairuo Wang for STA303/1002, U of T\n",
                       "Source: Ontario Ministry of Health, Integrated Public",
                       " Health Information System and CORES\n",
                       date_daily[1, 1]))+
  #set limit and format of date on x-axis
  scale_x_date(labels = scales::date_format("%d %b %y"),
               limits = c(date("2020-01-01"), Sys.Date()+7))+
  #set limit f value on y-axis
  scale_y_continuous(limits=c(0, max(outbreak$total_case)))+
  #legend with no title and set position of legend
  theme(legend.title=element_blank(), legend.position=c(.15, .8))+
  #set color of fills
  scale_fill_manual(values=c("#86BCB6", "#B9CA5D"))
```


Cases by outbreak type and week in Toronto, Canada
Confirmed and probable cases
6000
4000
2000
0
Case count
Sporadic
Outbreak associated
01 Jan 20
01 Apr 20
01 Jul 20
01 Oct 20
01 Jan 21
Date
Create by: Hairuo Wang for STA303/1002, U of T
Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES
Data as of January 29, 2021

Task 3: Neighbourhoods

Data wrangling: part 1

```
#create new dataset income and start with variable nbhood_profile
income = nbhood_profile %>%
  #choose the row with id 1143 since it is relevant
  filter(`_id` == 1143) %>%
  #The data is not tidy thus alter it to be tidy
  pivot_longer(-c(`_id`, Category, Topic, `Data Source`, Characteristic),
               names_to = "neighbourhood_name", values_to = "low_income_percentage") %>%
  #convert percentage from character strings to numbers
  mutate(low_income_percentage=parse_number(low_income_percentage)) %>%
  #since the neighbourhood name is not same in two dataset, so need to make it same.
  mutate(neighbourhood_name = str_replace(neighbourhood_name, "Pelham", "Pellam")) %>%
  mutate(neighbourhood_name = str_replace(neighbourhood_name, ". J", ".J"))
```

Data wrangling: part 2

```
#create a new dataset nbhood_raw_new since we need to ensure the neighbourhood are correctly matched
#thus we want to change some element in dataset nbhood_raw
nbhood_raw_new = nbhood_raw %>%
  #remove the useless and not matched column
  filter(neighbourhood_name != "Missing Address/Postal Code") %>%
  #make the neighbourhood name same
  mutate(neighbourhood_name = str_replace(neighbourhood_name, "Pelham", "Pellam")) %>%
  mutate(neighbourhood_name = str_replace(neighbourhood_name, ". J", ".J"))
#create a new dataset nbhoods_all and start with dataset nbhoods_shape_new
nbhoods_all= nbhoods_shape_raw %>%
  #remove space and number in parentheses in AREA_NAME variable
  #and save them into new variable
  mutate(neighbourhood_name = str_remove(AREA_NAME, "\\s\\(\\d+\\)$")) %>%
  #merge income so we have low income percentages
  left_join(income, by = "neighbourhood_name") %>%
  #merge nvhood_raw_new so we have case per 100000 people and case rate
  left_join(nbhood_raw_new, by = "neighbourhood_name") %>%
  #rename the case rate variable
  rename(rate_per_100000 = rate_per_100_000_people)
```

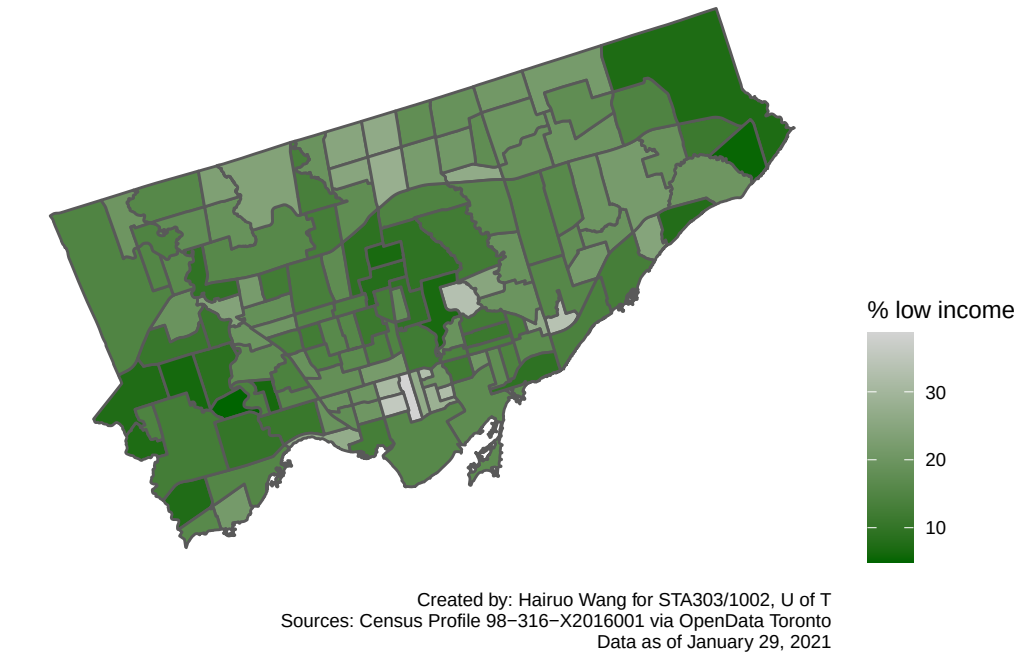
Data wrangling: part 3

```
#create a new dataset and start with dataset nbhoods_all
nbhoods_final = nbhoods_all %>%
  #create a new variable med_inc and get the median of low_income_percentage overall
  mutate(med_inc = median(low_income_percentage)) %>%
  #create a new variable med_rate and get the median of case rate overall
  mutate(med_rate = median(rate_per_100000)) %>%
  #create a new variable nbhood_type and classify the type of cases
  mutate(nbhood_type = case_when(
    low_income_percentage >= med_inc&rate_per_100000 >= med_rate ~
      "Higher low income rate, higher case rate",
    low_income_percentage >= med_inc&rate_per_100000 < med_rate ~
      "Higher low income rate, lower case rate",
```

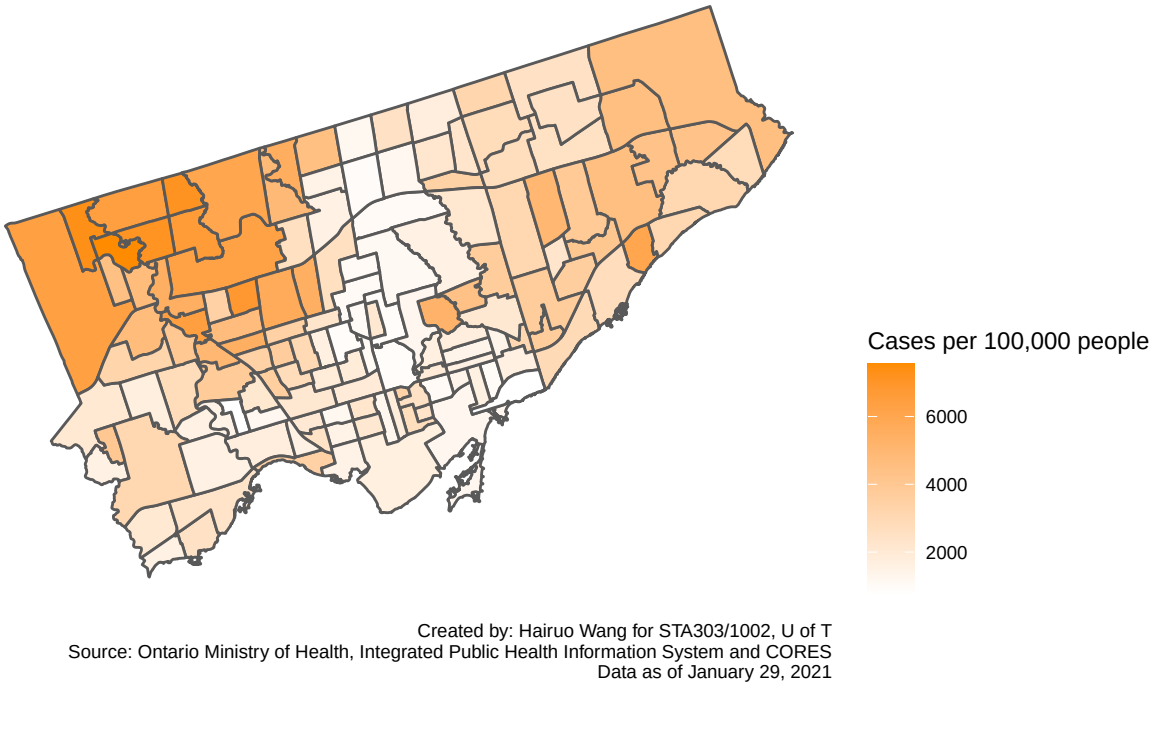
```
    low_income_percentage < med_inc&rate_per_100000 >= med_rate ~
      "Lower low income rate, higher case rate",
    low_income_percentage < med_inc&rate_per_100000 < med_rate ~
      "Lower low income rate,lower case rate"
  ))
```

Data visualization

```
#visualize percentage low income within map
ggplot(data=nbhoods_final) +
  #set fills
  geom_sf(aes(fill = low_income_percentage))+
  #create map
  theme_map()+
  #add title, subtitle and caption of the figure
  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: Hairuo Wang for STA303/1002, U of T\n",
                       "Sources: Census Profile 98-316-X2016001 via OpenData Toronto\n",
                       date_daily[1, 1]))+
  #set name of legend and color of degree
  scale_fill_gradient(name="% low income", low= "darkgreen", high="lightgrey")+
  #set the position of the legend
  theme(legend.position="right")
```



```
#visualize case rate within map
ggplot(data=nbhoods_final) +
  #set fills
  geom_sf(aes(fill = rate_per_100000))+
  #create map
  theme_map()+
  #add title and caption of the figure
  labs(title = "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
       caption = str_c("Created by: Hairuo Wang for STA303/1002, U of T\n",
                       "Source: Ontario Ministry of Health, Integrated Public Health",
                       " Information System and CORES\n",
                       date_daily[1, 1]))+
  #set name of legend and color of degree
  scale_fill_gradient(name="Cases per 100,000 people", low= "white", high="darkorange")+
  #set position of legend
  theme(legend.position="right")
```



```
#visualize type of case and low income within map
ggplot(data=nbhoods_final) +
  #set fill
  geom_sf(aes(fill = nbhood_type))+
  #create map
  theme_map()+
  #add title and caption
  labs(title = "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
       caption = str_c("Created by: Hairuo Wang 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",
                       date_daily[1, 1]))+
  #set title of legend and color of each type
  scale_fill_brewer(palette = "Set1", name=str_c("% of 18 to 64 year-olds in\n",
                                                 "low income families and\n",
                                                 "COVID-19 case rates"))+
  #set position of legend
  theme(legend.position="right")
```