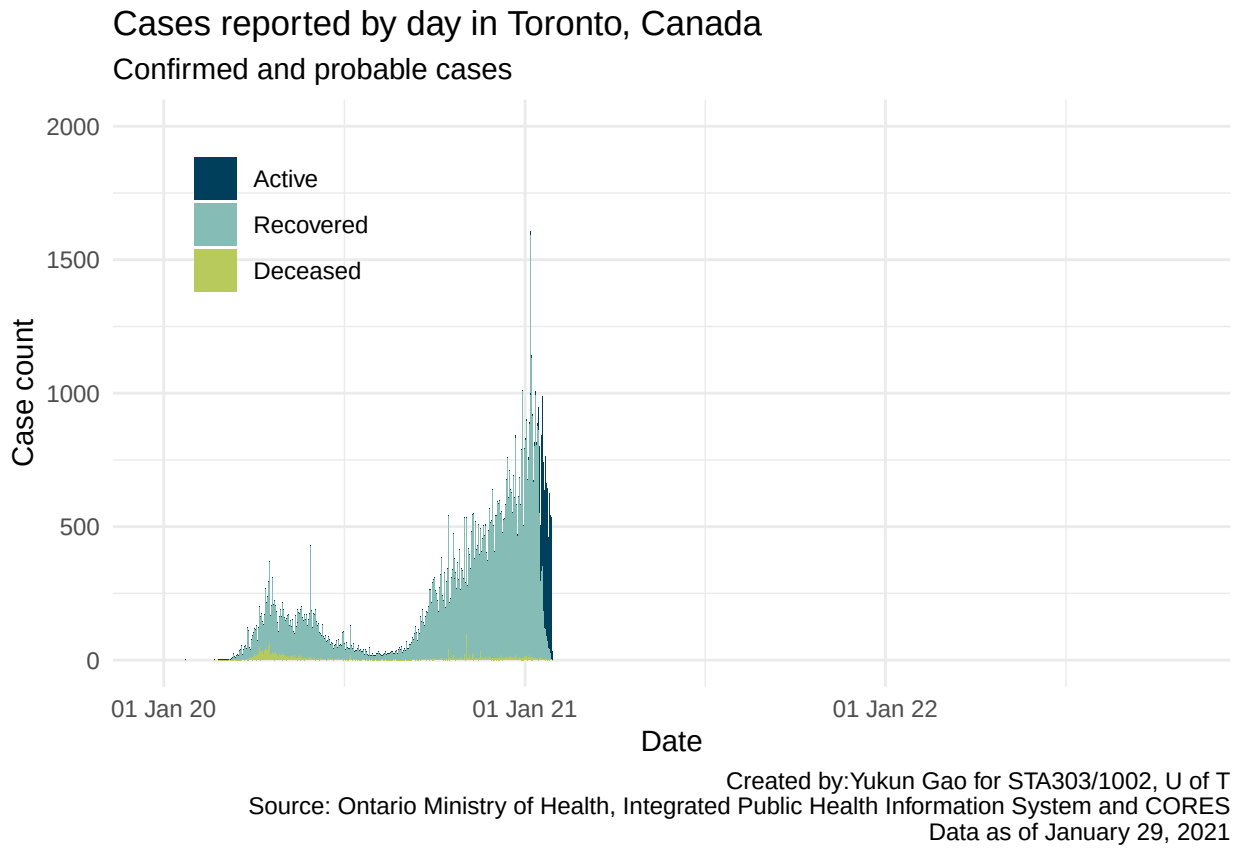
Task 1: Daily cases

Data wrangling

```
reported<- reported_raw %>%
   mutate_if(is.numeric,replace_na,replace=0) %>% #replaceNA with0
   mutate(reported_date= date(reported_date))%>%
   pivot_longer(-c(reported_date),names_to="Type", values_to="Number")%>%
   mutate(Type= str_to_sentence(Type))%>%
   mutate(Type=fct_relevel(Type, "Deceased",after = 2)) #make it appear in correct order
```

Data visualization

```
reported %>%
  ggplot(aes(x = reported_date, y = Number, fill = Type)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Cases reported by day in Toronto, Canada",
       subtitle = "Confirmed and probable cases",
       x = "Date",
       y = "Case count",
       caption = str_c("Created by:Yukun Gao for STA303/1002, U of T\n",
      "Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES\n",
                  date_daily[1,1])) +
  theme(legend.title = element_blank(),
        legend.position = c(0.15, 0.8)) +
  scale_x_date(labels = scales::date_format("%d %b %y"),
               limits= c(date("2020-01-01"),Sys.Date()))+
  scale_y_continuous(limits=c(0,2000))+
  scale_fill_manual(values=c("#003F5C","#86BCB6","#B9CA5D") )
```



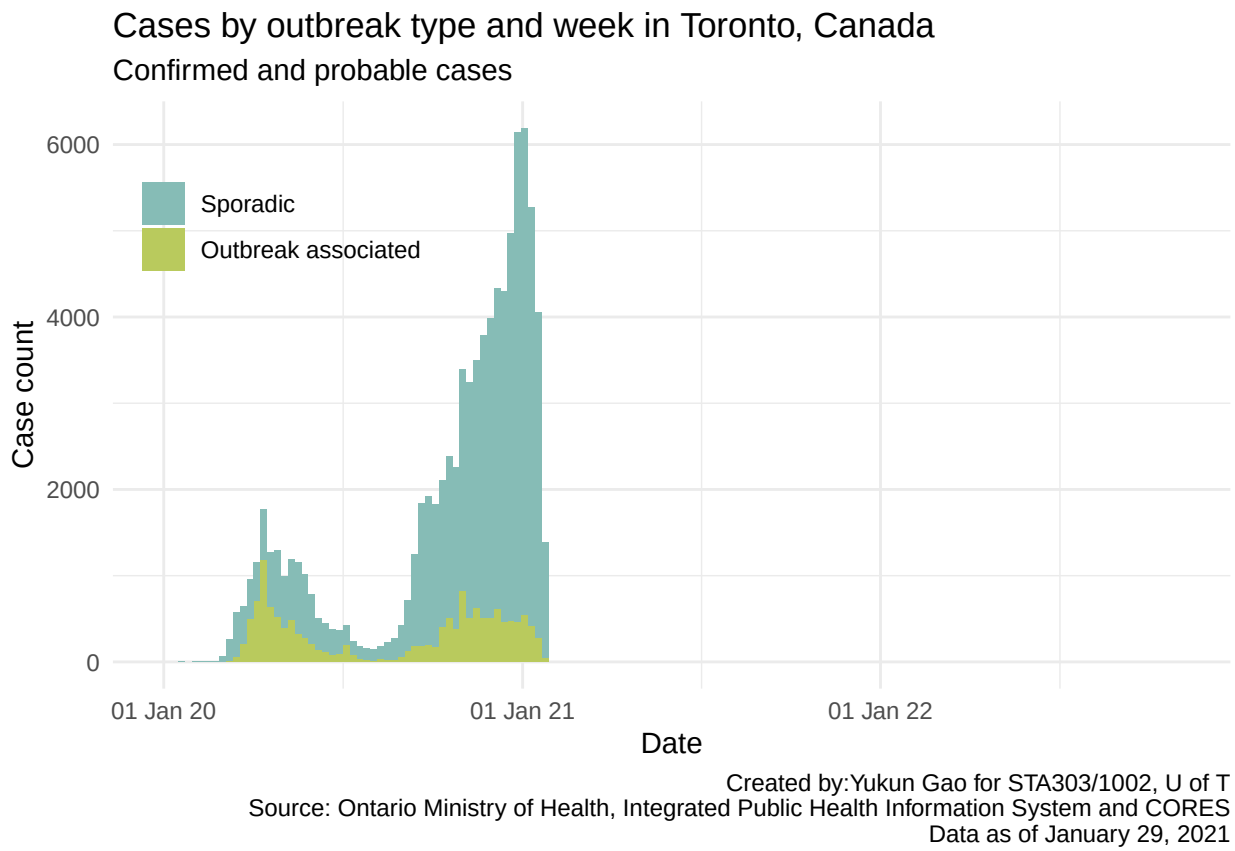
Task 2: Outbreak type

Data wrangling

```
outbreak <-outbreak_raw
outbreak$episode_week <-date(outbreak$episode_week)
outbreak$outbreak_or_sporadic <-
  str_replace_all(outbreak$outbreak_or_sporadic,
                  "OB Associated",  "Outbreak associated")

total = summarise(group_by(outbreak,episode_week),
                  total_cases=sum(cases))

outbreak1= left_join(outbreak,total) #merge them together

outbreak1$outbreak_or_sporadic =factor(outbreak1$outbreak_or_sporadic,
                                       levels = c("Sporadic","Outbreak associated"))
#make it appear in correct order
outbreak<-outbreak1
```

Data visualization

```
y_max <-max(outbreak$total_cases)
outbreak %>%
  ggplot(aes(x = episode_week, y = cases,
             fill = outbreak_or_sporadic)) +
  geom_bar(stat = "identity") +
  scale_x_date(labels = scales::date_format("%d %b %y"),
               limits= c(date("2020-01-01"),Sys.Date()+7))+
  scale_y_continuous(limits=c(0,y_max))+
  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:Yukun Gao for STA303/1002, U of T\n",
"Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES\n",
    date_daily[1,1])) +
  theme(legend.title = element_blank(),
        legend.position = c(0.15, 0.8)) + scale_fill_manual(values=c("#86BCB6","#B9CA5D"))
```



Task 3: Neighbourhoods

Data wrangling: part 1

```
income1 <-nbhood_profile%>%filter(`_id` == 1143)
income2 <-income1 %>%
  pivot_longer(-c(`_id` ,Category,Topic,`Data Source`,Characteristic),
               names_to="neighbourhood_name",values_to="percentage")%>%
  mutate(percentage=parse_number(percentage)) #transpose the data
income<-income2
```

Data wrangling: part 2

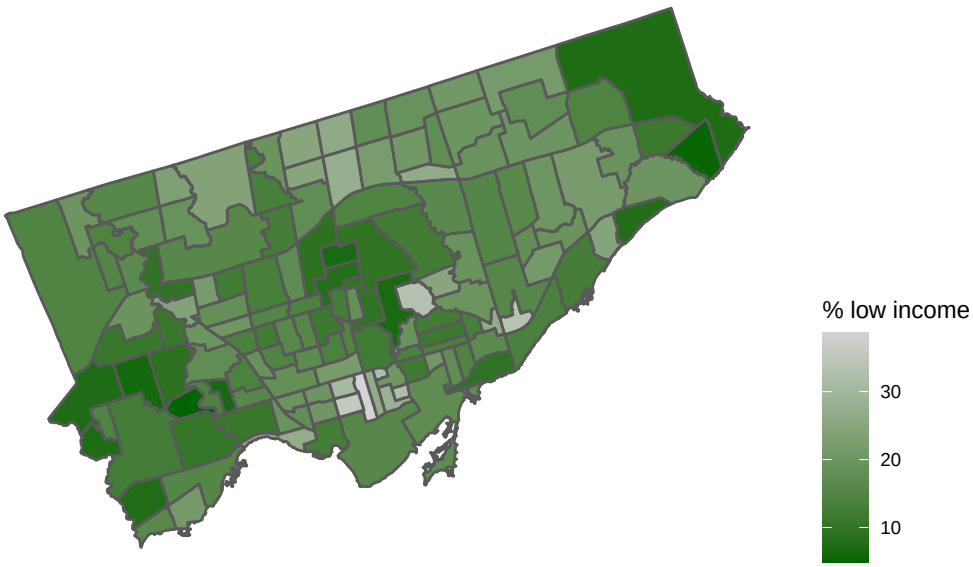
```
nbhoods_new <-
  nbhoods_shape_raw%>% mutate(neighbourhood_name=str_remove(AREA_NAME,
                                        "\\s\\(\\d+\\)$")) %>%
  mutate(neighbourhood_name=case_when(
  neighbourhood_name=="North St.James Town" ~ "North St. James Town",
  neighbourhood_name=="Weston-Pellam Park" ~ "Weston-Pelham Park",
  neighbourhood_name=="Cabbagetown-South St.James Town" ~
    "Cabbagetown-South St. James Town",TRUE ~ neighbourhood_name))
#the name is not correct so revise it
nbhoods_all1 <-left_join(nbhoods_new,income,by="neighbourhood_name")
nbhoods_all2 <-left_join(nbhoods_all1,nbhood_raw,by="neighbourhood_name")
nbhoods_all <- nbhoods_all2%>%rename(rate_per_100000=rate_per_100_000_people)
```

Data wrangling: part 3

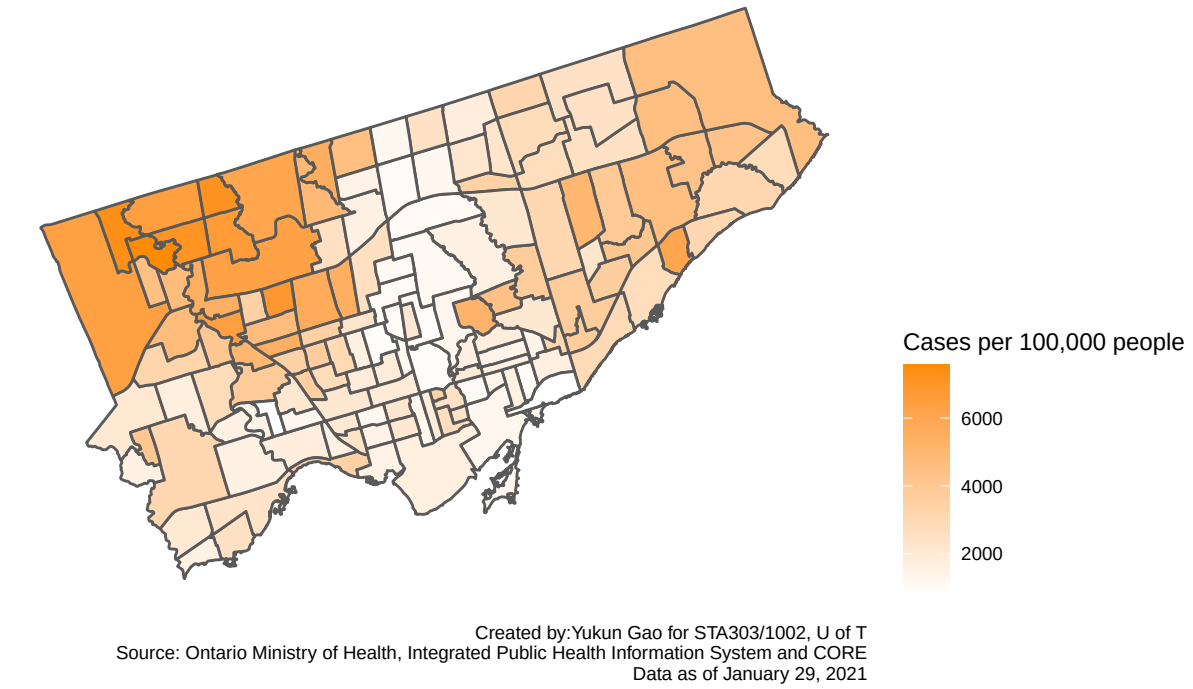
```
med_inc<-median(nbhoods_all$percentage,na.rm=TRUE)
med_rate<-median(nbhoods_all$rate_per_100000,na.rm=TRUE)
#calculate the median
nbhoods_final <- nbhoods_all%>%
  mutate(nbhood_type=case_when(
    percentage>=med_inc &rate_per_100000>=med_rate
             ~"Higher low income rate, higher case rate",
    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, lower case rate"))
```

Data visualization

```
ggplot(data = nbhoods_final)+
  geom_sf(aes(fill=percentage))+
  theme_map()+
  scale_fill_gradient(name="% low income", low = "darkgreen", high = "lightgrey")+
   theme(
        legend.position = "right")+
    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:Yukun Gao for STA303/1002, U of T\n",
          "Source: Census Profile 98-316-X2016001 via OpenData Toronto\n",
                 date_daily[1,1]))
```



```
ggplot(data = nbhoods_final)+
  geom_sf(aes(fill=rate_per_100000))+
  theme_map()+
  scale_fill_gradient(name="Cases per 100,000 people", low = "white",
                      high = "darkorange")+
    theme(legend.position = "right")+
    labs(title =
           "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
       caption = str_c(
         "Created by:Yukun Gao for STA303/1002, U of T\n",
  "Source: Ontario Ministry of Health, Integrated Public Health Information System and CORE\n",
       date_daily[1,1]))
```



```
ggplot(data = nbhoods_final)+
  geom_sf(aes(fill=nbhood_type))+
  theme_map()+ scale_fill_brewer(palette="Set1",name=
        " % of 18 to 64 year-olds in\n low income families and\n COVID-19 case rates")+
   theme(legend.position = "right")+
    labs(title = "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
        caption = str_c("Created by:Yukun Gao 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
                       Health Information System and CORES\n", date_daily[1,1]))
```