```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'googledrive' was built under R version 3.6.3
```

```
## Warning: package 'janitor' was built under R version 3.6.3
```

```
## Warning: package 'opendatatoronto' was built under R version 3.6.3
```

```
## Warning: package 'ggthemes' was built under R version 3.6.3
```

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

Task 1: Daily cases

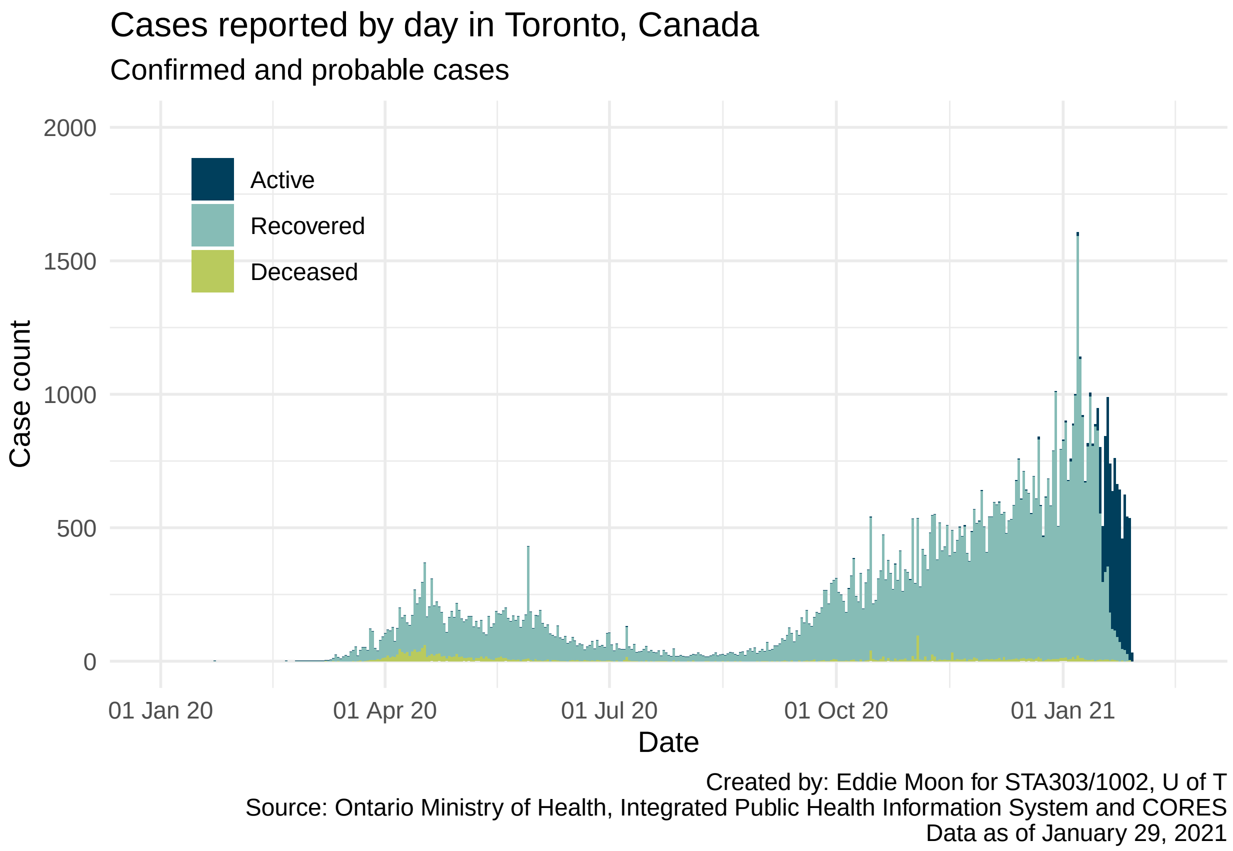
Data wrangling

```
reported <- reported_raw %>%
  mutate_if(is.numeric, replace_na, replace=0) %>%
  rename(Recovered = recovered) %>%
  rename(Active = active) %>%
  rename(Deceased = deceased) %>%
  pivot_longer(-c(reported_date), names_to = "Type", values_to = "Count") %>%
  mutate(Type = fct_relevel(Type, "Recovered", after = 1))

reported$reported_date = date(reported$reported_date)
```

Data visualization

```
reported %>%
  ggplot(aes(x=reported_date, y = Count, fill = Type)) +
  geom_bar(stat = "identity", width = 1) +
  scale_fill_manual(values = c("#003F5C", "#86BCB6", "#B9CA5D")) +
  scale_x_date(labels = scales::date_format("%d %b %y"), limits=c(date("2020-01-01"),Sys.Date())) +
  scale_y_continuous(limits = c(NA,2000)) +
  labs(title ="Cases reported by day in Toronto, Canada",
       subtitle = "Confirmed and probable cases",
       x = "Date",
       y = "Case count",
       caption = str_c("Created by: Eddie Moon for STA303/1002, U of T\nSource: Ontario Ministry of Heal
  theme_minimal() +
  theme(legend.title = element_blank(), legend.position = c(.15, .8))
```



Cases reported by day in Toronto, Canada

Confirmed and probable cases

Created by: Eddie Moon 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

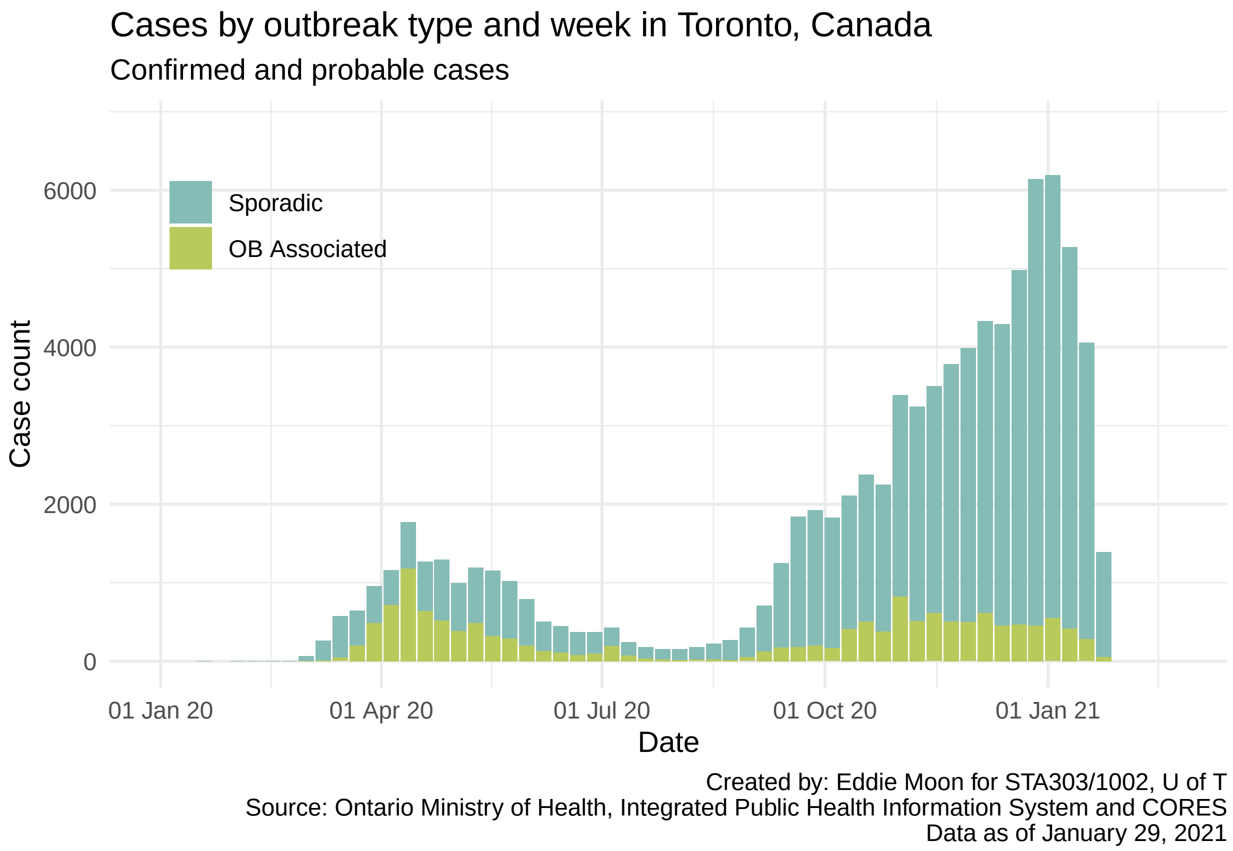
Data wrangling

```
outbreak <- outbreak_raw %>%
  rename(Type = outbreak_or_sporadic) %>%
  rename(Week = episode_week) %>%
  rename(Cases = cases) %>%
  mutate(Type = fct_relevel(Type, "OB Associated", after = 1))

outbreak$Week = date(outbreak$Week)
```

Data visualization

```
outbreak %>%
  ggplot(aes(x=Week, y = Cases, fill = Type)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("#86BCB6", "#B9CA5D")) +
  scale_x_date(labels = scales::date_format("%d %b %y"), limits=c(date("2020-01-01"),Sys.Date()+7)) +
  scale_y_continuous(limits = c(NA,6800)) +
  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: Eddie Moon for STA303/1002, U of T\nSource: Ontario Ministry of Heal
  theme_minimal() +
  theme(legend.title = element_blank(), legend.position = c(.15, .8))
```



Created by: Eddie Moon 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

```
income <- nbhood_profile %>%
  filter(‘_id‘ == 1143) %>%
  select(-c(Category, Topic, "Data Source", Characteristic)) %>%
  pivot_longer(-c(‘_id‘), names_to = "Neighbourhood", values_to = "Percentage")

income$Percentage = as.numeric(income$Percentage)
```

Data wrangling: part 2

```
nbhoods_shape_raw[19, 7] = "North St. James Town"
nbhoods_shape_raw[41, 7] = "Weston-Pelham Park"
nbhoods_shape_raw[115, 7] = "Cabbagetown-South St. James Town"

nbhoods_all <- nbhoods_shape_raw %>%
  rename(neighbourhood_name = AREA_NAME) %>%
  mutate(neighbourhood_name = str_remove(neighbourhood_name, "\\s\\(\\d+\\)$")) %>%
  right_join(income,  by = c("neighbourhood_name" = "Neighbourhood")) %>%
  left_join(nbhood_raw,  by = c("neighbourhood_name" = "neighbourhood_name")) %>%
  rename(rate_per_100000 = rate_per_100_000_people) %>%
  filter(neighbourhood_name != "City of Toronto")
```

Data wrangling: part 3

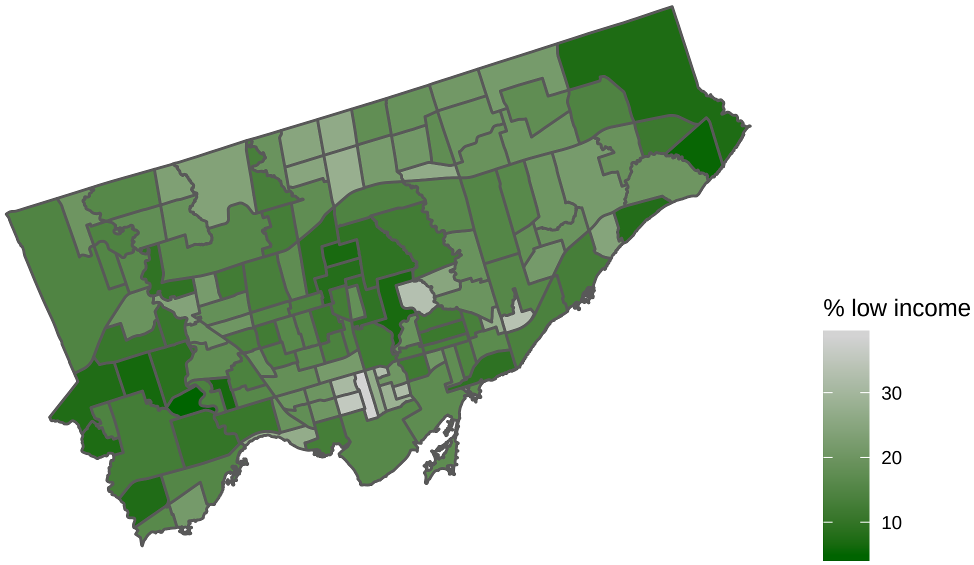
```
nbhoods_final <- nbhoods_all %>%
  mutate(med_inc = median(Percentage)) %>%
  mutate(med_rate = median(rate_per_100000)) %>%
  mutate(nbhood_type = case_when(
    (Percentage >= med_inc & rate_per_100000 >= med_rate) ~ "Higher low income rate, higher case rate",
    (Percentage >= med_inc & rate_per_100000 < 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

```r
nbhoods_final %>%
  ggplot() +
  geom_sf(aes(fill = Percentage)) +
  scale_fill_gradient(name = "% low income", low = "darkgreen", high = "lightgrey") +
  theme_map() +
  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: Eddie Moon for STA303/1002, U of T\nSource: Census Profile 98-316-X2
  theme(legend.position = "right")
```

Percentage of 18 to 64 year olds living in a low income family (2015)

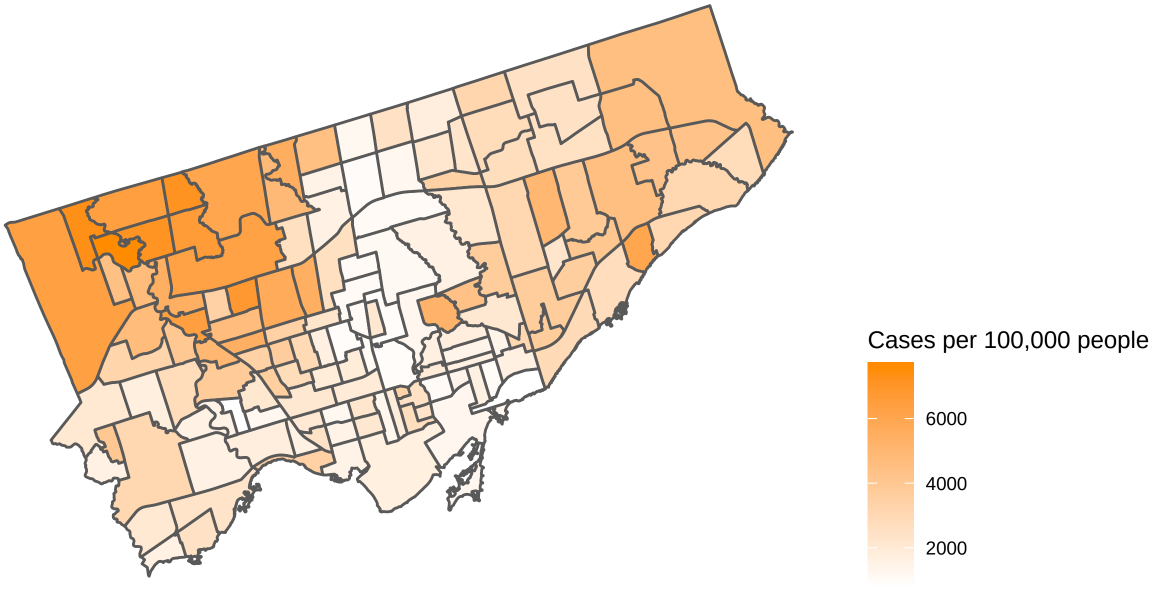
Neighbourhoods of Toronto, Canada



Created by: Eddie Moon for STA303/1002, U of T
Source: Census Profile 98−316−X2016001 via OpenData Toronto
Data as of January 28, 2021

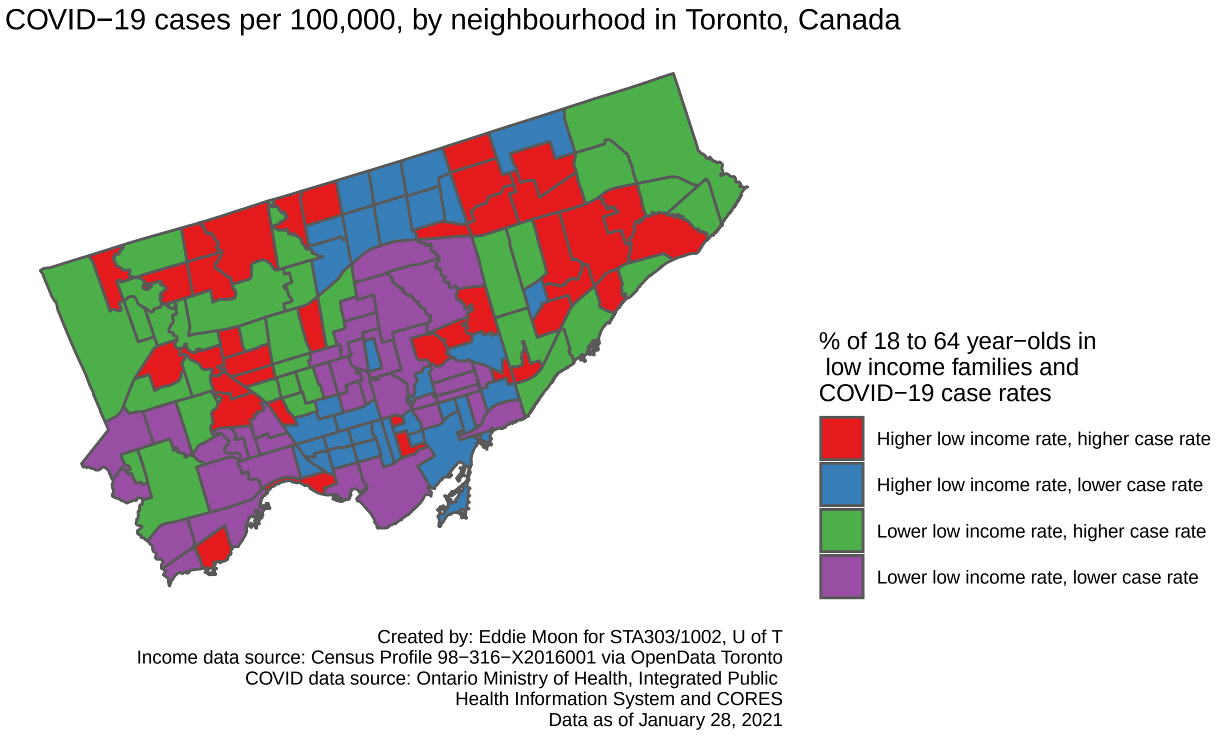
```r
nbhoods_final %>%
  ggplot() +
  geom_sf(aes(fill = rate_per_100000)) +
  scale_fill_gradient(name = "Cases per 100,000 people", low = "white", high = "darkorange") +
  theme_map() +
  labs(title = "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
       caption = str_c("Created by: Eddie Moon for STA303/1002, U of T\nSource: Ontario Ministry of Hea
  theme(legend.position = "right")
```

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



Created by: Eddie Moon for STA303/1002, U of T
Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES
Data as of January 28, 2021

```
nbhoods_final %>%
  ggplot() +
  geom_sf(aes(fill = nbhood_type)) +
  scale_fill_brewer(name = "% of 18 to 64 year-olds in\n low income families and\nCOVID-19 case rates",
                    palette = "Set1") +
  theme_map() +
  labs(title = "COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
       caption = str_c("Created by: Eddie Moon for STA303/1002, U of T\nIncome data source: Census Profi
  theme(legend.position = "right")
```



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
🟪 Lower low income rate, lower case rate

Created by: Eddie Moon 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 January 28, 2021

```
# This chunk of code helps you prepare your assessment for submission on Crowdmark
# This is optional. If it isn’t working, you can do it manually/take another approach.

# Run this chunk by hand after knitting your final version of your pdf for submission.
# A new file called ’to_submit’ will appear in your working directory with each page of your assignment

# Install the required packages
if(!match("staplr", installed.packages()[,1], nomatch = FALSE))
  {install.packages("staplr")}

# Don’t edit anything in this function
prep_for_crowdmark <- function(pdf=NULL){
  # Get the name of the file you’re currently in.
  this_file <- rstudioapi::getSourceEditorContext()$path
  pdf_name <- sub(".Rmd", ".pdf", sub(’.*/’, ’’, this_file))

  # Create a file called to_submit to put the individual files in
  # This will be in the same folder as this file is saved
  if(!match("to_submit", list.files(), nomatch = FALSE))
    {dir.create("to_submit")}

  # Split the files
  if(is.null(pdf)){
  staplr::split_pdf(pdf_name, output_directory = "to_submit", prefix = "page_")} else {
    staplr::split_pdf(pdf, output_directory = "to_submit", prefix = "page_")
  }
}

prep_for_crowdmark()
```