

Code last run 2021-02-12.

Daily: Data as of January 29, 2021.

Neighbourhood: Data as of January 28, 2021.

Task 1: Daily cases

Data wrangling

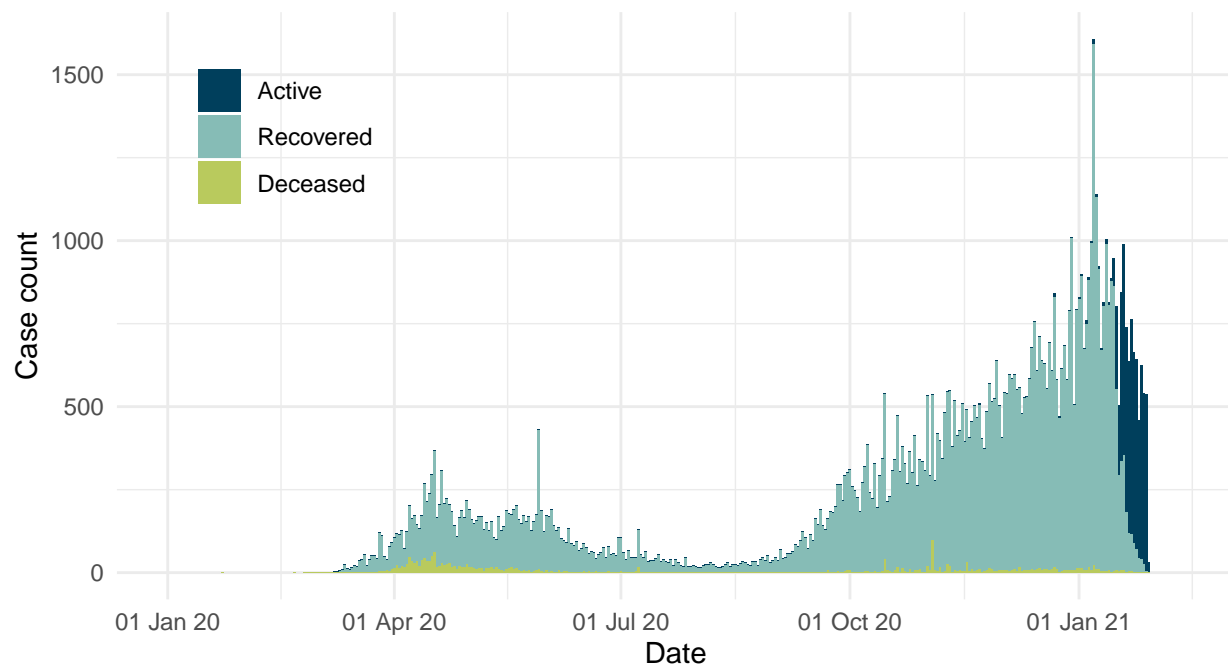
```
reported <- reported_raw %>%  
  mutate_if(is.numeric, replace_na, replace = 0) %>%  
  mutate(reported_date = date(reported_date)) %>%  
  rename(Active = active, Recovered = recovered, Deceased = deceased) %>%  
  pivot_longer(-c(reported_date), names_to = "status", values_to = "case_count") %>%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

Data visualization

```
reported %>%
  ggplot(aes(x = reported_date, y = case_count, fill = status)) +
  geom_bar(stat = "identity") +
  scale_x_date(labels = scales::date_format("%d %b %y"),
               limits = c(date("2020-01-01"), Sys.Date())) +
  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: Andy Vu 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("#003F5C", "#86BCB6", "#B9CA5D"))
```

Cases reported by day in Toronto, Canada

Confirmed and probable cases



Created by: Andy Vu 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 %>%  
  mutate(episode_week = date(episode_week)) %>%  
  mutate(outbreak_or_sporadic = str_replace_all(outbreak_or_sporadic,  
                                                "OB Associated", "Outbreak associated")) %>%  
  group_by(episode_week) %>%  
  mutate(total_cases = sum(cases)) %>%  
  mutate(outbreak_or_sporadic = fct_relevel(outbreak_or_sporadic,  
                                            "Outbreak associated", after = 1))
```

Data visualization

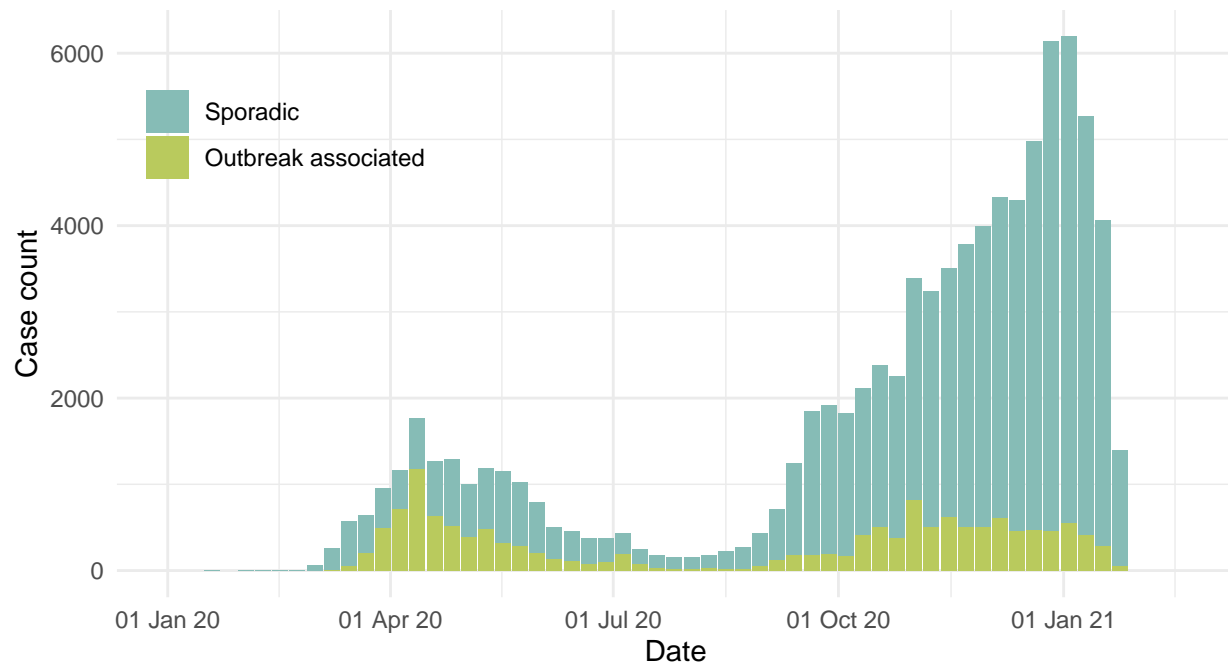
```

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, max(outbreak$total_cases))) +
  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: Andy Vu 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"))

```

Cases by outbreak type and week in Toronto, Canada

Confirmed and probable cases



Created by: Andy Vu 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) %>%
  pivot_longer(-c(`_id`, Category, Topic, `Data Source`, Characteristic,
                  `City of Toronto`), names_to = "neighbourhood_name",
              values_to = "percentage") %>%
  mutate(percentage = parse_number(percentage))
```

Data wrangling: part 2

```
nbhoods_all <- nbhoods_shape_raw %>%
  mutate(neighbourhood_name = str_remove(AREA_NAME, "\\s\\(\\d+\\)$")) %>%
  left_join(nbhood_raw, by = c("neighbourhood_name" = "neighbourhood_name")) %>%
  left_join(income, by = c("neighbourhood_name" = "neighbourhood_name")) %>%
  mutate_if(is.numeric, replace_na, replace = 0) %>%
  rename(rate_per_100000 = rate_per_100_000_people)
```

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