Code last run 2021-02-06.
Daily: Data as of January 29, 2021.
Neighbourhood: Data as of January 31, 2021.

Task 1: Daily cases

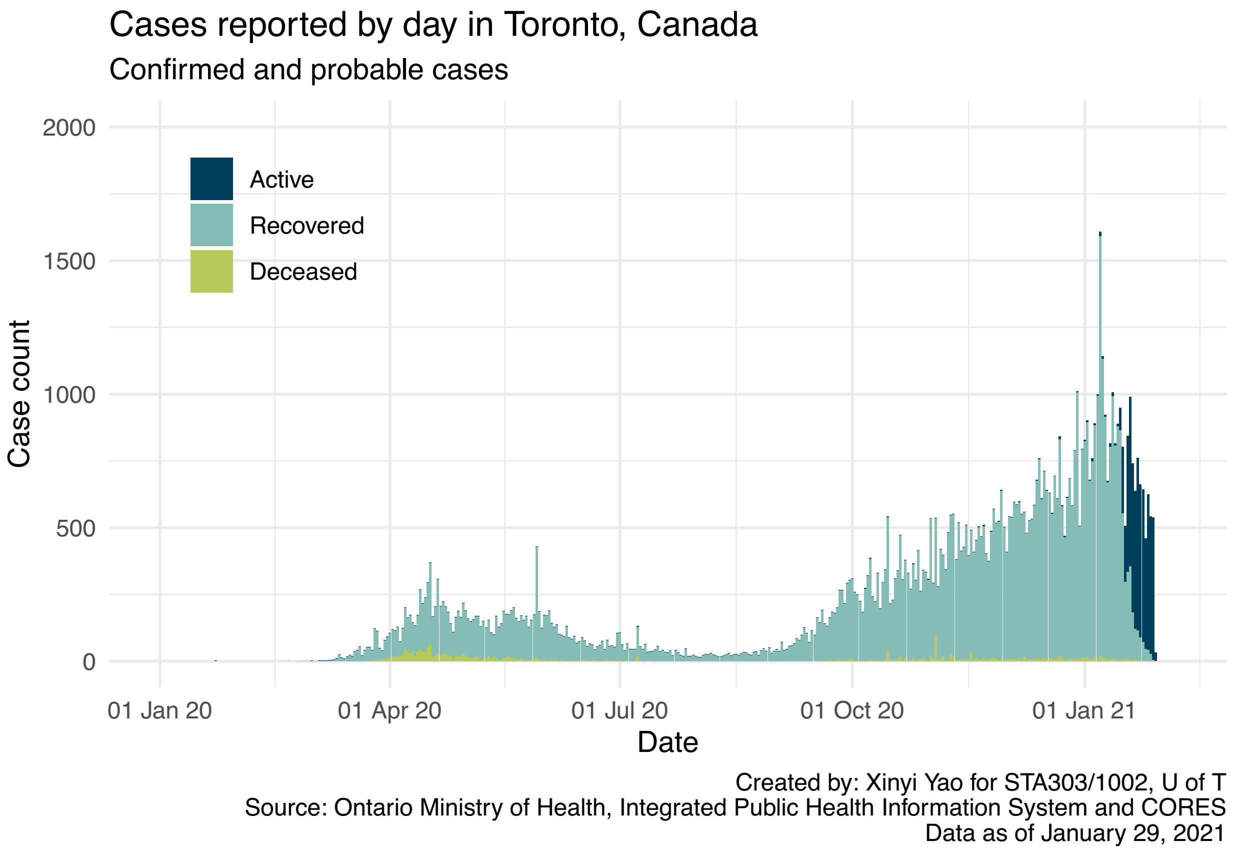
Data wrangling

```
reported<-reported_raw %>%
mutate_if(is.numeric,replace_na,replace=0) %>%
mutate(reported_date=date(reported_date)) %>%
pivot_longer(-c(reported_date),
             names_to = "case_situation",
             values_to = "number_of_persons") %>%
mutate(case_situation = str_to_sentence(case_situation)) %>%
mutate(case_situation=fct_relevel(case_situation,"Deceased",after=2))

head(reported)
```

```
## # A tibble: 6 x 3
##   reported_date case_situation number_of_persons
##   <date>        <fct>                      <dbl>
## 1 2021-01-29    Recovered                      0
## 2 2021-01-29    Active                        33
## 3 2021-01-29    Deceased                       0
## 4 2021-01-28    Recovered                      2
## 5 2021-01-28    Active                       532
## 6 2021-01-28    Deceased                       3
```

Data visualization

```
reported %>%
ggplot(aes(x=reported_date,y=number_of_persons,fill=case_situation))+
geom_bar(stat="identity")+
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),breaks=seq(0,2000,by=500)) +
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: Xinyi Yao for STA303/1002, U of T\n",
                     "Source: Ontario Ministry of Health, Integrated Public Health Information System ar
  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: Xinyi Yao 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

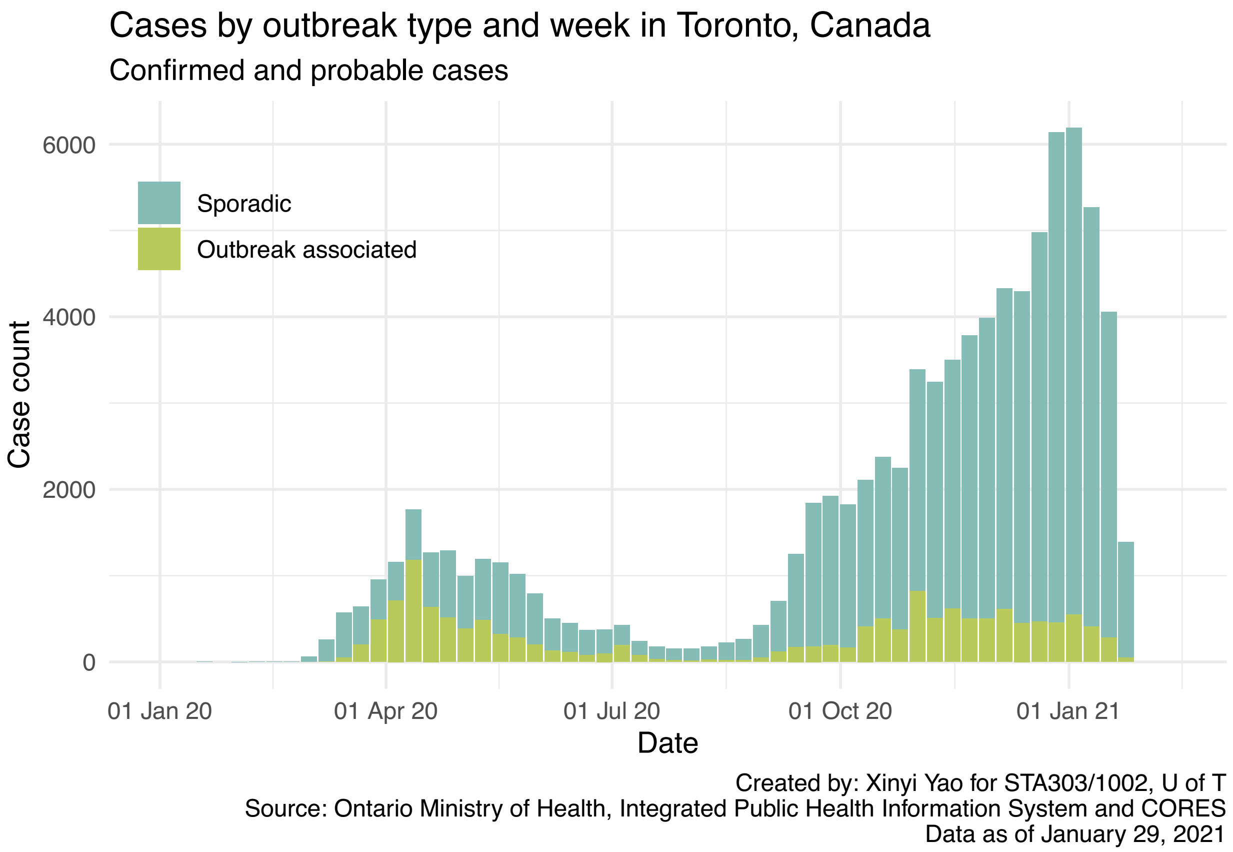
Data wrangling

```
outbreak<-outbreak_raw %>%
mutate(outbreak_or_sporadic=str_replace_all(outbreak_or_sporadic,
                                            "OB Associated",
                                            "Outbreak associated"))%>%
mutate(episode_week=date(episode_week)) %>%
mutate(outbreak_or_sporadic=fct_relevel(outbreak_or_sporadic,
                                        "Outbreak associated",
                                        after=2)) %>%
group_by(episode_week) %>%
mutate(total_cases=sum(cases))



head(outbreak)
```

```
## # A tibble: 6 x 4
## # Groups:   episode_week [3]
##   outbreak_or_sporadic episode_week cases total_cases
##   <fct>                <date>       <dbl>       <dbl>
## 1 Outbreak associated  2021-01-24      55        1391
## 2 Sporadic             2021-01-24    1336        1391
## 3 Outbreak associated  2021-01-17     286        4059
## 4 Sporadic             2021-01-17    3773        4059
## 5 Outbreak associated  2021-01-10     416        5273
## 6 Sporadic             2021-01-10    4857        5273
```

Data visualization

```
outbreak %>%
ggplot(aes(x=episode_week,y=cases,fill=outbreak_or_sporadic)) +
geom_bar(stat="identity") +
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: Xinyi Yao for STA303/1002, U of T\n",
"Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES\n",
                   date_daily[1,1])) +
  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$total_cases)),
                     breaks=seq(0,max(outbreak$total_cases),
                                by=2000)) +
  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: Xinyi Yao 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(Category=="Income") %>%
filter(Topic=="Low income in 2015") %>%
filter(Characteristic=="  18 to 64 years (%)") %>%
pivot_longer(-c(1,2,3,4,5),
             names_to="neighbourhood_name",
             values_to="percentage") %>%
mutate(percentage=parse_number(percentage)) %>%
group_by(neighbourhood_name) %>%
summarize(percentage=mean(percentage))

head(income)
```

```
## # A tibble: 6 x 2
##   neighbourhood_name           percentage
##   <chr>                             <dbl>
## 1 Agincourt North                    18.4
## 2 Agincourt South-Malvern West       21.0
## 3 Alderwood                           7.7
## 4 Annex                              20.9
## 5 Banbury-Don Mills                  13.2
## 6 Bathurst Manor                     14.2
```

Data wrangling: part 2

```
nbhoods_all<-nbhoods_shape_raw %>%
mutate(neighbourhood_name=str_remove(AREA_NAME, "\\s\\(\\d+\\)$")) %>%
mutate(neighbourhood_name=str_replace(neighbourhood_name,
                                      "North St.James Town",
                                      "North St. James Town")) %>%
mutate(neighbourhood_name=str_replace_all(neighbourhood_name,
                                          "Cabbagetown-South St.James Town",
                                          "Cabbagetown-South St. James Town"))

income<-income %>%
filter(neighbourhood_name != "City of Toronto") %>%
mutate(neighbourhood_name=str_replace(neighbourhood_name,
                                      "Weston-Pelham Park",
                                      "Weston-Pellam Park"))

nbhood_raw <- nbhood_raw %>%
mutate(neighbourhood_name=str_replace(neighbourhood_name,
                                      "Weston-Pelham Park",
                                      "Weston-Pellam Park"))



nbhoods_all<-nbhoods_all %>%
left_join(income,by="neighbourhood_name") %>%
left_join(nbhood_raw, by = "neighbourhood_name") %>%
```

```r
rename(rate_per_100000=rate_per_100_000_people)


head(nbhoods_all)
```

```
## Simple feature collection with 6 features and 22 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -79.46462 ymin: 43.66358 xmax: -79.24308 ymax: 43.8366
## geographic CRS: WGS 84
## # A tibble: 6 x 23
##   `_id` AREA_ID AREA_ATTR_ID PARENT_AREA_ID AREA_SHORT_CODE AREA_LONG_CODE
##   <int>   <int>        <int> <lgl>          <chr>           <chr>
## 1 10641 2480141     26005521 NA             096             096
## 2 10642 2480140     26005520 NA             095             095
## 3 10643 2480139     26005519 NA             109             109
## 4 10644 2480064     26005444 NA             064             064
## 5 10645 2480063     26005443 NA             103             103
## 6 10646 2480062     26005442 NA             130             130
## # ... with 17 more variables: AREA_NAME <chr>, AREA_DESC <chr>, X <lgl>,
## #   Y <lgl>, LONGITUDE <lgl>, LATITUDE <lgl>, OBJECTID <int>,
## #   Shape__Area <dbl>, Shape__Length <dbl>, geometry <POLYGON [°]>,
## #   CLASSIFICATION <chr>, CLASSIFICATION_CODE <chr>, neighbourhood_name <chr>,
## #   percentage <dbl>, neighbourhood_id <dbl>, rate_per_100000 <dbl>,
## #   case_count <dbl>
```

Data wrangling: part 3

```r
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"))




head(nbhoods_final)
```

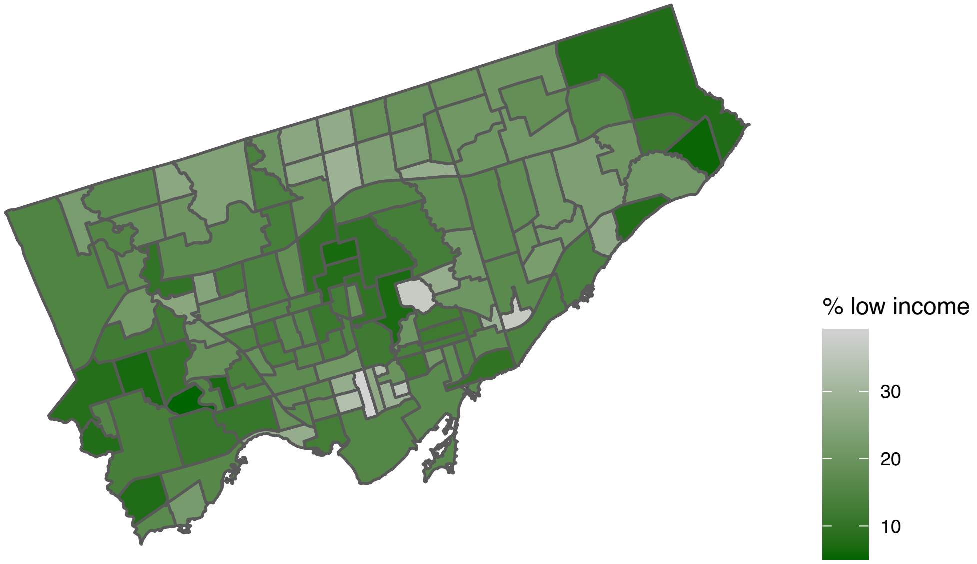
```
## Simple feature collection with 6 features and 25 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -79.46462 ymin: 43.66358 xmax: -79.24308 ymax: 43.8366
## geographic CRS: WGS 84
## # A tibble: 6 x 26
##   `_id` AREA_ID AREA_ATTR_ID PARENT_AREA_ID AREA_SHORT_CODE AREA_LONG_CODE
##   <int>   <int>        <int> <lgl>          <chr>           <chr>
## 1 10641 2480141     26005521 NA             096             096
## 2 10642 2480140     26005520 NA             095             095
## 3 10643 2480139     26005519 NA             109             109
## 4 10644 2480064     26005444 NA             064             064
## 5 10645 2480063     26005443 NA             103             103
## 6 10646 2480062     26005442 NA             130             130
## # ... with 20 more variables: AREA_NAME <chr>, AREA_DESC <chr>, X <lgl>,
## #   Y <lgl>, LONGITUDE <lgl>, LATITUDE <lgl>, OBJECTID <int>,
## #   Shape__Area <dbl>, Shape__Length <dbl>, geometry <POLYGON [°]>,
## #   CLASSIFICATION <chr>, CLASSIFICATION_CODE <chr>, neighbourhood_name <chr>,
## #   percentage <dbl>, neighbourhood_id <dbl>, rate_per_100000 <dbl>,
## #   case_count <dbl>, med_inc <dbl>, med_rate <dbl>, nbhood_type <chr>
```

Data visualization

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

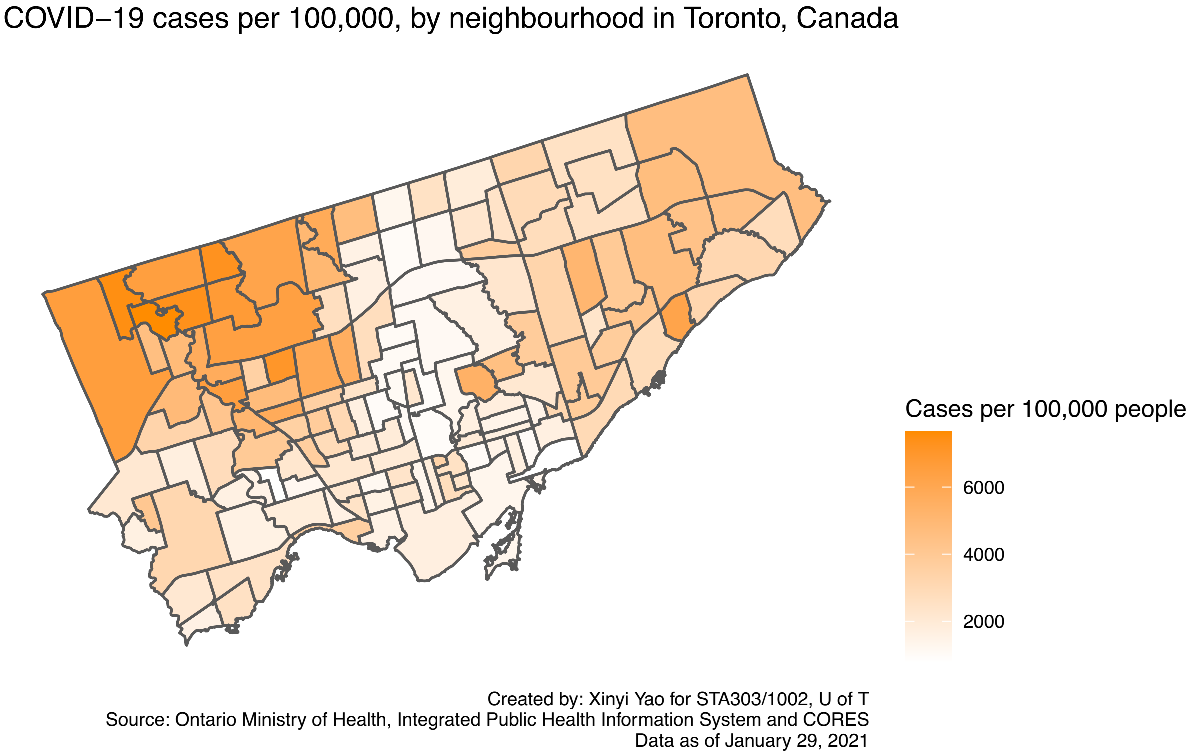
Percentage of 18 to 64 year olds living in a low income family (2015)

Neighbourhoods of Toronto, Canada



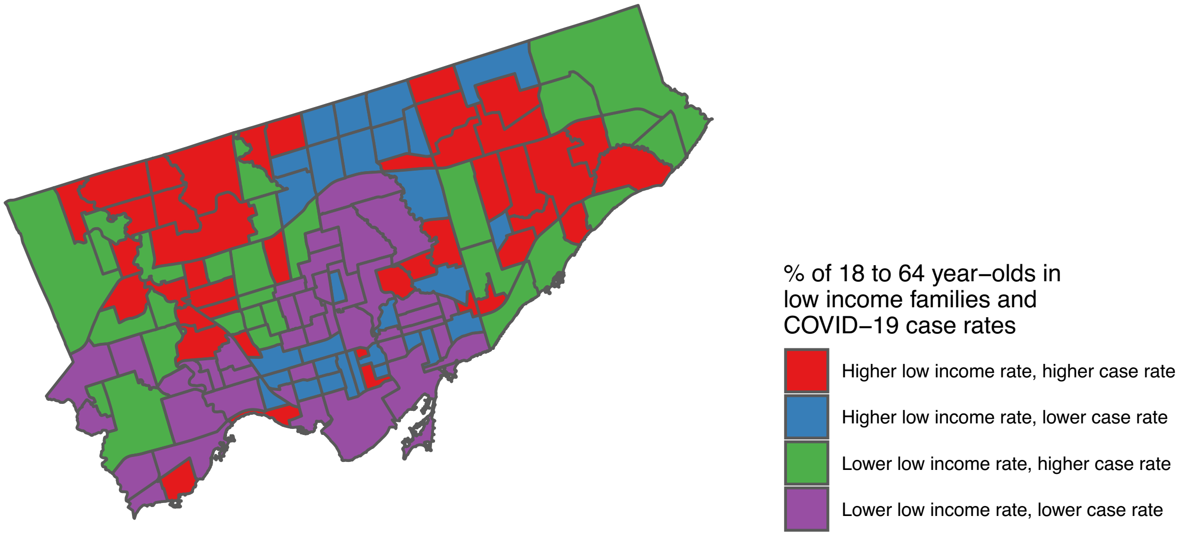
Created by: Xinyi Yao for STA303/1002, U of T
Source: Census Profile 98–316–X2016001 via OpenData Toronto
Data as of January 29, 2021

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


COVID−19 cases per 100,000, by neighbourhood in Toronto, Canada
Cases per 100,000 people
6000
4000
2000
Created by: Xinyi Yao for STA303/1002, U of T
Source: Ontario Ministry of Health, Integrated Public Health Information System and CORES
Data as of January 29, 2021

```r
ggplot(data=nbhoods_final) +
  geom_sf(data=nbhoods_final,aes(fill=nbhood_type,geometry=geometry))+
  theme_map()+
  theme(legend.position="right")+
  labs(title="COVID-19 cases per 100,000, by neighbourhood in Toronto, Canada",
       caption=str_c("Created by: Xinyi Yao for STA303/1002, U of T\n",
"Income data sourse: Census Profile 98-316-X2016001 via OpenData Toronto\n",
"COVID data sourse: Ontario Ministry of Health, Integrated Public\n",
"Health Information System and CORES\n",
                     date_daily[1,1])) +
  scale_fill_brewer(palette="Set1",
  name="% of 18 to 64 year-olds in\nlow income families and \nCOVID-19 case rates")
```

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: Xinyi Yao for STA303/1002, U of T
Income data sourse: Census Profile 98–316–X2016001 via OpenData Toronto
COVID data sourse: Ontario Ministry of Health, Integrated Public
Health Information System and CORES
Data as of January 29, 2021

```r
# 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()
```