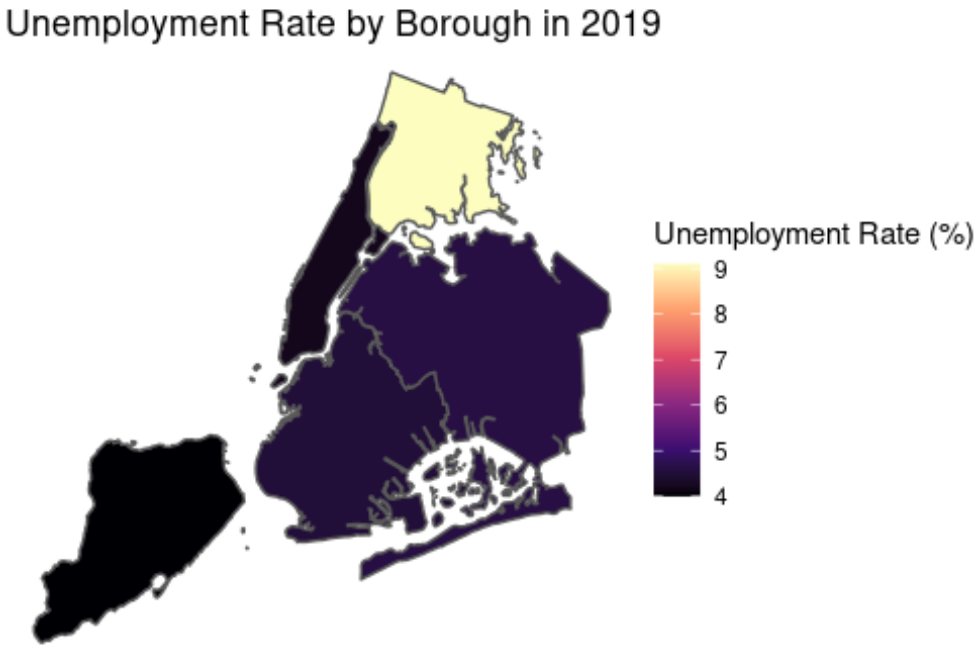
```
library(lehmansociology)

##
## Attaching package: 'lehmansociology'

## The following object is masked from 'package:stats':
##
##     frequency

library(sf)

## Linking to GEOS 3.4.2, GDAL 2.4.2, PROJ 4.8.0

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(readr)
library(nycgeo)

acs_data2 <- read_csv("acs_data2.csv", skip = 1)

##
## ── Column specification
──────────────────────────────────────────────────────────

## cols(
##   .default = col_double(),
##   id = col_character(),
##   `Geographic Area Name` = col_character(),
##   `Percent Margin of Error!!EMPLOYMENT STATUS!!Population 16 years and
over` = col_character(),
##   `Percent Margin of Error!!EMPLOYMENT STATUS!!Civilian labor force` =
col_character(),
##   `Estimate!!EMPLOYMENT STATUS!!Civilian labor force!!Unemployment Rate` =
col_character(),
##   `Margin of Error!!EMPLOYMENT STATUS!!Civilian labor force!!Unemployment
Rate` = col_character(),
##   `Percent Margin of Error!!EMPLOYMENT STATUS!!Females 16 years and over`
= col_character(),
##   `Percent Margin of Error!!EMPLOYMENT STATUS!!Own children of the
householder under 6 years` = col_character(),
##   `Percent Margin of Error!!EMPLOYMENT STATUS!!Own children of the
```

```
householder 6 to 17 years` = col_character(),
##   `Percent Margin of Error!!COMMUTING TO WORK!!Workers 16 years and over`
= col_character(),
##   `Percent!!COMMUTING TO WORK!!Workers 16 years and over!!Mean travel time
to work (minutes)` = col_character(),
##   `Percent Margin of Error!!COMMUTING TO WORK!!Workers 16 years and
over!!Mean travel time to work (minutes)` = col_character(),
##   `Percent Margin of Error!!OCCUPATION!!Civilian employed population 16
years and over` = col_character(),
##   `Percent Margin of Error!!INDUSTRY!!Civilian employed population 16
years and over` = col_character(),
##   `Percent Margin of Error!!CLASS OF WORKER!!Civilian employed population
16 years and over` = col_character(),
##   `Percent Margin of Error!!INCOME AND BENEFITS (IN 2019 INFLATION-
ADJUSTED DOLLARS)!!Total households` = col_character(),
##   `Percent!!INCOME AND BENEFITS (IN 2019 INFLATION-ADJUSTED
DOLLARS)!!Total households!!Median household income (dollars)` =
col_character(),
##   `Percent Margin of Error!!INCOME AND BENEFITS (IN 2019 INFLATION-
ADJUSTED DOLLARS)!!Total households!!Median household income (dollars)` =
col_character(),
##   `Percent!!INCOME AND BENEFITS (IN 2019 INFLATION-ADJUSTED
DOLLARS)!!Total households!!Mean household income (dollars)` =
col_character(),
##   `Percent Margin of Error!!INCOME AND BENEFITS (IN 2019 INFLATION-
ADJUSTED DOLLARS)!!Total households!!Mean household income (dollars)` =
col_character()
##   # ... with 73 more columns
## )
## ℹ Use `spec()` for the full column specifications.

#View(acs_data2)

nyc_boundaries(geography = "borough")

## Simple feature collection with 5 features and 6 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 913213.6 ymin: 120131.4 xmax: 1067379 ymax: 272798.5
## CRS:            EPSG:2263
## # A tibble: 5 x 7
##   geoid state_fips county_fips county_name borough_name  borough_id
##   <chr> <chr>      <chr>       <chr>       <chr>         <chr>
## 1 36061 36         061         New York    Manhattan     1
## 2 36005 36         005         Bronx       Bronx         2
## 3 36047 36         047         Kings       Brooklyn      3
## 4 36081 36         081         Queens      Queens        4
## 5 36085 36         085         Richmond    Staten Island 5
## # … with 1 more variable: geometry <MULTIPOLYGON [US_survey_foot]>
```

```r
acs_nyc <- acs_data2 %>% select("id", "Geographic Area Name",
            "Percent!!EMPLOYMENT STATUS!!Civilian labor force!!Unemployment
Rate",
            "Estimate!!INCOME AND BENEFITS (IN 2019 INFLATION-ADJUSTED
DOLLARS)!!Total households!!Median household income (dollars)"
            )

names(acs_nyc)[names(acs_nyc)== "id"] <- "ID"
names(acs_nyc)[names(acs_nyc)== "Geographic Area Name"] <- "area"
names(acs_nyc)[names(acs_nyc)== "Percent!!EMPLOYMENT STATUS!!Civilian labor
force!!Unemployment Rate"] <- "unemployment_rate"
names(acs_nyc)[names(acs_nyc)== "Estimate!!INCOME AND BENEFITS (IN 2019
INFLATION-ADJUSTED DOLLARS)!!Total households!!Median household income
(dollars)"] <- "med_income"

# FOR NYC AS A WHOLE: unemployment = 4.4%, median income = $72,108
acs_nyc <-subset(acs_nyc, ID!= "0400000US36")

acs_nyc$area[acs_nyc$area == "Bronx County, New York"] <- "Bronx"
acs_nyc$area[acs_nyc$area == "Kings County, New York"] <- "Brooklyn"
acs_nyc$area[acs_nyc$area == "New York County, New York"] <- "Manhattan"
acs_nyc$area[acs_nyc$area == "Queens County, New York"] <- "Queens"
acs_nyc$area[acs_nyc$area == "Richmond County, New York"] <- "Staten Island"

test_map <- nyc_boundaries(
  geography = "borough",
  filter_by = "borough",
  region = c("brooklyn", "queens", "bronx", "staten island", "manhattan")
  )
ggplot(test_map) +
  geom_sf(aes(fill = )) +
  theme_minimal()
```


```
nyc_boundaries(geography = "borough") %>%
  left_join(acs_nyc, by = c("borough_name" = "area")) %>%
  ggplot() +
  geom_sf(aes(fill = unemployment_rate)) +
  scale_fill_viridis_c(name = "Unemployment Rate (%)", option = "magma") +
  theme_void() +
  theme(panel.grid = element_line(color = "transparent")) +
  labs(title = "Unemployment Rate by Borough in 2019")
```




```r
nyc_boundaries(geography = "borough") %>%
  left_join(acs_nyc, by = c("borough_name" = "area")) %>%
  ggplot() +
  geom_sf(aes(fill = med_income)) +
  scale_fill_viridis_c(name = "Median Income ($)", option = "magma") +
  theme_void() +
  theme(panel.grid = element_line(color = "transparent")) +
  labs(title = "Median Income by Borough in 2019")
```

Median Income by Borough in 2019