GES488 Final Project: Baltimore County

Kristian Nelson

4/11/2018

```
#install.packages("tidycensus")
#install.packages("tidyverse")
#install.packages("tigris")
#install.packages("sf")
library(tidycensus)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────────────────────
tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 2.2.1     ✔ purrr   0.2.4
## ✔ tibble  1.4.2     ✔ dplyr   0.7.4
## ✔ tidyr   0.8.0     ✔ stringr 1.2.0
## ✔ readr   1.1.1     ✔ forcats 0.2.0
```

```
## ── Conflicts ────────────────────────────────────────────────────────────── tidyv
erse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(tigris)
```

```
## To enable
## caching of data, set `options(tigris_use_cache = TRUE)` in your R script or .Rprofil
e.
```

```
##
## Attaching package: 'tigris'
```

```
## The following object is masked from 'package:graphics':
##
##     plot
```

```
library(sf)
```

```
## Linking to GEOS 3.6.1, GDAL 2.1.3, proj.4 4.9.3
```

```
library(RColorBrewer)
options(tigris_class = "sf")
options(tigris_use_cache = TRUE)
census_api_key("02fd95ffa4b152f4183ec87b9bc382caf029e468")
```

```
## To install your API key for use in future sessions, run this function with `install =
TRUE`.
```

Before knitting document, install all packages!

Part 1:

This is where I will get the data for Baltimore City. In the get.acs function I specificed Baltimore City as the county so it would only retrieve the data for that area.

```
fd <- get_acs(geography = "tract",
                variables = c("B08301_019","B08301_001","B19301_001"),
                state = c("MD"), county = "Baltimore County", geometry = TRUE, output =
 "wide")
```

```
## Getting data from the 2012-2016 5-year ACS
```

```
metros <- core_based_statistical_areas(cb = TRUE) %>%
  filter(GEOID %in% c("12580")) %>%
  select(metro_name = NAME)

balcit <- st_join(fd, metros, join = st_within, left = FALSE)
```

```
## although coordinates are longitude/latitude, st_within assumes that they are planar
```

```
head(balcit)
```

```
## Simple feature collection with 6 features and 9 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -76.82131 ymin: 39.26233 xmax: -76.45567 ymax: 39.47489
## epsg (SRID):    4269
## proj4string:    +proj=longlat +datum=NAD83 +no_defs
##         GEOID                                              NAME B08301_019E
## 1 24005400100    Census Tract 4001, Baltimore County, Maryland          29
## 2 24005401507 Census Tract 4015.07, Baltimore County, Maryland         151
## 3 24005402505 Census Tract 4025.05, Baltimore County, Maryland           0
## 4 24005404202 Census Tract 4042.02, Baltimore County, Maryland         487
## 5 24005408503 Census Tract 4085.03, Baltimore County, Maryland         116
## 6 24005411303 Census Tract 4113.03, Baltimore County, Maryland          21
##   B08301_019M B08301_001E B08301_001M B19301_001E B19301_001M
## 1          33        1125         180       33298        3483
## 2         228        2724         319       22887        3451
## 3          12        1603         215       26078        3867
## 4         286        4766         448       20746        2079
## 5          83        2042         244       31223        5416
## 6          30        3011         385       39039        5397
##                      metro_name                       geometry
## 1 Baltimore-Columbia-Towson, MD MULTIPOLYGON (((-76.71872 3...
## 2 Baltimore-Columbia-Towson, MD MULTIPOLYGON (((-76.7702 39...
## 3 Baltimore-Columbia-Towson, MD MULTIPOLYGON (((-76.81871 3...
## 4 Baltimore-Columbia-Towson, MD MULTIPOLYGON (((-76.79755 3...
## 5 Baltimore-Columbia-Towson, MD MULTIPOLYGON (((-76.64111 3...
## 6 Baltimore-Columbia-Towson, MD MULTIPOLYGON (((-76.48836 3...
```

Since the tenure only gives us total number of households, total number of owners, and total number of renters, I divided the amount renters by the total amount of households to get the percentage.

```
balcit$walking_rate <- balcit$B08301_019E / balcit$B08301_001E
```

```
balcit.lm <- lm(balcit$B19301_001E ~ balcit$walking_rate)
```

Part 2:

```
balcit_comp <- balcit %>% filter(walking_rate >= 0 & B19301_001E >= 0)
```

```
#install.packages("spdep")
library(spdep)
```

```
## Loading required package: sp
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
## Loading required package: spData
```

```
library(sf)
c <- poly2nb(as(balcit_comp, "Spatial"), row.names=balcit_comp$GEOID)
ww <- nb2listw(c)
balcit_comp.moran <- moran.test(balcit_comp$B19301_001E, listw=ww, randomisation=FALSE)
balcit_comp.moran
```

```
##
##  Moran I test under normality
##
## data:  balcit_comp$B19301_001E
## weights: ww
##
## Moran I statistic standard deviate = 13.713, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.573580349      -0.004761905       0.001778837
```

```
balcit_comp.z <- (balcit_comp.moran$estimate[1] - balcit_comp.moran$estimate[2]) /
                           balcit_comp.moran$estimate[3]
summary(balcit_comp.z)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   325.1   325.1   325.1   325.1   325.1   325.1
```

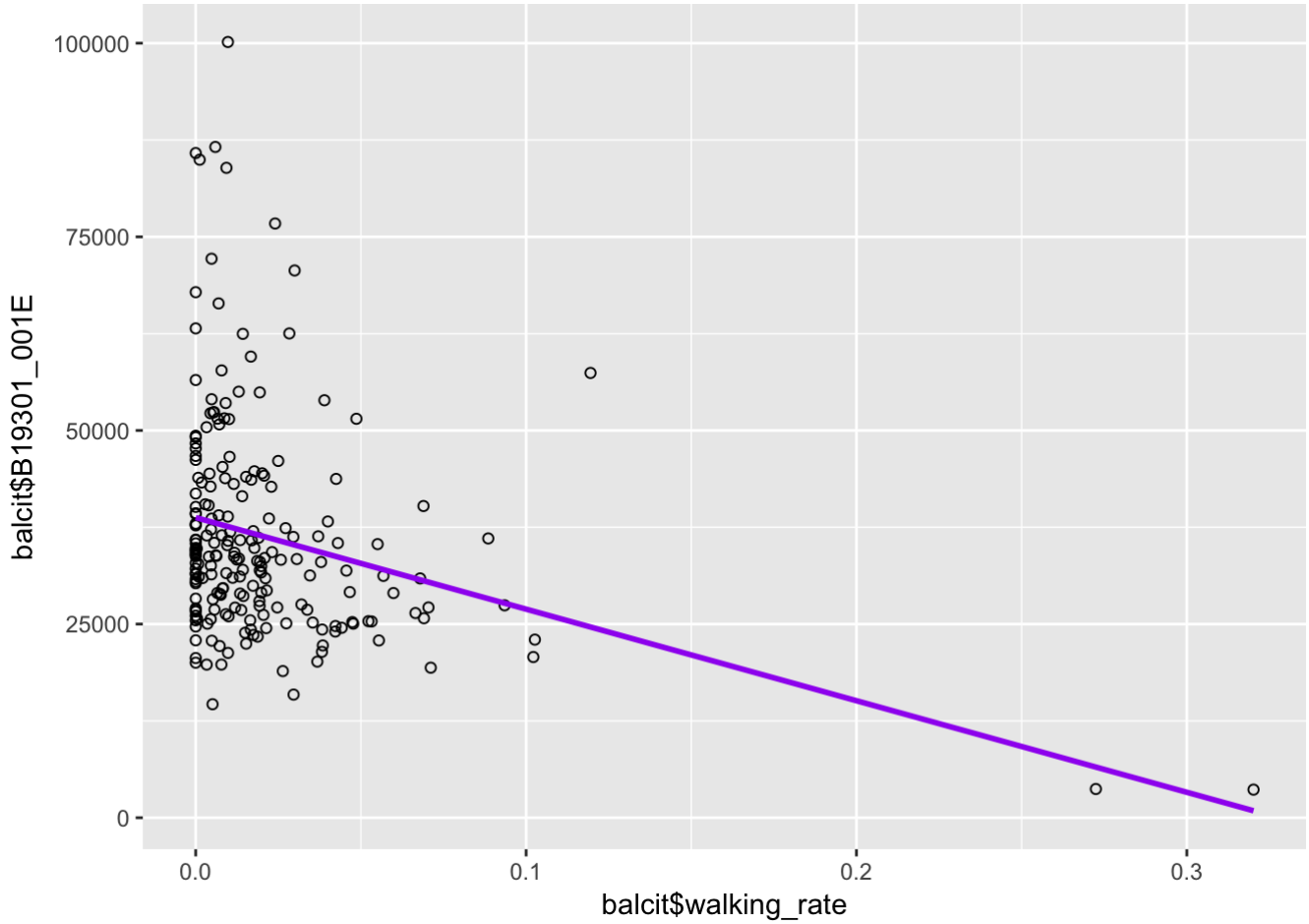
From this code I learned that household median income and the percentage of renters are spatially autocorrelated. Using Moran's I it was found that there was some levels of clustering between the two sets of data. The Moran's I statistic was 0.533 which is closer to 1 than it is to 0 which indicates clustering. The p value < 2.2e-16 which is less than 0.05 which shows this statistic is significant. The z-score was then calculated to find to what extent the clustering was. The z-score came out to be 311 which is a positive score that means there is a significant amount of clustering. All of these values show that these datasets are spatially autocorrelated.

Part 3:

```
balcit.lm <- lm(balcit$B19301_001E ~ balcit$walking_rate)
ggplot(balcit, aes(x = balcit$walking_rate , y = balcit$B19301_001E)) +
  geom_point(shape = 1) +
  geom_smooth(method = "lm", se = FALSE, color = "purple")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

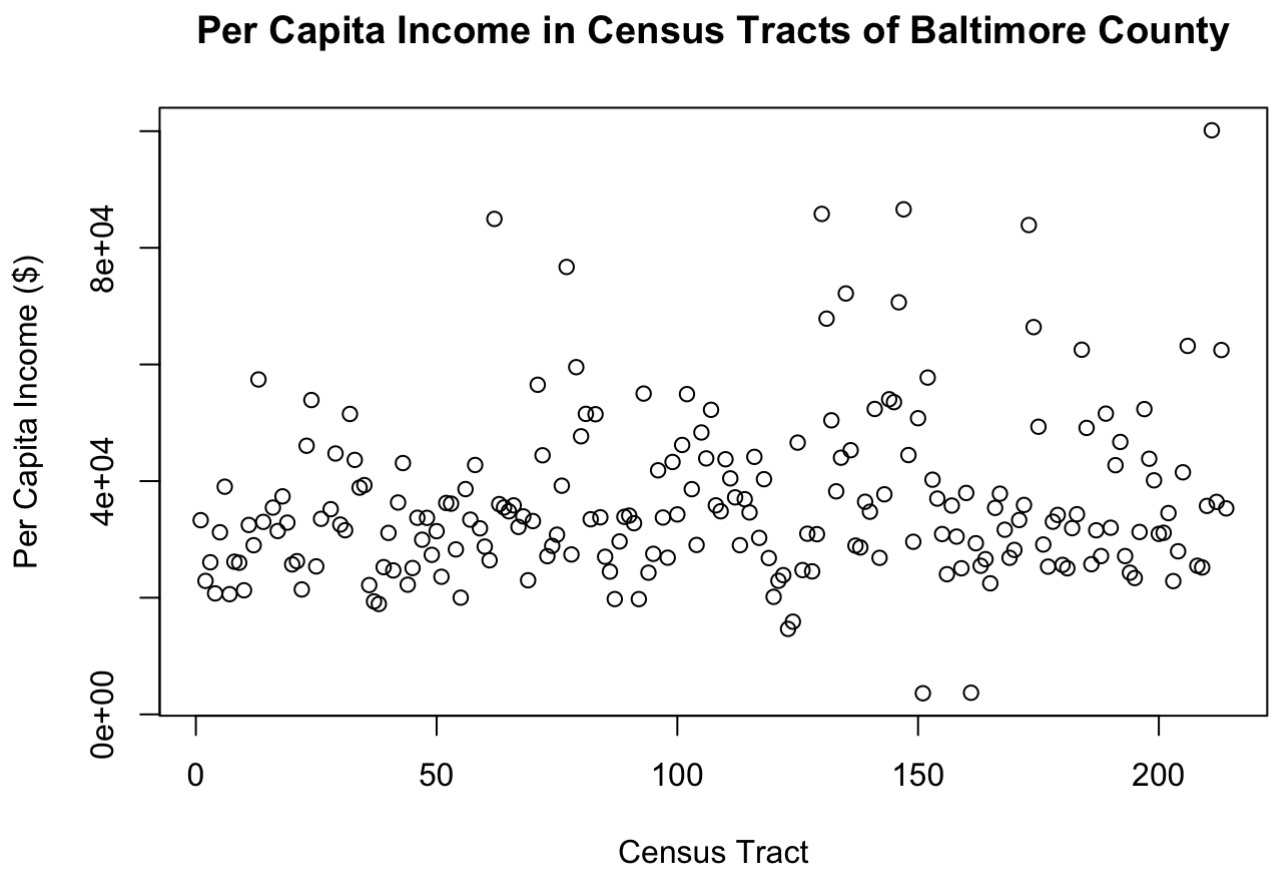
```
## Warning: Removed 3 rows containing missing values (geom_point).
```



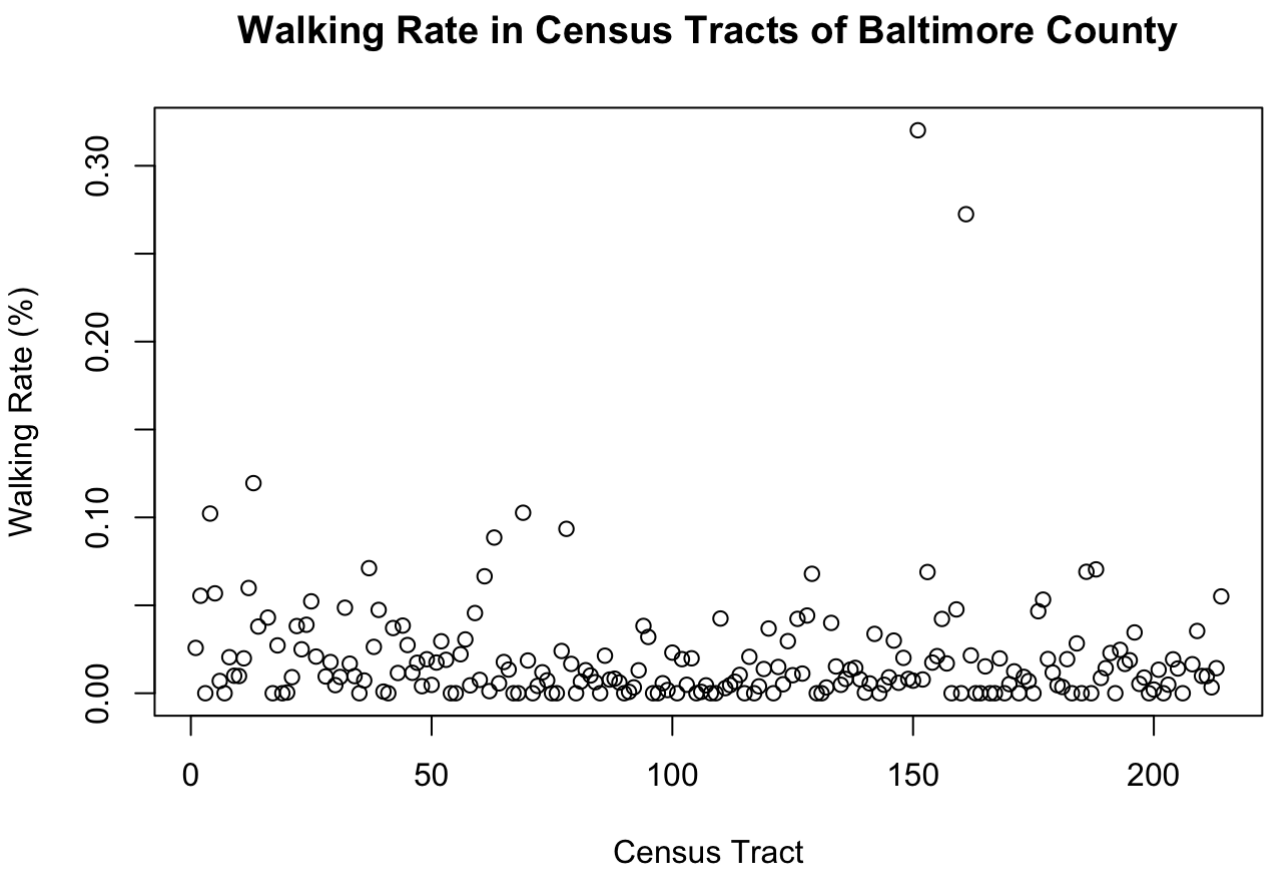
The graph shown above shows that theses two datasets are in fact correlated. It shows that as household income decreases, the rental percentage increases. This means that in the Baltimore City area, families with a lower income are forced to rent the places they live at instead of purchasing a house. Families with higher income do not need to rent.

Part 4:

```
plot(balcit$B19301_001E, main="Per Capita Income in Census Tracts of Baltimore County",
    xlab="Census Tract", ylab="Per Capita Income ($)")
```




```
plot(balcit$walking_rate, main="Walking Rate in Census Tracts of Baltimore County",
    xlab="Census Tract", ylab="Walking Rate (%)")
```



Part 5:

```
library(broom)
balcit_comp.sp <- as(balcit_comp, 'Spatial')
balcit_df <- tidy(balcit_comp.sp)
```

```
## Regions defined for each Polygons
```

```
head(balcit_df)
```

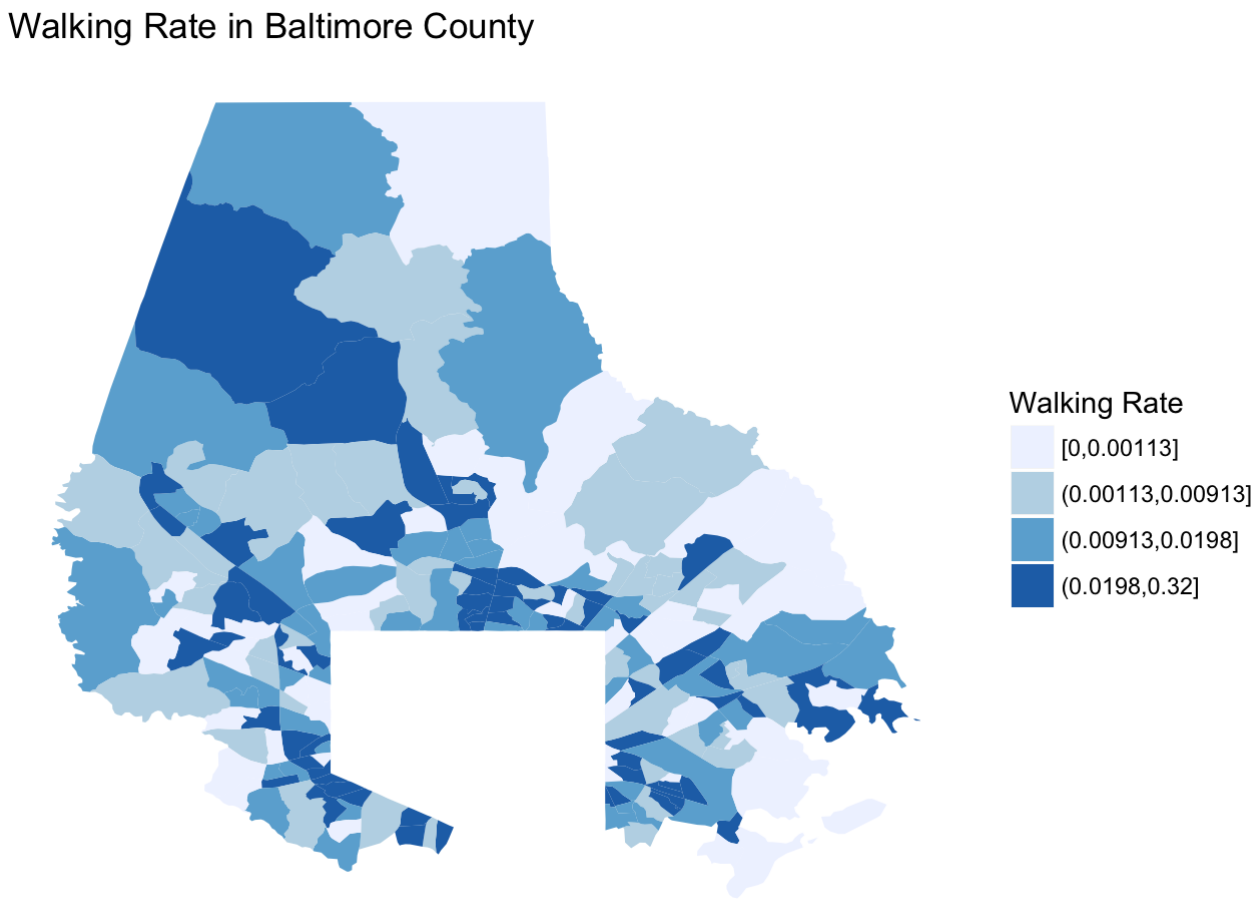
```
##        long      lat order  hole piece group id
## 1 -76.71872 39.27112     1 FALSE     1   1.1  1
## 2 -76.71391 39.27027     2 FALSE     1   1.1  1
## 3 -76.71133 39.27182     3 FALSE     1   1.1  1
## 4 -76.70987 39.27251     4 FALSE     1   1.1  1
## 5 -76.70467 39.27182     5 FALSE     1   1.1  1
## 6 -76.69889 39.27270     6 FALSE     1   1.1  1
```

```
balcit_comp.sp$polyID <- sapply(slot(balcit_comp.sp, "polygons"), function(x) slot(x, "I
D"))
balcit_df <- merge(balcit_df, balcit_comp.sp, by.x = "id", by.y="polyID")
head(balcit_df)
```

```
##   id      long      lat order  hole piece group       GEOID
## 1  1 -76.71872 39.27112     1 FALSE     1   1.1 24005400100
## 2  1 -76.71391 39.27027     2 FALSE     1   1.1 24005400100
## 3  1 -76.71133 39.27182     3 FALSE     1   1.1 24005400100
## 4  1 -76.70987 39.27251     4 FALSE     1   1.1 24005400100
## 5  1 -76.70467 39.27182     5 FALSE     1   1.1 24005400100
## 6  1 -76.69889 39.27270     6 FALSE     1   1.1 24005400100
##                                           NAME B08301_019E B08301_019M
## 1 Census Tract 4001, Baltimore County, Maryland          29          33
## 2 Census Tract 4001, Baltimore County, Maryland          29          33
## 3 Census Tract 4001, Baltimore County, Maryland          29          33
## 4 Census Tract 4001, Baltimore County, Maryland          29          33
## 5 Census Tract 4001, Baltimore County, Maryland          29          33
## 6 Census Tract 4001, Baltimore County, Maryland          29          33
##   B08301_001E B08301_001M B19301_001E B19301_001M
## 1        1125         180       33298        3483
## 2        1125         180       33298        3483
## 3        1125         180       33298        3483
## 4        1125         180       33298        3483
## 5        1125         180       33298        3483
## 6        1125         180       33298        3483
##                       metro_name walking_rate
## 1 Baltimore-Columbia-Towson, MD   0.02577778
## 2 Baltimore-Columbia-Towson, MD   0.02577778
## 3 Baltimore-Columbia-Towson, MD   0.02577778
## 4 Baltimore-Columbia-Towson, MD   0.02577778
## 5 Baltimore-Columbia-Towson, MD   0.02577778
## 6 Baltimore-Columbia-Towson, MD   0.02577778
```

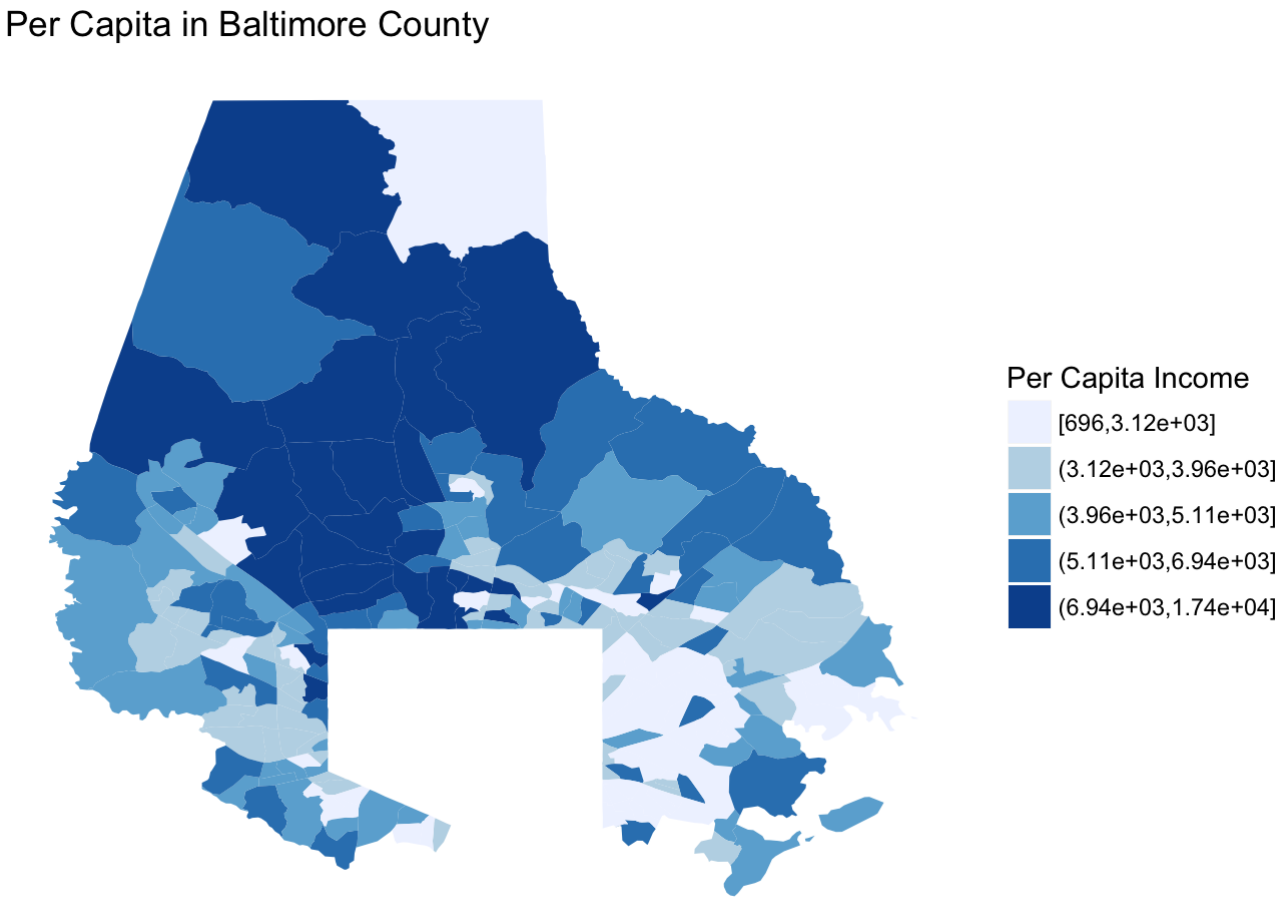
Below is the map for Rental Rate in Baltimore City.

```
library(ggplot2)

ggplot() +
  geom_polygon(
    data = balcit_df,
    aes(x = long, y = lat, group = group,
        fill = cut_number(walking_rate, 4))) +
  scale_fill_brewer("Walking Rate", palette = "Blues") +
  ggtitle("Walking Rate in Baltimore County") +
  theme(line = element_blank(),
        axis.text=element_blank(),
        axis.title=element_blank(),
        panel.background = element_blank()) +
  coord_equal()
```



Below is the map for Median Household Income in Baltimore City.

```
library(ggplot2)

ggplot() +
  geom_polygon(
    data = balcit_df,
    aes(x = long, y = lat, group = group,
        fill = cut_number(B19301_001M, 5))) +
  scale_fill_brewer("Per Capita Income", palette = "Blues") +
  ggtitle("Per Capita in Baltimore County") +
  theme(line = element_blank(),
        axis.text=element_blank(),
        axis.title=element_blank(),
        panel.background = element_blank()) +
  coord_equal()
```



Part 6

```
#install.packages("ggmap")
library(ggmap)
myLocation <- myLocation <- c(lon = -76.61, lat = 39.29)
geocode("Maryland")
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Maryl
and&sensor=false
```

```
## Warning: geocode failed with status OVER_QUERY_LIMIT, location = "Maryland"
```

```
##   lon lat
## 1  NA  NA
```

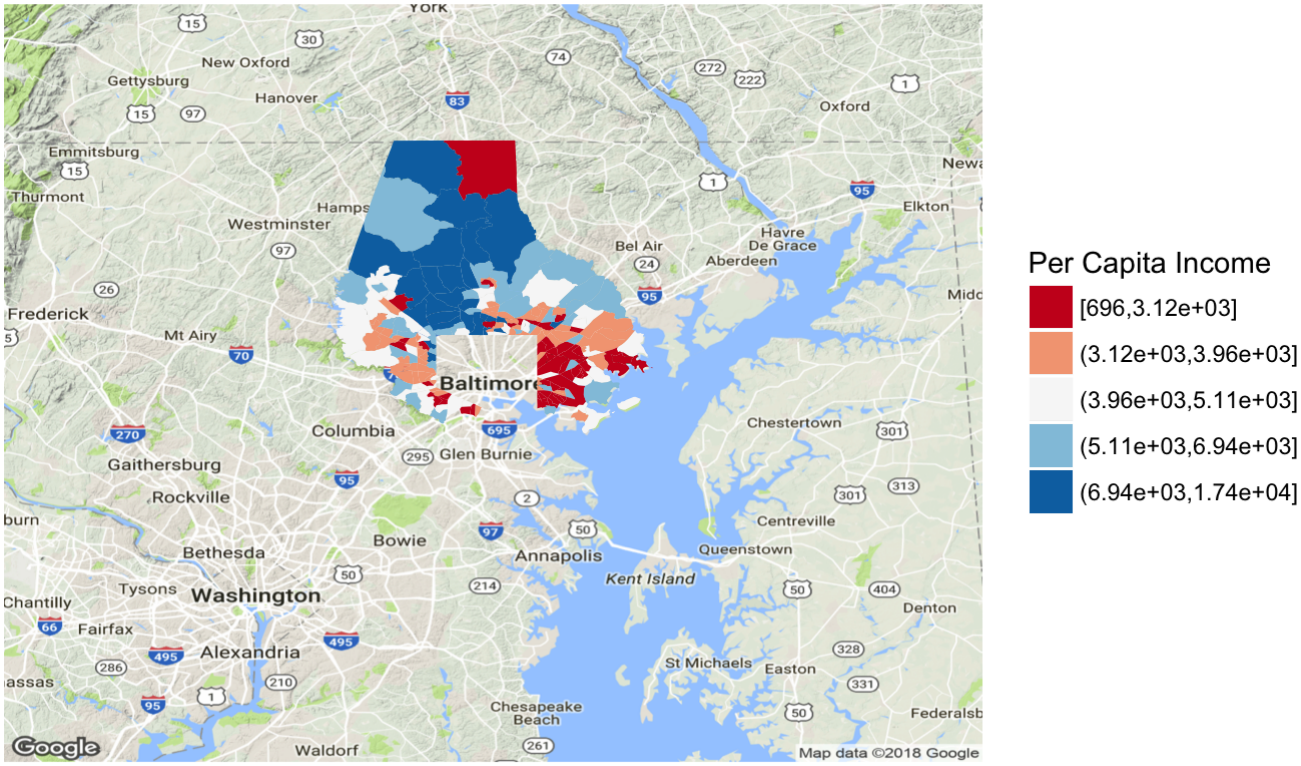
Below is a map of Median Household Income in Baltimore City using ggmap.

```
BaltimoreMap <- get_map(location=myLocation, zoom = 9,
source="google", maptype="terrain", crop=FALSE)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=39.29,-76.61&zoom
=9&size=640x640&scale=2&maptype=terrain&language=en-EN&sensor=false
```

```
ggmap(BaltimoreMap) +
  geom_polygon(
    data = balcit_df,
    aes(x = long, y = lat, group = group,
        fill = cut_number(B19301_001M, 5))) +
  scale_fill_brewer("Per Capita Income", palette = "RdBu") +
  ggtitle("Per Capita Income in Baltimore County") +
  theme(line = element_blank(),
        axis.text=element_blank(),
        axis.title=element_blank(),
        panel.background = element_blank()) +
  coord_equal()
```

Per Capita Income in Baltimore County

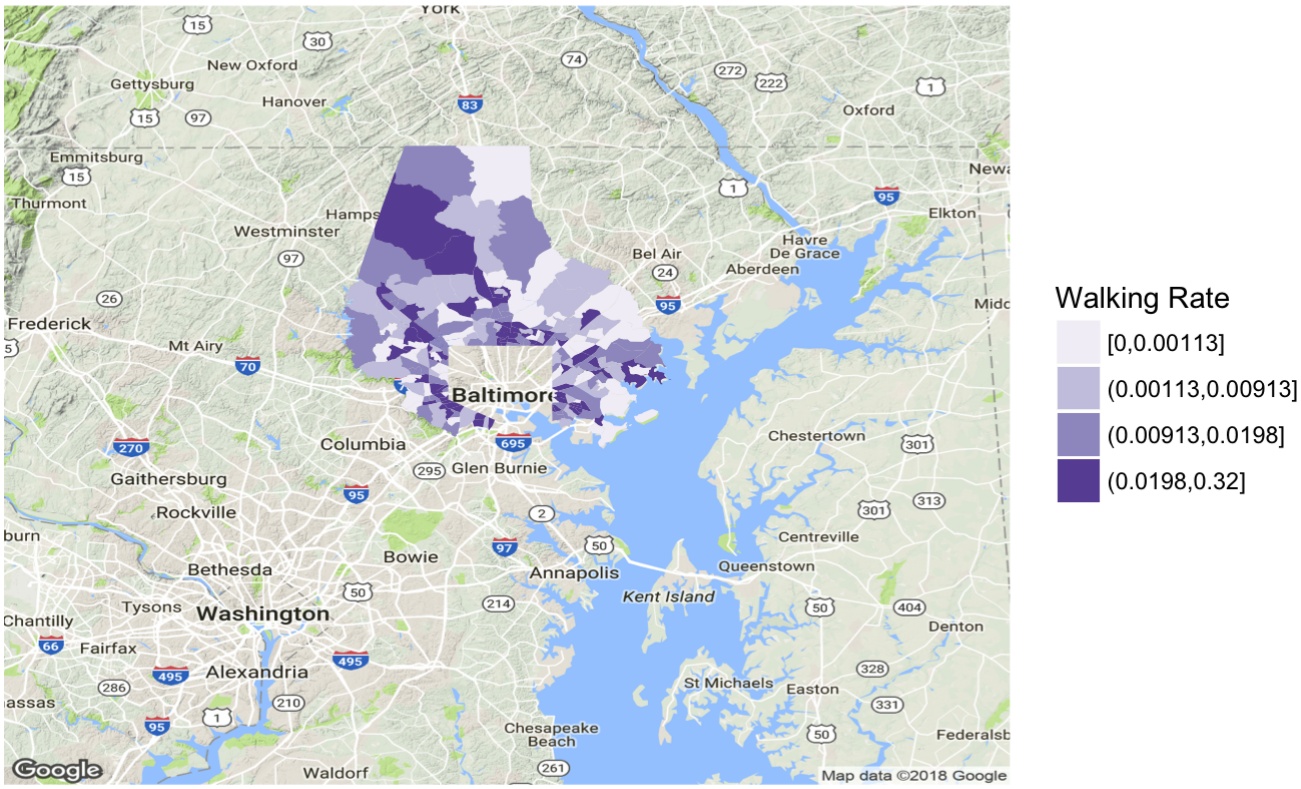


Below is a map of Rental Rate in Baltimore City using ggmap.

```
BaltimoreMap <- get_map(location=myLocation, zoom =9,
source="google", maptype="terrain", crop=FALSE)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=39.29,-76.61&zoom
=9&size=640x640&scale=2&maptype=terrain&language=en-EN&sensor=false
```

```
balcit_comp.sp <- as(balcit_comp, 'Spatial')
ggmap(BaltimoreMap) +
  geom_polygon(
    data = balcit_df,
    aes(x = long, y = lat, group = group,
        fill = cut_number(walking_rate, 4))) +
  scale_fill_brewer("Walking Rate", palette = "Purples") +
  ggtitle("Walking Rate in Baltimore County") +
  theme(line = element_blank(),
        axis.text=element_blank(),
        axis.title=element_blank(),
        panel.background = element_blank()) +
  coord_equal()
```



EXTRA CREDIT LETS GO!!

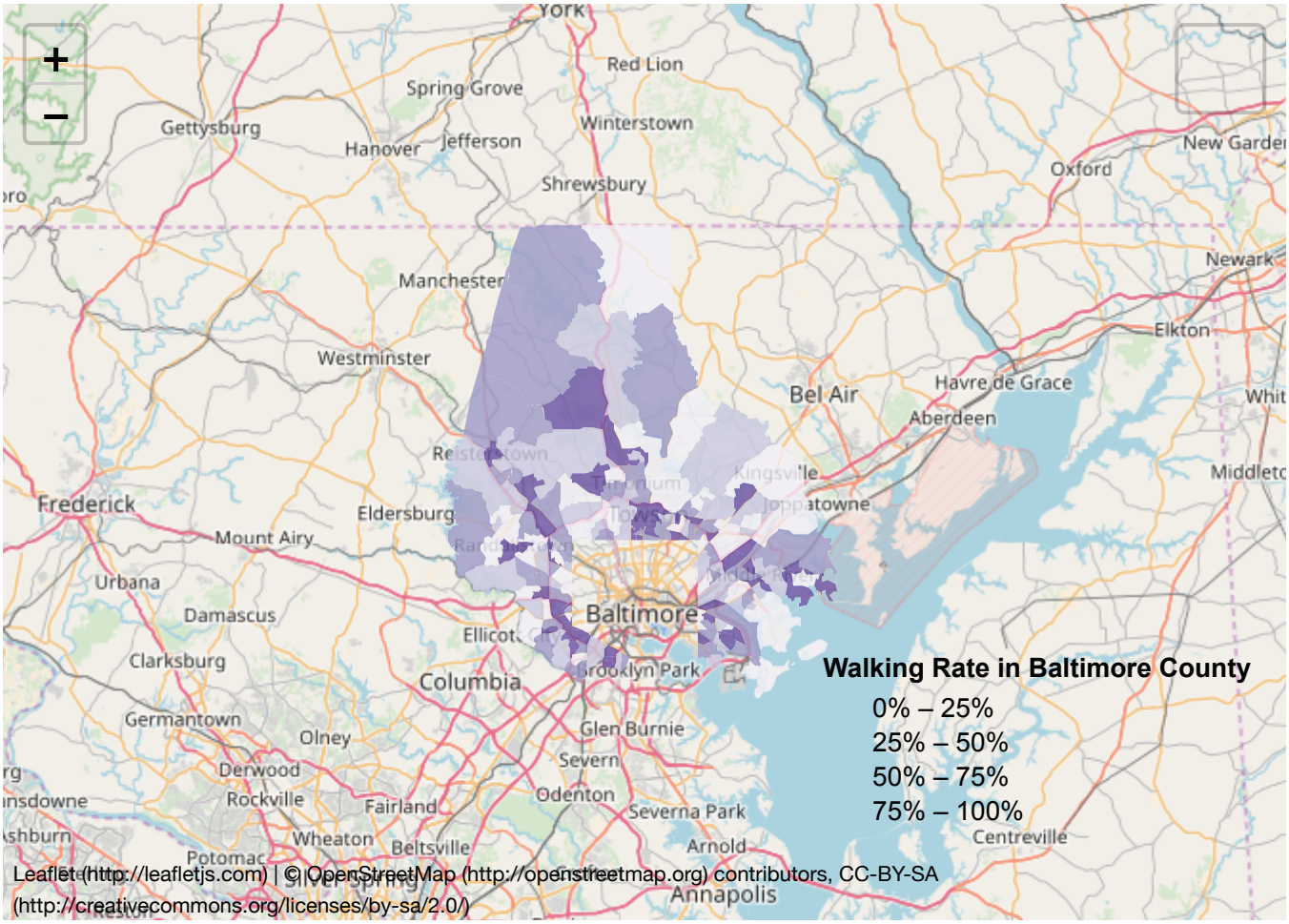
```
#install.packages("leaflet")
#install.packages("tmap")
library(leaflet)
```

```
## Warning: package 'leaflet' was built under R version 3.4.4
```

```
library(tmap)
```

```
## Warning: package 'tmap' was built under R version 3.4.4
```

```
pal_fun <- colorQuantile("Purples", NULL, n = 4)

p_popup <- paste0("<strong>Walking Rate: </strong>", balcit_comp.sp$walking_rate)
leaflet(balcit_comp.sp) %>%
  addPolygons(
    stroke = FALSE,
    fillColor = ~pal_fun(walking_rate),
    fillOpacity = 0.8, smoothFactor = 0.5,
    popup = p_popup,
    group = "Baltimore County") %>%
    addTiles(group = "OSM") %>%
  addProviderTiles("CartoDB.DarkMatter", group = "Carto") %>%
  addLegend("bottomright",
            pal=pal_fun,
            values=~walking_rate,
            title = 'Walking Rate in Baltimore County') %>%
  addLayersControl(baseGroups = c("OSM", "Carto"),
                   overlayGroups = c("Baltimore County"))
```



```
library(leaflet)
library(tmap)
library(classInt)
pal <- brewer.pal(5, "OrRd")
breaks_qt <- classIntervals(balcit_comp.sp$B19301_001E,
                            n = 5,
                            style = "quantile")
pal_fun <- colorQuantile("Reds", NULL, n = 5)

p_popup <- paste0("<strong>B19301_001E: </strong>", balcit_comp.sp$B19301_001E)

leaflet(balcit_comp.sp) %>%
  addPolygons(
    stroke = FALSE,
    fillColor = ~pal_fun(B19301_001E),
    fillOpacity = 0.8, smoothFactor = 0.5,
    popup = p_popup,
    group = "Baltimore County") %>%
    addTiles(group = "OSM") %>%
  addProviderTiles("CartoDB.DarkMatter", group = "Carto") %>%
  addLegend("bottomright",
            colors = brewer.pal(5, "YlOrRd"),
             labels = paste0("$", as.character(round(breaks_qt$brks[-1]))),
             title = 'Per Capita Income ') %>%
  addLayersControl(baseGroups = c("OSM", "Carto"),
                   overlayGroups = c("Baltimore County"))
```