

Spatial Epi Project

Leah Moubadder, Jasmine Aqua, Susan Hoffman

11/11/2021

Contents

Background	2
Outcomes of interest (by census tract):	2
Exposure of interest (aggregated to census tract):	2
Setting up	2
Editing data	3
Creating redlining maps	6
Redline Maps Output	11
Creating the 15 panel map	14
Density plots	19
Global Moran's I and Spatial Durbin Model	23
Create neighbors	23
Aspatial regression: Georgia	24
Spatial regression: Spatial Durbin Model (SDM) among all cities	25
Aspatial regression: Atlanta	25
Spatial regression: SDM in Atlanta	26
Aspatial regression: Macon	27
Spatial regression: SDM in Macon	27
Aspatial regression: Columbus	28
Spatial regression: SDM in Columbus	28
Aspatial regression: Augusta	29
Spatial regression: SDM in Augusta	30
Aspatial regression: Savannah	30
Spatial regression: SDM in Savannah	31

Background

A prime example of structural racism is the 1930s practice of redlining by the Home Owners Loan Corporation (HOLC). Although redlining is now illegal, the structure remains and impacts health outcomes today.

Aim: to understand the impact of redlining in Georgia's HOLC regions on environmental health hazards (diesel particulate matter [PM] levels, respiratory hazard index, and cancer risk due to air toxics)

Outcomes of interest (by census tract):

- (1) National Scale Air Toxics Assessment Air Toxics Cancer Risk (CANCER)
- (2) National Scale Air Toxics Assessment Respiratory Hazard Index (RESP)
- (3) National Scale Air Toxics Assessment Diesel PM (DPM) (DSLPM)

Hazard index: The sum of the hazard quotients for toxics that affect the same target organ/target system. An HI <=1 indicates noncancer effect not likely to occur

Cancer risk: the results of cancer dose-response assessments are converted into a unit risk estimate. This is multiplied by the estimated inhalation exposure concentration over a lifetime (70 years) to estimate an individuals lifetime cancer risk.

Data were publicly available and provided by the Environmental Protection Agency's National Air Toxics Assessment (NATA). Our data was for the year 2014 at the census-tract level and can be accessed at: <https://www.epa.gov/national-air-toxics-assessment/2014-nata-assessment-results>.

Exposure of interest (aggregated to census tract):

Historic redlining score

Data were publicly available and obtained from University of Michigan Social Science Institute, by Helen Meier at: <https://www.openicpsr.org/openicpsr/project/141121/version/V2/view>.

Historic redlining grades were coded as following:

A or "Best" = 1

B or "Desirable" = 2

C or "Declining" = 3

D or "Hazardous" = 4

The historical grades were then weighted to a census-tract level continuous score based on methods available in detail at: <https://ncrc.org/holc-health/>. The continuous score ranged from 1.0 to 4.0, with higher values representing more hazardous grades.

Setting up

NOTE: EJScreen data

For ease of importing to GitHub (file size issues), we limited the data to only areas within GA before loading the dataset to the project

```

# Packages
library(tmap) # Mapping spatial file
library(tidyverse) # data wrangling tools
library(dplyr) # data wrangling tools
library(sp) # spatial methods package
library(readxl) # import excel files
library(rgdal) # spatial methods package
library(readr) # data file loading package
library(sf) # spatial methods package
library(viridis) # color maps in base R
library(ggmap) # make maps from Google Maps
library(spdep) # to create spatial neighbors
library(spatialEco) # Spatial data cleaning
library(spatialreg) # spatial methods package

setwd("C:/Users/SSBUCKE/OneDrive - Emory University/Spatial Project/spatialfinal2021")

# Data files

## outcome data
ej_ga <- read_csv("Data/ej_ga.csv",
                    col_types = cols(...1 = col_skip())) %>%
  select('ID', 'DSLPM', 'CANCER', 'RESP')

## exposure geometry data
HOLC_map <- readOGR(dsn=path.expand("Data/HRS2010-Shapefiles/HRS2010"),
                      layer="HRS2010")

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\SSBUCKE\OneDrive - Emory University\Spatial Project\spatialfinal2021\Data\HRS2010"
## with 12888 features
## It has 16 fields
## Integer64 fields read as strings: OBJECTID_1

## exposure attribute data
HOLC_score <- read_excel("Data/Historic Redlining Score 2010.xlsx")

## Importing Georgia Census Tract Geographic Boundary file
## updated 2020 data

ga_tracts_10 <- readOGR(dsn=path.expand("Data/tl_2020_13_all"),layer="tl_2020_13_tract10")

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\SSBUCKE\OneDrive - Emory University\Spatial Project\spatialfinal2021\Data\tl_2020"
## with 1969 features
## It has 12 fields
## Integer64 fields read as strings: ALAND10 AWATER10

```

Editing data

Data convention HOLC_map_citizenname will be used to create datasets with extra census tracts for mapping purposes.

- HOLC_full: Census tract + HOLC data, all Georgia census tracts
- HOLC_full_georgia: Census tract + HOLC data, all Georgia census tracts, only GA census tracts with HOLC data
- HOLC_full_atlanta: Census tract + HOLC data, only census tracts in Atlanta with HOLC data
- HOLC_full_augusta: Census tract + HOLC data, only census tracts in Augusta with HOLC data
- HOLC_full_columbus: Census tract + HOLC data, only census tracts in Columbus with HOLC data
- HOLC_full_macon: Census tract + HOLC data, only census tracts in Macon with HOLC data
- HOLC_full_savannah: Census tract + HOLC data, only census tracts in Savannah with HOLC data

To prepare for our redlining maps, we first created four new categories in the variable `HRS10_bins` based on the continuous HOLC score measure `HRS10`. Although previous work using these data have presented redlining maps using quartiles of the continuous HOLC measure, we decided to create these new categories with fixed cutpoints in order to preserve the interpretation / meaning of the original HOLC categories.

- $1.0 \leq HRS10 < 1.5$: “Best”
- $1.5 \leq HRS10 < 2.5$: “Still Desirable”
- $2.5 \leq HRS10 < 3.5$: “Definitely Declining”
- $3.5 \leq HRS10 \leq 4.0$: “Hazardous”

```
# Cleaning HOLC score data
HOLC_score2 <- HOLC_score %>%
  filter(substr(GEOID10, 1, 2) == "13") # restricting to only GA

# cleaning EJScreen data

ej_ga2 <- ej_ga %>%
  mutate(GEOID10 = substr(ej_ga$ID, 1, 11)) %>%
  select(-'ID')

# combining all 3 datasets together

ej_ga2 <- ej_ga2[!duplicated(ej_ga2$GEOID10),]

ej_ga2 <- sp::merge(ga_tracts_10, ej_ga2,
  by = 'GEOID10',
  all.y = T,
  duplicateGeoms = T,
  na.strings = 'Missing')

full_data <- sp::merge(ej_ga2, HOLC_score2,
  by = 'GEOID10',
  all.y = T,
  duplicateGeoms = T,
  na.strings = 'Missing')

# Subsetting data into
```

```

## All GA (only redlined areas)
## Atlanta redlined areas
## Augusta redlined areas
## Macon redlined areas
## Columbus redlined areas
## Savannah redlined areas

# All GA
ga_ids <- HOLC_score2$GEOID10
full_data_georgia <- subset(full_data, GEOID10 %in% ga_ids)

# Creating new, interpretable HOLC categories based on continuous measure

full_data_georgia@data <- full_data_georgia@data %>%
  mutate(HRS10_bins = case_when(HRS10 >= 1 & HRS10 < 1.5 ~ "1", # new categories
                                HRS10 >= 1.5 & HRS10 < 2.5 ~ "2",
                                HRS10 >= 2.5 & HRS10 < 3.5 ~ "3",
                                HRS10 >= 3.5 & HRS10 <= 4 ~ "4"))

# writeOGR(obj=full_data_georgia, dsn="tempdir", layer="full_data_georgia", driver="ESRI Shapefile")

# Atlanta only
atlanta <- HOLC_score2 %>%
  filter(CBSA=="12060")
atlanta_ids <- atlanta$GEOID10
full_data_atlanta <- subset(full_data_georgia, GEOID10 %in% atlanta_ids)

# writeOGR(obj=full_data_atlanta, dsn="tempdir", layer="full_data_atlanta", driver="ESRI Shapefile")

# Augusta only
augusta <- HOLC_score2 %>%
  filter(CBSA=="12260")
augusta_ids <- augusta$GEOID10
full_data_augusta<- subset(full_data_georgia, GEOID10 %in% augusta_ids)

# writeOGR(obj=full_data_augusta, dsn="tempdir", layer="full_data_augusta", driver="ESRI Shapefile")

# Columbus only
columbus <- HOLC_score2 %>%
  filter(CBSA=="17980")
columbus_ids <- columbus$GEOID10
full_data_columbus <- subset(full_data_georgia, GEOID10 %in% columbus_ids)

# writeOGR(obj=full_data_columbus, dsn="tempdir", layer="full_data_columbus", driver="ESRI Shapefile")

# Macon only
macon <- HOLC_score2 %>%
  filter(CBSA=="31420")
macon_ids <- macon$GEOID10
full_data_macon <- subset(full_data_georgia, GEOID10 %in% macon_ids)

# writeOGR(obj=full_data_macon, dsn="tempdir", layer="full_data_macon", driver="ESRI Shapefile")

```

```

# Savannah only
savannah <- HOLC_score2 %>%
  filter(CBSA=="42340")
savannah_ids <- savannah$GEOID10
full_data_savannah <- subset(full_data_georgia, GEOID10 %in% savannah_ids)

# writeOGR(obj=full_data_savannah, dsn="tempdir", layer="full_data_savannah", driver="ESRI Shapefile")

```

Creating redlining maps

Using the HOLC categories described above, we created a map for each HOLC region layered on top of Google Maps in order to better visualize the surrounding environment using methods described in detail at: <https://www.r-bloggers.com/2016/03/plotting-choropleths-from-shapefiles-in-r-with-ggmap-toronto-neighbourhoods-by-population/>.

```

register_google(key = "Your Google Key Here")

# Creating a custom map palette
MyPalette <- c("#A8FF33", "#81FOFF", "#FAFF93", "#FF9693")
MyLabels <- c("Low (Q1)", "Medium (Q2)", "High (Q3)", "Very High (Q4)")

## Creating redlining map of Atlanta
qmap('Atlanta, GA', zoom = 11) +
  geom_polygon(aes(x = long, y = lat, group = group), data = full_data_atlanta,
               colour = 'white', fill = 'black', alpha = .4, size = .3)

# Add demographic data
# The GEOID10 ID is a string - change it to a integer
full_data_atlanta@data$GEOID10 <- as.numeric(full_data_atlanta@data$GEOID10)

# GGPLOT
points_atlanta <- fortify(full_data_atlanta, region = 'GEOID10')

# Plot the census tracts
atlanta_gmap <- qmap("Atlanta, Georgia", zoom=11)
atlanta_gmap +geom_polygon(aes(x=long,y=lat, group=group), data=points_atlanta, fill=NA) +
  geom_polygon(aes(x=long,y=lat, group=group), data=points_atlanta, color='black', fill=NA)

# Merge the shapefile data with the HOLC data, using GEOID10
points_atlanta_2 <- merge(points_atlanta, full_data_atlanta, by.x='id', by.y='GEOID10', all.x=TRUE)
points_atlanta_2$HRS10_bins <- sapply(points_atlanta_2$HRS10_bins, as.factor)

# Plot
atlanta_gmap + geom_polygon(aes(x=long,y=lat, group=group, fill=HRS10_bins), data=points_atlanta_2, colo
  scale_fill_manual(name = "Historic Redlining Score Quartile",
                    labels = c("1" = "Low (Q1)",
                              "2" = "Medium (Q2)",
                              "3" = "High (Q3)",
                              "4" = "Very High (Q4)"),
                    values = c("1" = "#A8FF33",

```

```

        "2" = "#81FOFF",
        "3" = "#FAFF93",
        "4" = "#FF9693")) +
theme(legend.position="top",
      legend.key.size = unit(0.1, 'cm'), #change legend key size
      legend.key.height = unit(0.1, 'cm'), #change legend key height
      legend.key.width = unit(0.3, 'cm'), #change legend key width
      legend.title = element_text(size=8), #change legend title font size
      legend.text = element_text(size=8)) #change legend text font size

## Creating redlining map of Augusta
qmap('Augusta, GA', zoom = 12) +
  geom_polygon(aes(x = long, y = lat, group = group), data = full_data_augusta,
               colour = 'white', fill = 'black', alpha = .4, size = .3)

# Add demographic data
# The GEOID10 ID is a string - change it to a integer
full_data_augusta@data$GEOID10 <- as.numeric(full_data_augusta@data$GEOID10)

# GG PLOT
points_augusta <- fortify(full_data_augusta, region = 'GEOID10')

# Plot the census tracts
augusta_gmap <- qmap("Augusta, Georgia", zoom=12)
augusta_gmap +geom_polygon(aes(x=long,y=lat, group=group), data=points_augusta, fill=NA) +
  geom_polygon(aes(x=long,y=lat, group=group), data=points_augusta, color='black', fill=NA)

# Merge the shapefile data with the HOLC data, using GEOID10
points_augusta_2 <- merge(points_augusta, full_data_augusta, by.x='id', by.y='GEOID10', all.x=TRUE)
points_augusta_2$HRS10_bins <- sapply(points_augusta_2$HRS10_bins, as.factor)

# Plot
augusta_gmap + geom_polygon(aes(x=long,y=lat, group=group, fill=HRS10_bins), data=points_augusta_2, col
  scale_fill_manual(name = "Historic Redlining Score Quartile",
                    labels = c("1" = "Low (Q1)",
                              "2" = "Medium (Q2)",
                              "3" = "High (Q3)",
                              "4" = "Very High (Q4)"),
                    values = c("1" = "#A8FF33",
                              "2" = "#81FOFF",
                              "3" = "#FAFF93",
                              "4" = "#FF9693")) +
theme(legend.position="top",
      legend.key.size = unit(0.1, 'cm'), #change legend key size
      legend.key.height = unit(0.1, 'cm'), #change legend key height
      legend.key.width = unit(0.3, 'cm'), #change legend key width
      legend.title = element_text(size=8), #change legend title font size
      legend.text = element_text(size=8)) #change legend text font size

## Creating redlining map of Columbus
qmap('Columbus, GA', zoom = 12) +
  geom_polygon(aes(x = long, y = lat, group = group), data = full_data_columbus,
               colour = 'white', fill = 'black', alpha = .4, size = .3)

```

```

# Add demographic data
# The GEOID10 ID is a string - change it to a integer
full_data_columbus@data$GEOID10 <- as.numeric(full_data_columbus@data$GEOID10)

# GG PLOT
points_columbus <- fortify(full_data_columbus, region = 'GEOID10')

# Plot the census tracts
columbus_gmap <- qmap("Columbus, Georgia", zoom=12)
columbus_gmap +geom_polygon(aes(x=long,y=lat, group=group), data=points_columbus, fill=NA) +
  geom_polygon(aes(x=long,y=lat, group=group), data=points_columbus, color='black', fill=NA)

# Merge the shapefile data with the HOLC data, using GEOID10
points_columbus_2 <- merge(points_columbus, full_data_columbus, by.x='id', by.y='GEOID10', all.x=TRUE)

points_columbus_2$HRS10_bins <- sapply(points_columbus_2$HRS10_bins, as.factor)

# Plot
columbus_gmap + geom_polygon(aes(x=long,y=lat, group=group, fill=HRS10_bins), data=points_columbus_2, color='black') +
  scale_fill_manual(name = "Historic Redlining Score Quartile",
    labels = c("1" = "Low (Q1)",
              "2" = "Medium (Q2)",
              "3" = "High (Q3)",
              "4" = "Very High (Q4)"),
    values = c("1" = "#A8FF33",
              "2" = "#81FOFF",
              "3" = "#FAFF93",
              "4" = "#FF9693")) +
  theme(legend.position="top",
        legend.key.size = unit(0.1, 'cm'), #change legend key size
        legend.key.height = unit(0.1, 'cm'), #change legend key height
        legend.key.width = unit(0.3, 'cm'), #change legend key width
        legend.title = element_text(size=8), #change legend title font size
        legend.text = element_text(size=8)) #change legend text font size

## Creating redlining map of Macon
qmap('Macon, GA', zoom = 12) +
  geom_polygon(aes(x = long, y = lat, group = group), data = full_data_macon,
               colour = 'white', fill = 'black', alpha = .4, size = .3)

# Add demographic data
# The GEOID10 ID is a string - change it to a integer
full_data_macon@data$GEOID10 <- as.numeric(full_data_macon@data$GEOID10)

# GG PLOT
points_macon <- fortify(full_data_macon, region = 'GEOID10')

# Plot the census tracts
macon_gmap <- qmap("Macon, Georgia", zoom=12)
macon_gmap +geom_polygon(aes(x=long,y=lat, group=group), data=points_macon, fill=NA) +
  geom_polygon(aes(x=long,y=lat, group=group), data=points_macon, color='black', fill=NA)

```

```

# Merge the shapefile data with the HLOC data, using GEOID10
points_macon_2 <- merge(points_macon, full_data_macon, by.x='id', by.y='GEOID10', all.x=TRUE)
points_macon_2$HRS10_bins <- sapply(points_macon_2$HRS10_bins, as.factor)

# Plot
macon_gmap + geom_polygon(aes(x=long,y=lat, group=group, fill=HRS10_bins), data=points_macon_2, color='black')
  scale_fill_manual(name = "Historic Redlining Score Quartile",
    labels = c("1" = "Low (Q1)",
              "2" = "Medium (Q2)",
              "3" = "High (Q3)",
              "4" = "Very High (Q4)"),
    values = c("1" = "#A8FF33",
              "2" = "#81FOFF",
              "3" = "#FAFF93",
              "4" = "#FF9693")) +
  theme(legend.position="top",
    legend.key.size = unit(0.1, 'cm'), #change legend key size
    legend.key.height = unit(0.1, 'cm'), #change legend key height
    legend.key.width = unit(0.3, 'cm'), #change legend key width
    legend.title = element_text(size=8), #change legend title font size
    legend.text = element_text(size=8)) #change legend text font size

## Creating redlining map of Savannah
qmap('Savannah, GA', zoom = 13) +
  geom_polygon(aes(x = long, y = lat, group = group), data = full_data_savannah,
               colour = 'white', fill = 'black', alpha = .4, size = .3)

# Add demographic data
# The GEOID10 ID is a string - change it to a integer
full_data_savannah@data$GEOID10 <- as.numeric(full_data_savannah@data$GEOID10)

# GGPLOT
points_savannah <- fortify(full_data_savannah, region = 'GEOID10')

# Plot the census tracts
savannah_gmap <- qmap("Savannah, Georgia", zoom=13)
savannah_gmap +geom_polygon(aes(x=long,y=lat, group=group), data=points_savannah, fill=NA) +
  geom_polygon(aes(x=long,y=lat, group=group), data=points_savannah, color='black', fill=NA)

# Merge the shapefile data with the HLOC data, using GEOID10
points_savannah_2 <- merge(points_savannah, full_data_savannah, by.x='id', by.y='GEOID10', all.x=TRUE)

points_savannah_2$HRS10_bins <- sapply(points_savannah_2$HRS10_bins, as.factor)

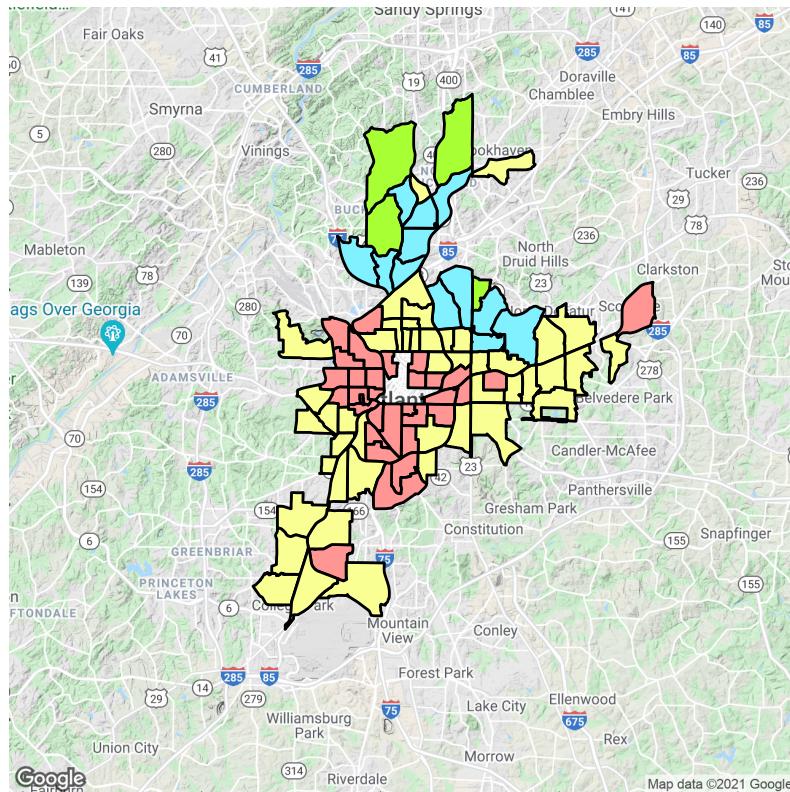
# Plot
savannah_gmap + geom_polygon(aes(x=long,y=lat, group=group, fill=HRS10_bins), data=points_savannah_2, color='black')
  scale_fill_manual(name = "Historic Redlining Score Quartile",
    labels = c("1" = "Low (Q1)",
              "2" = "Medium (Q2)",
              "3" = "High (Q3)",
              "4" = "Very High (Q4)"),
    values = c("1" = "#A8FF33",
              "2" = "#81FOFF",
              "3" = "#FAFF93",
              "4" = "#FF9693"))

```

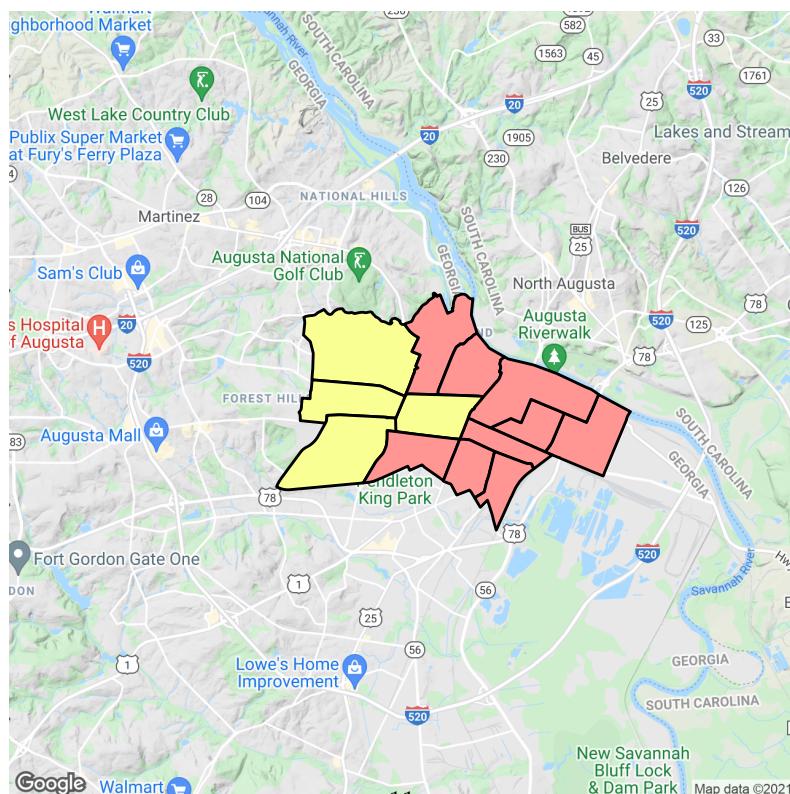
```
    "3" = "#FAFF93",
    "4" = "#FF9693")) +
theme(legend.position="top",
  legend.key.size = unit(0.1, 'cm'), #change legend key size
  legend.key.height = unit(0.1, 'cm'), #change legend key height
  legend.key.width = unit(0.3, 'cm'), #change legend key width
  legend.title = element_text(size=8), #change legend title font size
  legend.text = element_text(size=8)) #change legend text font size
```

Redline Maps Output

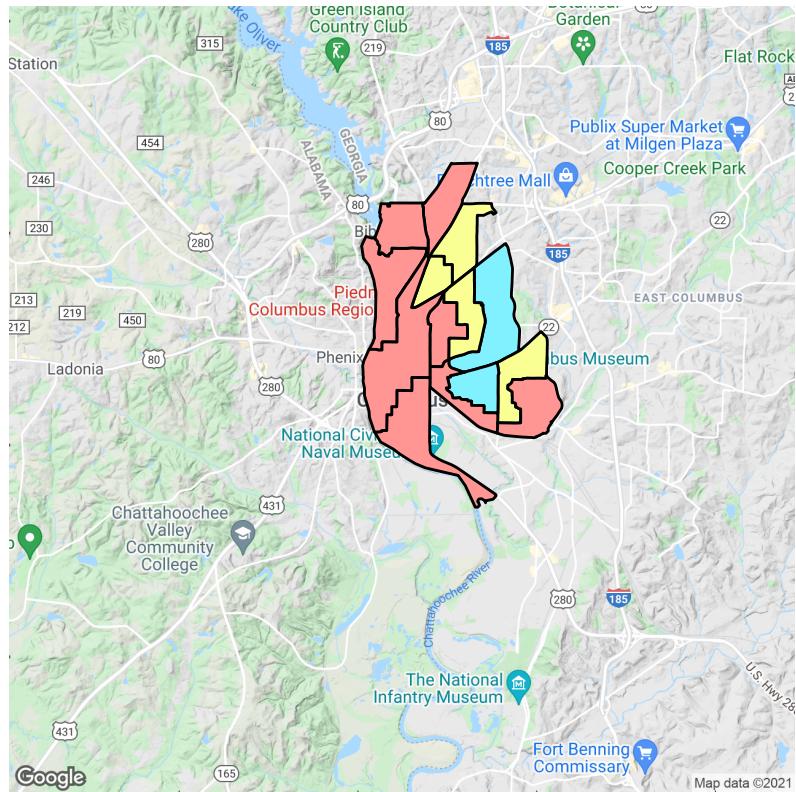
Historic Redlining Score Quartile Low (Q1) Medium (Q2) High (Q3) Very High (Q4)



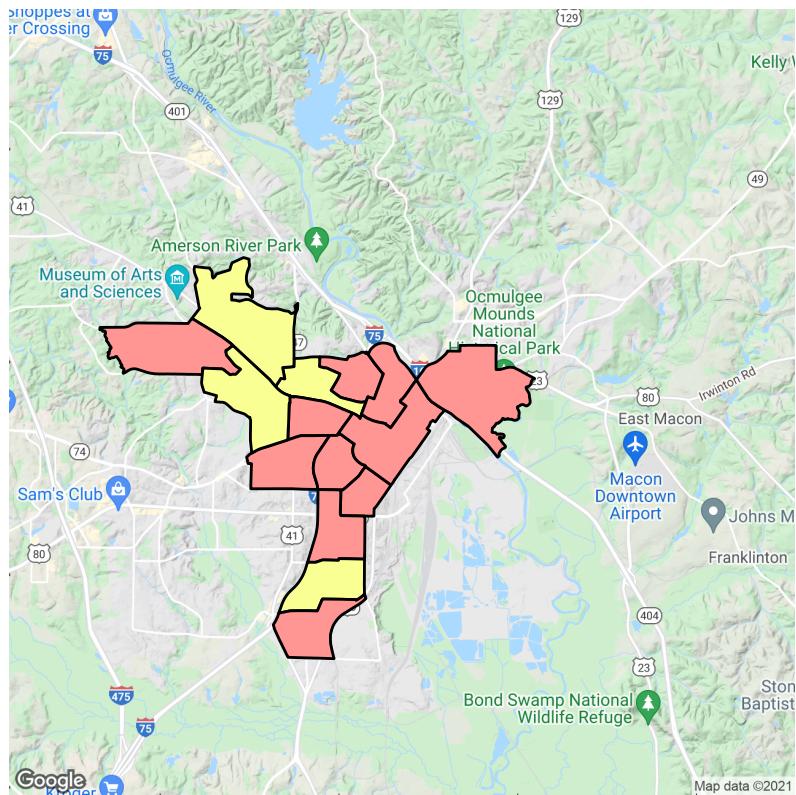
Historic Redlining Score Quartile Low (Q1) Medium (Q2) High (Q3) Very High (Q4)



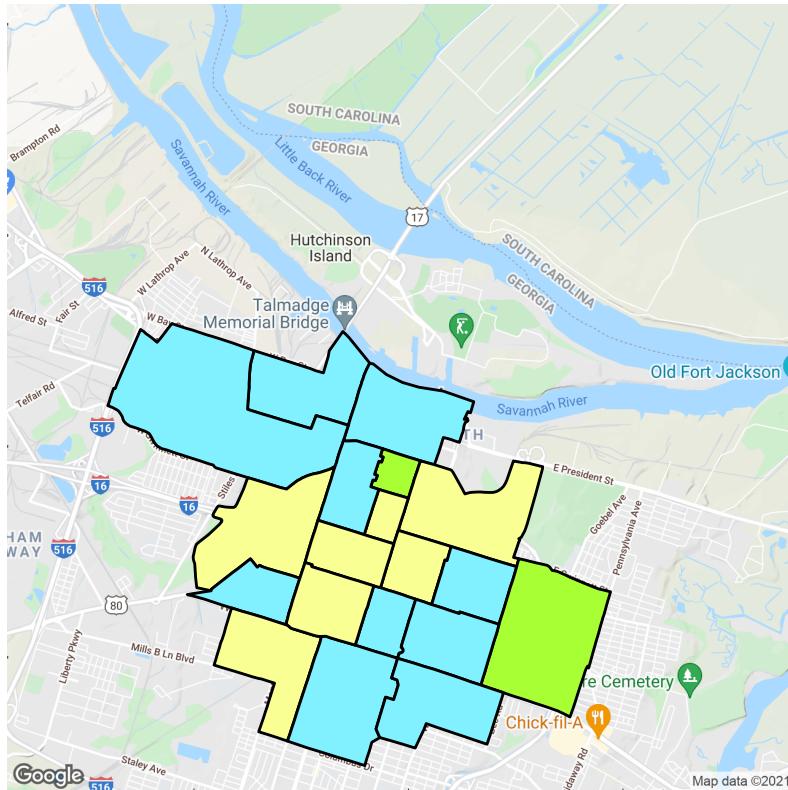
Historic Redlining Score Quartile Low (Q1) Medium (Q2) High (Q3) Very High (Q4)



Historic Redlining Score Quartile Low (Q1) Medium (Q2) High (Q3) Very High (Q4)



Historic Redlining Score Quartile █ Low (Q1) █ Medium (Q2) █ High (Q3) █ Very High (Q4)



Creating the 15 panel map

Note: for the poster, individual images were exported in order to make a “cleaner” figure. For this RMarkdown, we are producing the 15 panel map using `tmap_arrange`, so the two figures might look different. Since we are treating each city as spatial islands, local quantiles were used.

```
# Diesel PM

dpm_atl <- tm_shape(full_data_atlanta) +
  tm_fill('DSLPM',
    style = 'quantile',
    palette = 'BuPu',
    title = 'Diesel PM') +
  tm_borders() +
  tm_layout(main.title = 'Diesel PM: \nAtlanta',
    main.title.size = 0.9,
    legend.outside = T,
    legend.outside.size = .5,
    frame = F)

dpm_aug <- tm_shape(full_data_augusta) +
  tm_fill('DSLPM',
    style = 'quantile',
```

```

    palette = 'BuPu',
    title = 'Diesel PM') +
tm_borders() +
tm_layout(main.title = 'Diesel PM: \nAugusta',
          main.title.size = 0.9,
          legend.outside = T,
          legend.outside.size = .5,
          frame = F)

dpm_mac <- tm_shape(full_data_macon) +
  tm_fill('DSLPM',
          style = 'quantile',
          palette = 'BuPu',
          title = 'Diesel PM') +
  tm_borders() +
  tm_layout(main.title = 'Diesel PM: \nMacon',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

dpm_sav <- tm_shape(full_data_savannah) +
  tm_fill('DSLPM',
          style = 'quantile',
          palette = 'BuPu',
          title = 'Diesel PM') +
  tm_borders() +
  tm_layout(main.title = 'Diesel PM: \nSavannah',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

dpm_col <- tm_shape(full_data_columbus) +
  tm_fill('DSLPM',
          style = 'quantile',
          palette = 'BuPu',
          title = 'Diesel PM') +
  tm_borders() +
  tm_layout(main.title = 'Diesel PM: \nColumbus',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

# RESP

res_atl <- tm_shape(full_data_atlanta) +
  tm_fill('RESP',

```

```

    style = 'quantile',
    palette = 'RdPu',
    title = 'Respiratory Hazard') +
tm_borders() +
tm_layout(main.title = 'Respiratory Hazard: \nAtlanta',
          main.title.size = 0.9,
          legend.outside = T,
          legend.outside.size = .5,
          frame = F)

res_aug <- tm_shape(full_data_augusta) +
  tm_fill('RESP',
          style = 'quantile',
          palette = 'RdPu',
          title = 'Respiratory Hazard') +
  tm_borders() +
  tm_layout(main.title = 'Respiratory Hazard: \nAugusta',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

res_mac <- tm_shape(full_data_macon) +
  tm_fill('RESP',
          style = 'quantile',
          palette = 'RdPu',
          title = 'Respiratory Hazard') +
  tm_borders() +
  tm_layout(main.title = 'Respiratory Hazard: \nMacon',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

res_sav <- tm_shape(full_data_savannah) +
  tm_fill('RESP',
          style = 'quantile',
          palette = 'RdPu',
          title = 'Respiratory Hazard') +
  tm_borders() +
  tm_layout(main.title = 'Respiratory Hazard: \nSavannah',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

res_col <- tm_shape(full_data_columbus) +
  tm_fill('RESP',
          style = 'quantile',

```

```

    palette = 'RdPu',
    title = 'Respiratory Hazard') +
tm_borders() +
tm_layout(main.title = 'Respiratory Hazard: \nColumbus',
          main.title.size = 0.9,
          legend.outside = T,
          legend.outside.size = .5,
          frame = F)

# Cancer

can_atl <- tm_shape(full_data_atlanta) +
  tm_fill('CANCER',
          style = 'quantile',
          palette = 'PuBu',
          title = 'Cancer Risk') +
  tm_borders() +
  tm_layout(main.title = 'Cancer Hazard: \nAtlanta',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

can_aug <- tm_shape(full_data_augusta) +
  tm_fill('CANCER',
          style = 'quantile',
          palette = 'PuBu',
          title = 'Cancer Risk') +
  tm_borders() +
  tm_layout(main.title = 'Cancer Hazard: \nAugusta',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

can_mac <- tm_shape(full_data_macon) +
  tm_fill('CANCER',
          style = 'quantile',
          palette = 'PuBu',
          title = 'Cancer Risk') +
  tm_borders() +
  tm_layout(main.title = 'Cancer Hazard: \nMacon',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

can_sav <- tm_shape(full_data_savannah) +
  tm_fill('CANCER',

```

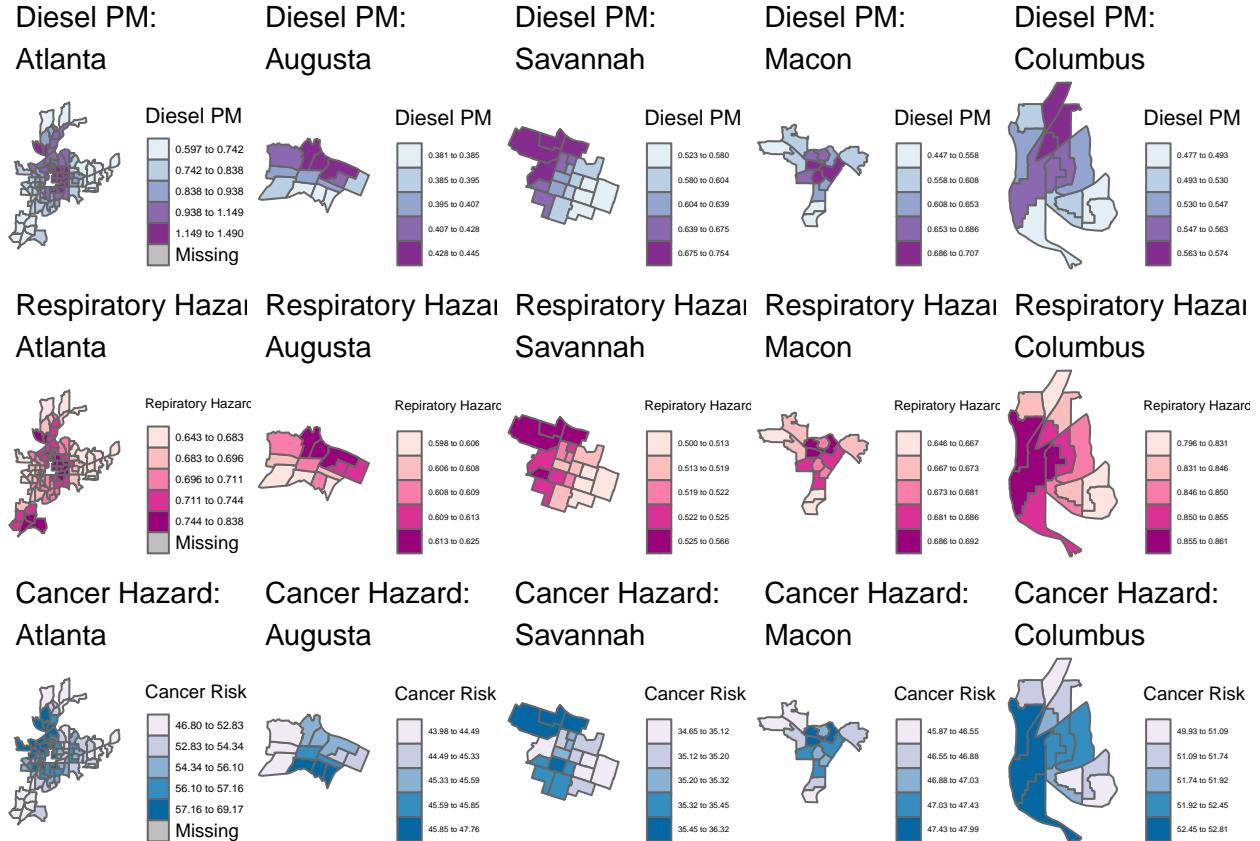
```

    style = 'quantile',
    palette = 'PuBu',
    title = 'Cancer Risk') +
tm_borders() +
tm_layout(main.title = 'Cancer Hazard: \nSavannah',
          main.title.size = 0.9,
          legend.outside = T,
          legend.outside.size = .5,
          frame = F)

can_col <- tm_shape(full_data_columbus) +
  tm_fill('CANCER',
          style = 'quantile',
          palette = 'PuBu',
          title = 'Cancer Risk') +
  tm_borders() +
  tm_layout(main.title = 'Cancer Hazard: \nColumbus',
            main.title.size = 0.9,
            legend.outside = T,
            legend.outside.size = .5,
            frame = F)

tmap_arrange(dpm_atl, dpm_aug, dpm_sav, dpm_mac, dpm_col,
             res_atl, res_aug, res_sav, res_mac, res_col,
             can_atl, can_aug, can_sav, can_mac, can_col,
             nrow = 3,
             ncol = 5)

```



Density plots

In order to be transparent about the data distributions in each city, we created density plots of each health hazard by HOLC region in GA.

```
# Creating a dataset that removes the spatial aspect to run the plots

hist <- as.data.frame(full_data_georgia) %>%
  select(c(CBSA, CANCER, DSLPM, RESP))

# Creating a label for each of the HOLC regions in GA
hist$cities <- factor(hist$CBSA,
  levels = c(12060, 42340, 17980, 31420, 12260),
  labels = c('Atlanta', 'Savannah', 'Columbus', 'Macon', 'Augusta'))

# Creating desity plots

dslpm <- ggplot(hist, aes(x = DSLPM, fill = cities)) +
  geom_density(alpha = 0.4) +
  ggtitle('NATA Diesel PM Density per City in Georgia') +
  xlab('Diesel PM') +
  ylab('Density') +
  labs(fill = 'HOLC Regions') +
```

```

theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.background = element_blank(),
      axis.line = element_line(colour = "black"))

cancer <- ggplot(hist, aes(x = CANCER, fill = cities)) +
  geom_density(alpha = 0.4) +
  ggtitle('Cancer Risk per City in Georgia') +
  xlab('Cancer Risk') +
  ylab('Density') +
  labs(fill = 'HOLC Regions') +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"))

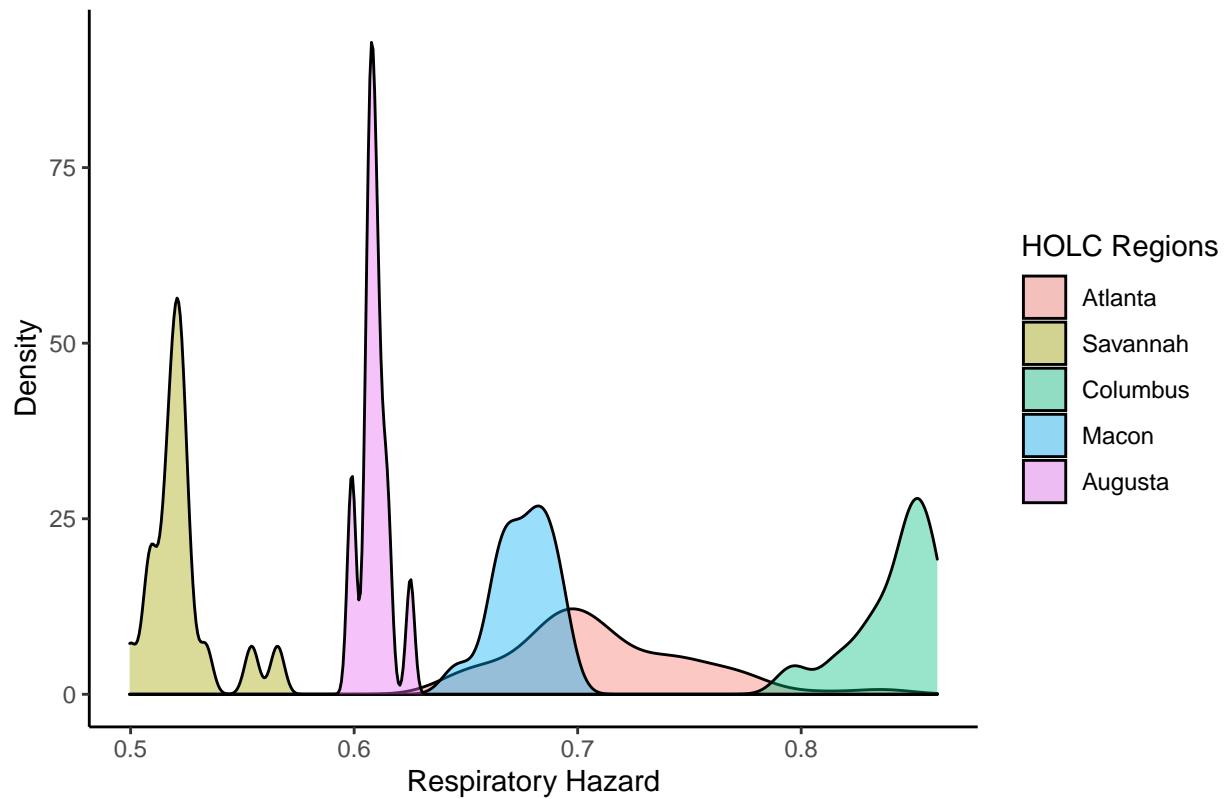
resp <- ggplot(hist, aes(x = RESP, fill = cities)) +
  geom_density(alpha = 0.4) +
  ggtitle('Respiratory Hazard per City in Georgia') +
  xlab('Respiratory Hazard') +
  ylab('Density') +
  labs(fill = 'HOLC Regions') +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"))

# Printing Density plots

resp

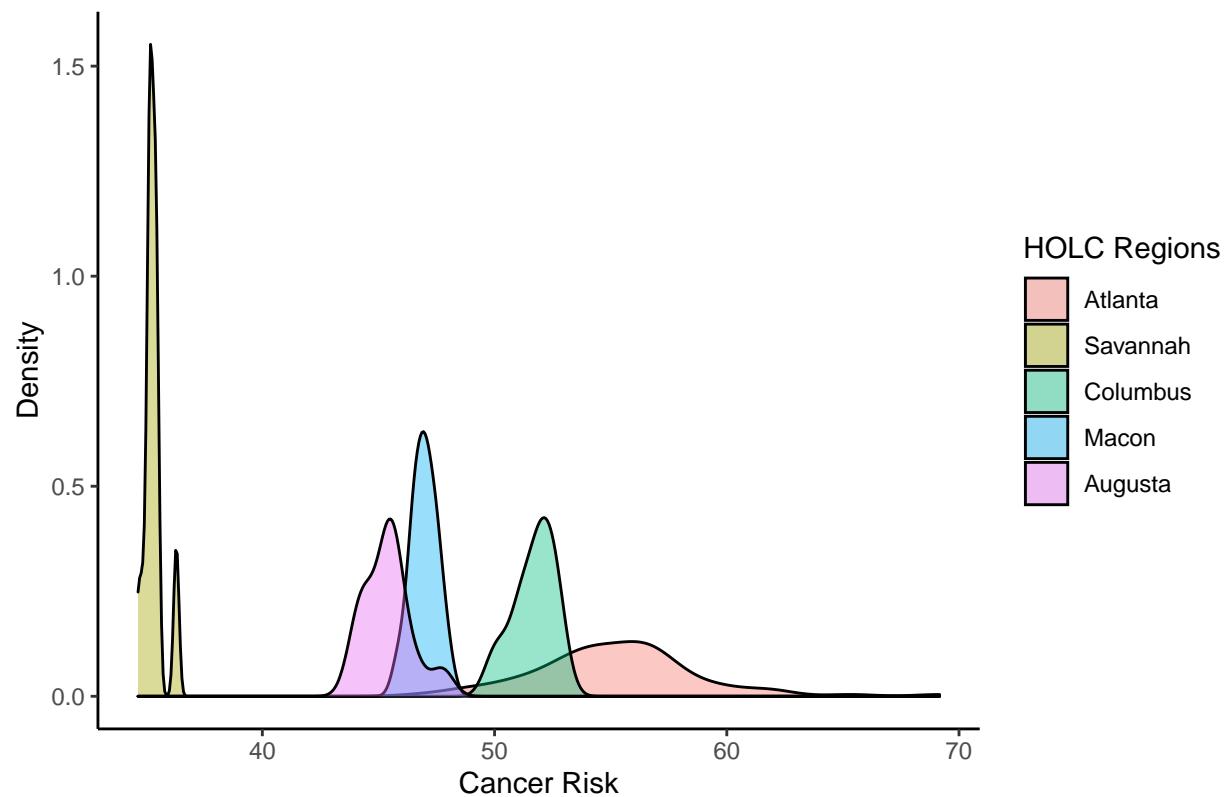
```

Respiratory Hazard per City in Georgia



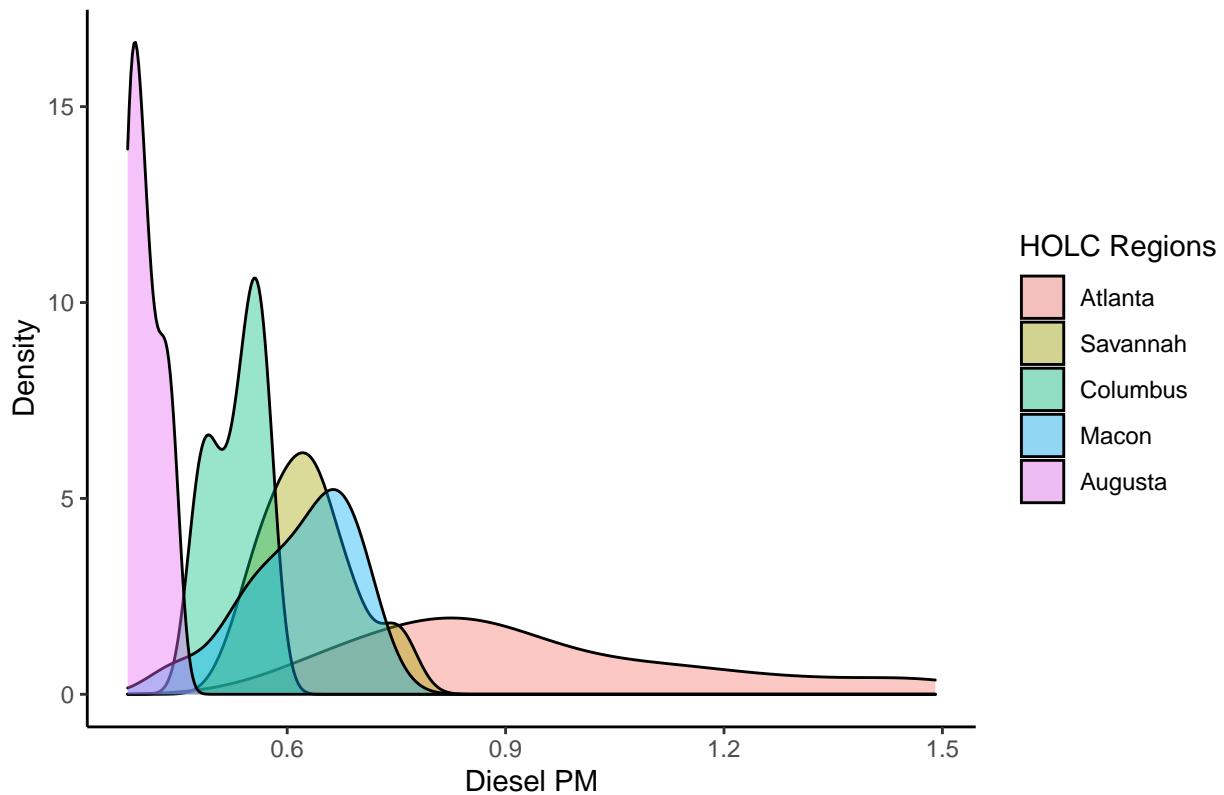
cancer

Cancer Risk per City in Georgia



ds1pm

NATA Diesel PM Density per City in Georgia



Global Moran's I and Spatial Durbin Model

Create neighbors

Creating queen contiguity neighbors among each city (spatial islands)

```
#create neighbors GA
full_data_georgia <- sp.na.omit(full_data_georgia)
gnb <- poly2nb(full_data_georgia)
g_listw <- nb2listw(gnb, style = 'W')

#create neighbors atlanta

full_data_atlanta <- sp.na.omit(full_data_atlanta)
anb <- poly2nb(full_data_atlanta)
a_listw <- nb2listw(anb, style = 'W')

#create neighbors macon
full_data_macon$RESP <- na.omit(full_data_macon$RESP)
full_data_macon$DSLPM <- na.omit(full_data_macon$DSLPM)
full_data_macon$CANCER <- na.omit(full_data_macon$CANCER)
mnb <- poly2nb(full_data_macon)
m_listw <- nb2listw(mnb, style = 'W')
```

```

#create neighbors columbus
full_data_columbus$RESP <- na.omit(full_data_columbus$RESP)
full_data_columbus$DSLPM <- na.omit(full_data_columbus$DSLPM)
full_data_columbus$CANCER <- na.omit(full_data_columbus$CANCER)
cnb <- poly2nb(full_data_columbus)
c_listw <- nb2listw(cnb, style = 'W')

#create neighbors augusta
full_data_augusta$RESP <- na.omit(full_data_augusta$RESP)
full_data_augusta$DSLPM <- na.omit(full_data_augusta$DSLPM)
full_data_augusta$CANCER <- na.omit(full_data_augusta$CANCER)
aanb <- poly2nb(full_data_augusta)
aa_listw <- nb2listw(aanb, style = 'W')

#create neighbors savannah
full_data_savannah$RESP <- na.omit(full_data_savannah$RESP)
full_data_savannah$DSLPM <- na.omit(full_data_savannah$DSLPM)
full_data_savannah$CANCER <- na.omit(full_data_savannah$CANCER)
snb <- poly2nb(full_data_savannah)
s_listw <- nb2listw(snb, style = 'W')

```

A spatial regression: Georgia

Determining whether there is evidence of *any* clustering as justification for subsequent spatial analyses.

```

#specify dependent variables for lm
outcomes<- c("RESP", "DSLPM", "CANCER")

#for loop:
for(i in 1:length(outcomes)){
model <- paste("model",i, sep="")
m <- lm(as.formula(paste(outcomes[i],"~ HRS10")), data=full_data_georgia)
assign(model,m)
}

#RESP
summary(model1)
#sig and positively associated
AIC(model1)

summary(model2)
#not associated

summary(model3)
#sig and positively associated

#RESP
lm.morantest(model1, listw = g_listw, zero.policy = T)
#morans: 0.933920024      , p: <2.2 e-16

#DSLPM
lm.morantest(model2, listw = g_listw, zero.policy = T)

```

```
#morans: 0.897474442      , p: <2.2 e-16

#CANCER
lm.morantest(model3, listw = g_listw, zero.policy = T)
#morans: 0.933282563      , p: <2.2 e-16
```

Spatial regression: Spatial Durbin Model (SDM) among all cities

```
#for loop:
for(i in 1:length(outcomes)){
  SDMmodel <- paste("SDMmodel", i, sep="")
  m <- lagsarlm(as.formula(paste(outcomes[i], "~ HRS10")),
                 data=full_data_georgia, listw = g_listw, Durbin=T)
  assign(SDMmodel, m)
}

#RESP
summary(SDMmodel1)

#DSLPM
summary(SDMmodel2)

#CANCER
summary(SDMmodel3)

#RESP
impacts(SDMmodel1, listw = g_listw)

#DSLPM
impacts(SDMmodel2, listw = g_listw)

#CANCER
impacts(SDMmodel3, listw = g_listw)
```

Aspatial regression: Atlanta

```
#for loop:
for(i in 1:length(outcomes)){
  model <- paste("model", i, sep="")
  m <- lm(as.formula(paste(outcomes[i], "~ HRS10")), data=full_data_atlanta)
  assign(model, m)
}

summary(model1)
#sig and positively associated
```

```

AIC(model1)

summary(model2)
#non-sig and positively associated

summary(model3)
#sig and positively associated

#RESP
lm.morantest(model1, listw = a_listw, zero.policy = T)
#morans: 0.710699601, p: <2.2 e-16

#DSLPM
lm.morantest(model2, listw = a_listw, zero.policy = T)
#morans: 0.810299413, p: <2.2 e-16

#CANCER
lm.morantest(model3, listw = a_listw, zero.policy = T)
#morans: 0.684958693, p: <2.2 e-16

```

Spatial regression: SDM in Atlanta

```

#for loop:
for(i in 1:length(outcomes)){
  SDMmodel <- paste("SDMmodel",i, sep="")
  m <- lagsarlm(as.formula(paste(outcomes[i],"~ HRS10")),
    data=full_data_atlanta, listw = a_listw, Durbin=T)
  assign(SDMmodel,m)
}

#RESP
summary(SDMmodel1)

#DSLPM
summary(SDMmodel2)

#CANCER
summary(SDMmodel3)

#RESP
impacts(SDMmodel1, listw = a_listw)

#DSLPM
impacts(SDMmodel2, listw = a_listw)

#CANCER
impacts(SDMmodel3, listw = a_listw)

```

A spatial regression: Macon

```
#for loop:  
for(i in 1:length(outcomes)){  
model <- paste("model",i, sep="")  
m <- lm(as.formula(paste(outcomes[i],"~ HRS10")), data=full_data_macon)  
assign(model,m)  
}  
  
summary(model1)  
#not associated  
AIC(model1)  
  
summary(model2)  
#not associated  
  
summary(model3)  
#not associated  
  
#RESP  
lm.morantest(model1, listw = m_listw, zero.policy = T)  
#morans: 0.36166334 , p: 0.006558  
  
#DSLPM  
lm.morantest(model2, listw = m_listw, zero.policy = T)  
#morans: 0.41996037, p: 0.002542  
  
#CANCER  
lm.morantest(model3, listw = m_listw, zero.policy = T)  
#morans: 0.33689573 , p: 0.009534
```

Spatial regression: SDM in Macon

```
#for loop:  
for(i in 1:length(outcomes)){  
SDMmodel <- paste("SDMmodel",i, sep="")  
m <- lagsarlm(as.formula(paste(outcomes[i],"~ HRS10")),  
                data=full_data_macon, listw = m_listw, Durbin=T)  
assign(SDMmodel,m)  
}  
  
#RESP  
summary(SDMmodel1)  
  
#DSLPM  
summary(SDMmodel2)  
  
#CANCER  
summary(SDMmodel3)  
  
#RESP
```

```

impacts(SDMmodel1, listw = m_listw)

#DSLPM
impacts(SDMmodel2, listw = m_listw)

#CANCER
impacts(SDMmodel3, listw = m_listw)

```

A spatial regression: Columbus

```

#for loop:
for(i in 1:length(outcomes)){
  model <- paste("model", i, sep="")
  m <- lm(as.formula(paste(outcomes[i], "~ HRS10")), data=full_data_columbus)
  assign(model, m)
}

summary(model1)
#not associated
AIC(model1)

summary(model2)
#not associated

summary(model3)
#not associated

#RESP
lm.morantest(model1, listw = c_listw, zero.policy = T)
#morans: 0.19419672 , p: 0.01714

#DSLPM
lm.morantest(model2, listw = c_listw, zero.policy = T)
#morans: 0.50627630, p: 1.037e-05

#CANCER
lm.morantest(model3, listw = c_listw, zero.policy = T)
#morans: 0.17682483 , p: 0.02288

```

Spatial regression: SDM in Columbus

```

#for loop:
for(i in 1:length(outcomes)){
  SDMmodel <- paste("SDMmodel", i, sep="")
  m <- lagsarlm(as.formula(paste(outcomes[i], "~ HRS10")),
                 data=full_data_columbus, listw = c_listw, Durbin=T)
  assign(SDMmodel, m)
}

```

```

#RESP
summary(SDMmodel1)

#DSLPM
summary(SDMmodel2)

#CANCER
summary(SDMmodel3)

#RESP
impacts(SDMmodel1, listw = c_listw)

#DSLPM
impacts(SDMmodel2, listw = c_listw)

#CANCER
impacts(SDMmodel3, listw = c_listw)

```

A spatial regression: Augusta

```

#for loop:
for(i in 1:length(outcomes)){
model <- paste("model",i, sep="")
m <- lm(as.formula(paste(outcomes[i]," ~ HRS10")), data=full_data_augusta)
assign(model,m)
}

summary(model1)
#not associated
AIC(model1)

summary(model2)
#not associated

summary(model3)
#sig and positively associated

#RESP
lm.morantest(model1, listw = aa_listw, zero.policy = T)
#morans: 0.26835958 , p: 0.003272

#DSLPM
lm.morantest(model2, listw = aa_listw, zero.policy = T)
#morans: 0.36258233, p: 0.0003841

#CANCER
lm.morantest(model3, listw = aa_listw, zero.policy = T)
#morans: 0.26876071 , p: 0.003245

```

Spatial regression: SDM in Augusta

```
#for loop:
for(i in 1:length(outcomes)){
SDMmodel <- paste("SDMmodel",i, sep="")
m <- lagsarlm(as.formula(paste(outcomes[i],"~ HRS10")),
               data=full_data_augusta, listw = aa_listw, Durbin=T)
assign(SDMmodel,m)
}

#RESP
summary(SDMmodel1)

#DSLPM
summary(SDMmodel2)

#CANCER
summary(SDMmodel3)

#RESP
impacts(SDMmodel1, listw = aa_listw)

#DSLPM
impacts(SDMmodel2, listw = aa_listw)

#CANCER
impacts(SDMmodel3, listw = aa_listw)
```

Aspatial regression: Savannah

```
#for loop:
for(i in 1:length(outcomes)){
model <- paste("model",i, sep="")
m <- lm(as.formula(paste(outcomes[i],"~ HRS10")), data=full_data_savannah)
assign(model,m)
}

summary(model1)
#not associated
AIC(model1)

summary(model2)
#not associated

summary(model3)
#sig and positively associated

#RESP
lm.morantest(model1, listw = s_listw, zero.policy = T)
#morans: 0.55149243 , p: 9.7e-07
```

```

#DSLPM
lm.morantest(model2, listw = s_listw, zero.policy = T)
#morans: 0.73040748, p: 4.246e-10

#CANCER
lm.morantest(model3, listw = s_listw, zero.policy = T)
#morans: 0.42231295 , p: 8.289e-05

```

Spatial regression: SDM in Savannah

```

#for loop:
for(i in 1:length(outcomes)){
SDMmodel <- paste("SDMmodel",i, sep="")
m <- lagsarlm(as.formula(paste(outcomes[i],"~ HRS10")),
               data=full_data_savannah, listw = s_listw, Durbin=T)
assign(SDMmodel,m)
}

#RESP
summary(SDMmodel1)

#DSLPM
summary(SDMmodel2)

#CANCER
summary(SDMmodel3)

#RESP
impacts(SDMmodel1, listw = s_listw)

#DSLPM
impacts(SDMmodel2, listw = s_listw)

#CANCER
impacts(SDMmodel3, listw = s_listw)

```

Results

Spatial dependence (Global Moran's I) was significant in all scenarios

Higher redlining was associated with an increase in the total impact of all health hazards in all HOLC regions, with the exception of diesel PM in Columbus and Augusta