

Spatial Data Analysis of Grocery Store Locations in Chicago

Sam Song

Introduction

Food Apartheid in Chicago

Food Apartheid, a term initially coined by food justice activist Karen Washington, refers to a systemic segregation that perpetuates disparities in access to nutritious food, rooted in systemic injustices. While sharing similarities with the concept of food deserts, the term food apartheid is gaining prominence as it better encapsulates the structural injustices and inequities in food access faced by low-income communities and communities of color. Unlike food deserts, which merely describe geographical areas with limited access to healthy food, food apartheid acknowledges the deep-seated history of racial discrimination and injustice.

Chicago, despite being the third largest city in the United States, grapples with severe food apartheid issues, with one in five households in the Chicago area experiencing food insecurity, as reported by the Greater Chicago Food Depository. This problem is particularly acute in the community areas of the south side of Chicago, where a majority of residents are African American. Insufficient access to grocery stores exacerbates food insecurity in these areas, with existing stores dwindling in number. The presence of a grocery store in a community area plays a pivotal role in food accessibility, offering a diverse array of nutritious groceries, including fresh produce, meat, deli items, and packaged goods, all essential components of a healthy diet.

In this analysis, I focus on the distribution of grocery store locations in Chicago and their potential relationship with demographic factors such as race and socioeconomic status. To address the main question of which areas of Chicago are affected by food apartheid and the unique characteristics of these areas, I computed Moran's I to assess the spatial autocorrelation of grocery store locations. Additionally, I conducted spatial regression using spatial autoregressive (SAR) models to examine the relationship between grocery store locations and various independent factors while accounting for spatial effects.

```
library(dplyr)
library(stringr)
library(tidyverse)
library(sf)
library(tmap)
library(spdep)
library(spatialreg)
library(kableExtra)
```

```
# read Chicago Community Boundary data (source: https://data.cityofchicago.org/Facilities-Geographic-Bo
chicago_sf <- st_read("~/Documents/Data/ChicagoCA/chicagoCA.shp") %>%
  select(2, 6, 8:10) %>%
  rename(ComAreaID = area_num_1) %>%
  mutate(ComAreaID = as.numeric(ComAreaID))
```

```
## Reading layer 'chicagoCA' from data source
##   '/Users/song8/Documents/Data/ChicagoCA/chicagoCA.shp' using driver 'ESRI Shapefile'
```

```

## Simple feature collection with 77 features and 9 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -87.94011 ymin: 41.64454 xmax: -87.52414 ymax: 42.02304
## Geodetic CRS: WGS84(DD)

# read Chicago demographic data (source: https://www.cmap.illinois.gov/data/data-hub)
chicago_census <- read_csv("~/Documents/Data/CMAP_2022/cds_202207/ReferenceCCAPProfiles20162020.csv") %>%
  select(GEOID, GEOG, `2020_POP`, WHITE, ASIAN, BLACK, HISP, OTHER, UNEMP, NO_VEH, MEDINC, INCPERCAP, INC_LT_25K, TRANSIT_LOW_PCT, WALKABLE_LOW_PCT) %>%
  rename(ComAreaID = GEOID,
         community = GEOG,
         Pop_2020 = `2020_POP`,
         White = WHITE,
         Asian = ASIAN,
         Hispanic = HISP,
         Black = BLACK,
         Other = OTHER,
         Unemployed = UNEMP,
         No_vehicle = NO_VEH,
         Med_income = MEDINC,
         Per_cap_income = INCPERCAP,
         Income_under_25K = INC_LT_25K,
         Pct_bad_transit = TRANSIT_LOW_PCT,
         Pct_not_walkable = WALKABLE_LOW_PCT) %>%
  mutate(Pct_white = round(White / Pop_2020 * 100, 2),
         Pct_asian = round(Asian / Pop_2020 * 100, 2),
         Pct_black = round(Black / Pop_2020 * 100, 2),
         Pct_hispanic = round(Hispanic / Pop_2020 * 100, 2),
         Pct_other = round(Other / Pop_2020 * 100, 2),
         Pct_unemployed = round(Unemployed / Pop_2020 * 100, 2),
         Pct_poverty = round(Income_under_25K / Pop_2020 * 100, 2),
         Pct_no_vehicle = round(No_vehicle / Pop_2020 * 100, 2),
         Pct_bad_transit = round(Pct_bad_transit, 3),
         Pct_not_walkable = round(Pct_not_walkable, 3),
         Med_income = Med_income * 1,
         Per_cap_income = Per_cap_income * 1,
         Pct_black = ifelse(Pct_black > 100, round((Black)/(Black + White + Asian + Hispanic + Other) * 100, 2), Pct_black))

# join the selected variables above onto the first dataset by community ID
chicago_sf <- chicago_sf %>%
  inner_join(chicago_census, by = c("ComAreaID"= "ComAreaID")) %>%
  select(-5) %>%
  rename(community = community.x)

# read the third dataset about the grocery store in Chicago (source: https://data.cityofchicago.org/Health-and-Human-Resources/Chicago-Grocery-Stores/8489-8687-7203-7203)
grocery_store <- read_csv("~/Documents/Data/grocery_chicago.csv")

grocery_store <- grocery_store %>%
  # drop rows with missing geometry information
  filter(is.na(Location) == FALSE) %>%
  # extract latitude and longitude from the string
  mutate(x = str_split(Location, " ", simplify = TRUE)[,2],

```

```

y = str_split(Location, " ", simplify = TRUE)[,3],
# convert the extracted value to numeric
x = as.numeric(str_replace_all(x, "\\(", "")),
y = as.numeric(str_replace_all(y, "\\)", "")) %>%
select(- Location, - `Last updated`) %>%
rename(status = `New status`,
Chain = `Store Name`) %>%
# filter out the online-only store as there is only one value
filter(status != 'ONLINE ORDERS ONLY',
status != 'CLOSED') %>%
# transform the dataset to a sf object
st_as_sf(coords = c("x", "y"), remove = FALSE) %>%
# assign the Coordinate Reference System (WGS 84)
st_set_crs(4236)

# transform the Coordinate Reference System to match that of the first dataset.
grocery_store <- st_transform(grocery_store, st_crs(chicago_sf))

# find grocery stores within each neighborhood
grocery_nb <- st_join(grocery_store, chicago_sf, join = st_within) %>%
filter(is.na(ComAreaID) == FALSE)

# count number of grocery stores
grocery_nb_cnt <- as_tibble(grocery_nb) %>%
count(ComAreaID)

# join the grocery counts onto the original dataset
chicago_sf <- left_join(chicago_sf, grocery_nb_cnt) %>%
rename(num_grocery = n)

# finalize data preparation
chicago_sf <- chicago_sf %>%
# make sure there is no NAs by turning missing values to 0
mutate(num_grocery = ifelse(is.na(num_grocery), 0 , num_grocery),
# create a column that shows the number of grocery stores per 100,000 residents
grocery_100k = num_grocery/Pop_2020 * 100000)

```

Spatial Autocorrelation

Exploratory Data Analysis

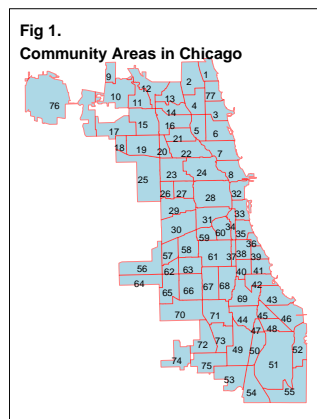
```

color_status = c("OPEN" = "#228B22",
"CLOSED" = "#EE4B2B")

tmap_mode("plot")

```

```
tm_shape(chicago_sf) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "lightblue") +
  tm_text("ComAreaID",
    size = .6,
    fontface = "bold",
    xmod = -.1,
    ymod = -.2) +
tm_layout(title = "Fig 1.\nCommunity Areas in Chicago",
  inner.margins = c(.05, .05, .12, .05),
  title.fontface = "bold",
  title.size = 1)
```



```
# List of Community Areas of Chicago
table <- chicago_sf %>%
  select(1:2) %>%
  arrange(ComAreaID) %>%
  st_drop_geometry() %>%
  rename(ID = ComAreaID,
    Name = community) %>%
  head()
table
```

```
##   ID      Name
## 1  1  ROGERS PARK
## 2  2   WEST RIDGE
## 3  3    UPTOWN
## 4  4 LINCOLN SQUARE
## 5  5   NORTH CENTER
## 6  6    LAKE VIEW
```

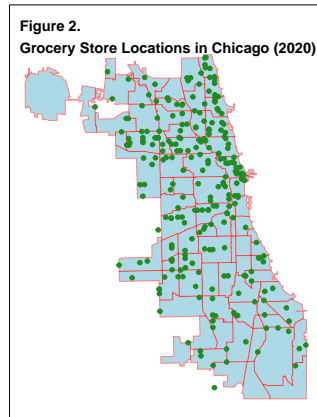
To begin with, I created a map of the community areas of Chicago. There are a total of 77 community areas with each area surrounded by red borderlines. The names of community areas corresponding to the ID can be found in the table.

```
# Grocery store location
tm_shape(chicago_sf) +
  tm_borders(col = "red", alpha = .5) +
```

```

tm_polygons(col = "lightblue") +
tm_shape(grocery_store) +
tm_dots(col = "#228B22",
        size = .1,
        palette = color_status) +
tm_layout(title = "Figure 2.\nGrocery Store Locations in Chicago (2020)",
          inner.margins = c(.05, .05, .12, .05),
          title.fontface = "bold",
          title.size = 1)

```



```

table2 <- chicago_sf %>%
  select(ComAreaID, community, num_grocery) %>%
  arrange(desc(num_grocery)) %>%
  rename(ID = ComAreaID,
         Name = community,
         `Number of Grocery Stores` = num_grocery) %>%
  st_drop_geometry() %>%
  head()
table2

```

##	ID	Name	Number of Grocery Stores
## 1	8	NEAR NORTH SIDE	15
## 2	19	BELMONT CRAGIN	13
## 3	22	LOGAN SQUARE	12
## 4	28	NEAR WEST SIDE	9
## 5	6	LAKE VIEW	9
## 6	7	LINCOLN PARK	8

```

more_than_10 <- chicago_sf %>%
  filter(num_grocery >= 10)

zero <- chicago_sf %>%
  filter(num_grocery == 0)

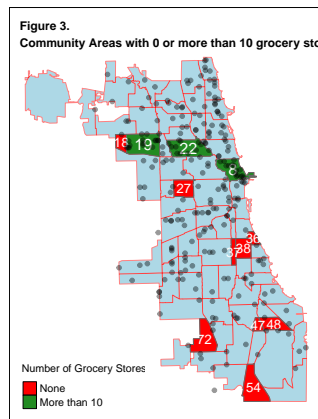
tm_shape(chicago_sf) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "lightblue") +
  tm_shape(more_than_10) +

```

```

tm_polygons(col = "#228B22") +
tm_text("ComAreaID") +
tm_shape(zero) +
tm_polygons(col = "red") +
tm_text("ComAreaID", size = .8) +
tm_shape(grocery_store) +
tm_dots(size = .1, alpha = .4) +
tm_layout(title = "Figure 3.\nCommunity Areas with 0 or more than 10 grocery stores",
           inner.margins = c(.05, .05, .12, .05),
           title.fontface = "bold",
           title.size = .8) +
tm_add_legend(title = "Number of Grocery Stores",
              labels = c("None",
                         "More than 10"),
              col = c("red", "#228B22"))

```



After creating the map of Chicago, I proceeded to plot the locations of grocery stores across the city in Figure 2, with each dot representing a grocery store location. A visual inspection of Figure 2 already suggests a higher concentration of grocery stores in the north side of Chicago compared to the south side. Table 2 below provides a breakdown of community areas alongside the corresponding number of grocery stores in each area. While several areas boast more than 10 grocery stores, there are also community areas devoid of any such establishments. In Figure 3, I filtered the community areas to highlight those with either more than 10 (depicted in green) or zero (depicted in red) grocery stores. This visualization underscores the disparities in grocery store distribution across community areas, particularly noting that those areas shaded in red predominantly lie in the south side of the city. However, it's crucial to refrain from drawing definitive conclusions based solely on this map, as it merely represents a count of grocery stores in each area without accounting for various other factors. For instance, although both areas 18 and 54 register zero grocery stores, the accessibility to grocery stores may vary significantly between them. This discrepancy could be attributed to several grocery stores situated near the border of areas 18 and 19, potentially offering better accessibility to residents in area 54 compared to those in area 18. Hence, it's imprudent to assume the same level of accessibility to grocery stores across all nine red community areas.

Moran's I

Based on the preliminary visualizations, there appears to be a tendency for values in close proximity to one another to exhibit similarity, as observed with the number of grocery stores in each community area. Recognizing that the spatial distribution of grocery stores does not display a completely random pattern, I opted to quantify this spatial pattern or clustering by computing Moran's I statistic.

```

# create neighbors
chicago_nb <- poly2nb(chicago_sf, queen = TRUE)
# Create neighbor weights
chicago_nbw <- nb2listw(chicago_nb, style = "W", zero.policy = TRUE)
# Check if zero policy attribute says "TRUE":
attr(chicago_nbw, "zero.policy")

```

```
## [1] TRUE
```

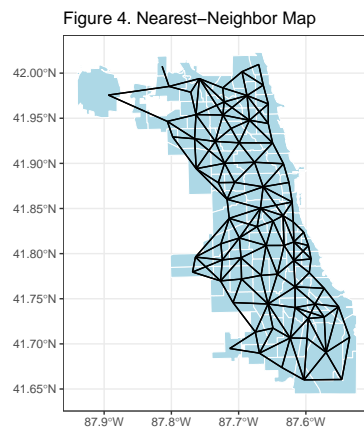
```

# measures the center point of each neighborhood
chicago_centroids <- chicago_sf %>%
  st_centroid() %>%
  st_coordinates()

# create a sf of neighbors
neighbors_sf <- nb2lines(chicago_nb,
                        coords = chicago_centroids,
                        as_sf = TRUE) %>%
  st_set_crs(st_crs(chicago_sf))

# plot the neighborhoods
ggplot(chicago_sf) +
  geom_sf(color = "white", fill = "lightblue") +
  geom_sf(data = neighbors_sf) +
  theme_bw() +
  labs(title = "Figure 4. Nearest-Neighbor Map")

```



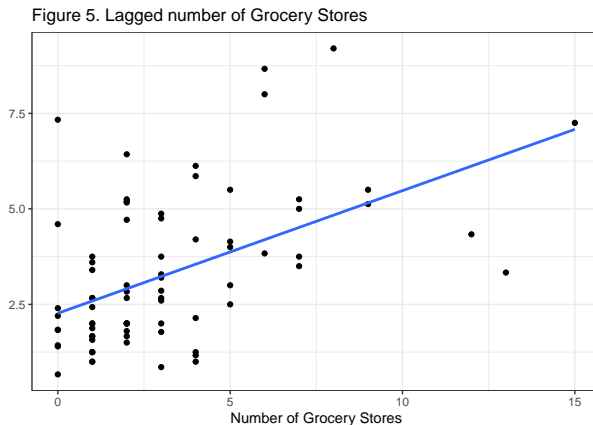
The Moran's I statistic serves as the correlation coefficient measuring the relationship between a variable, such as the number of grocery stores, and its neighboring values. However, before computing this correlation, it's essential to define the neighbors. While various methods exist for constructing a list of neighbors, I employed the `poly2nb` function, which generates a neighbors list based on regions with contiguous boundaries, indicating the sharing of one or more boundary points. Subsequently, spatial weights are added to this neighbors list, a crucial step aimed at normalizing the Moran's I statistic. This normalization ensures that the range of possible Moran's I values falls within -1 and 1, facilitating meaningful interpretation.

```

# create lagged value for the number of grocery stores in each community area of Chicago
chicago_sf$num_grocery_lag <- lag.listw(chicago_nbw, chicago_sf$num_grocery, zero.policy = TRUE)

```

```
# display the relationship between X and X_lagged
ggplot(chicago_sf) +
  geom_point(aes(x = num_grocery, y = num_grocery_lag)) +
  geom_smooth(aes(x = num_grocery, y = num_grocery_lag), method = "lm", se = FALSE) +
  labs(title = "Figure 5. Lagged number of Grocery Stores", x = "Number of Grocery Stores", y = "") +
  theme_bw()
```



```
# calculate Moran's I statistic
lm(num_grocery_lag ~ num_grocery, data = chicago_sf) %>%
  summary()

##
## Call:
## lm(formula = num_grocery_lag ~ num_grocery, data = chicago_sf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.1069 -1.0166 -0.3729  0.6489  5.0663
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.26701    0.27986   8.100 7.68e-12 ***
## num_grocery  0.32102    0.06381   5.031 3.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.69 on 75 degrees of freedom
## Multiple R-squared:  0.2523, Adjusted R-squared:  0.2423
## F-statistic: 25.31 on 1 and 75 DF, p-value: 3.248e-06
```

Once the neighbors list is established and the weights are determined, we can calculate the aggregated values for each neighborhood, representing the total number of grocery stores in the community area. This aggregated value is known as a spatially lagged value (x_{lag}). Utilizing the number of grocery stores in each community area of Chicago, as computed in the setup code chunk above, I generated a plot illustrating the summarized neighborhood value of the number of grocery stores (X_{lag}) against the number of grocery stores for each county (X). The Moran's I coefficient between X_{lag} and X represents the slope of the least squares regression line that best fits the points after normalizing the spread between both sets of data, a computation performed via linear regression.

There is a slightly more convenient way to compute the Moran's I statistic, which is to use a built-in `moran.test` function that would conveniently return the statistic. Steps are as follows:

```
num_grocery.moranI <- moran(chicago_sf$num_grocery,
                             chicago_nbw,
                             n = length(chicago_nbw),
                             S0 = Szero(chicago_nbw),
                             NAOK = TRUE)
# return Moran's statistic
moran.test(chicago_sf$num_grocery, chicago_nbw, zero.policy = TRUE)

##
## Moran I test under randomisation
##
## data:  chicago_sf$num_grocery
## weights: chicago_nbw
##
## Moran I statistic standard deviate = 4.7576, p-value = 9.796e-07
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.321015134      -0.013157895      0.004933677
```

The findings from both the linear regression and `moran.test` indicate a consistent Moran's I value of 0.287. Despite the relatively weak strength of this relationship, it suggests the presence of positive spatial autocorrelation. When there is minimal association between X_{lag} and X , the slope tends to approach flat, consequently yielding a Moran's I value close to 0.

Significance Test

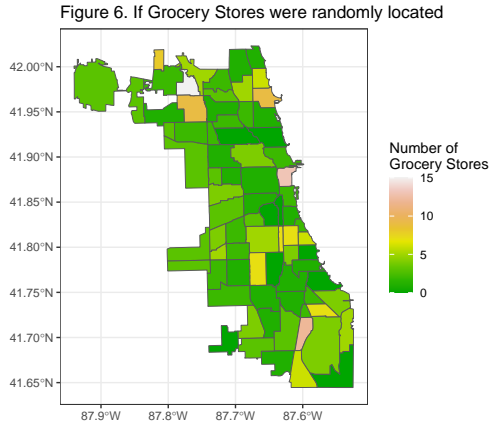
With a Moran's I value of 0.287 calculated, the next step is to assess its significance. To achieve this, I conducted a Monte Carlo test, wherein attribute values (in this case, the number of grocery stores) are randomly assigned to community areas within the dataset. For each permutation of attribute values, a Moran's I value is computed, resulting in a sampling distribution of Moran's I values under the Null Hypothesis, which assumes that attribute values are randomly distributed across Chicago. The observed Moran's I value is then compared to this sampling distribution. Below are the null and alternative hypotheses for this significance testing.

H_0 : There is **NO** spatial autocorrelation, I is close to 0

H_A : There **IS** spatial autocorrelation, $I \neq 0$.

```
# Null Hypothesis
chicago_sf$rand_grocery <- sample(chicago_sf$num_grocery, length(chicago_sf$num_grocery), replace = FALSE)

ggplot(chicago_sf) +
  geom_sf(aes(fill = rand_grocery)) +
  scale_fill_gradientn(colours = terrain.colors(10)) +
  labs(title = "Figure 6. If Grocery Stores were randomly located",
       fill = "Number of\nGrocery Stores") +
  theme_bw()
```



```
# Moran's I under the Null Hypothesis
moran(chicago_sf$rand_grocery, listw = chicago_nbw, S0 = Szero(chicago_nbw), n = length(chicago_nbw), z

## $I
## [1] 0.0009642338
##
## $K
## [1] 6.277104

# Monte-Carlo test for Moran's I:
moran.mc(chicago_sf$num_grocery,
  listw = chicago_nbw,
  nsim = 499,
  zero.policy = TRUE)

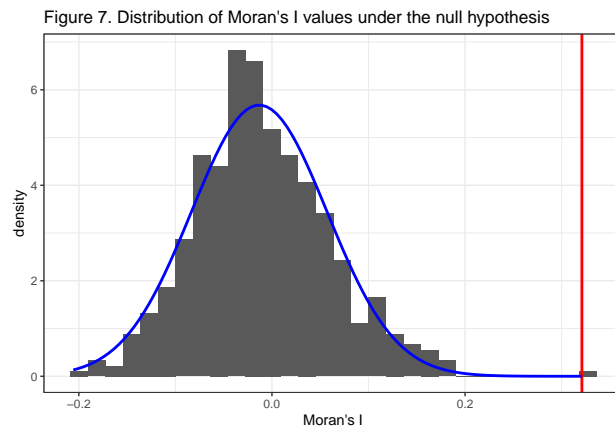
##
## Monte-Carlo simulation of Moran I
##
## data:  chicago_sf$num_grocery
## weights:  chicago_nbw
## number of simulations + 1: 500
##
## statistic = 0.32102, observed rank = 500, p-value = 0.002
## alternative hypothesis: greater
```

The last step is to create a visualization of 499 sampling distribution of simulated Moran's I values in histogram and see where the observed Moran's I value of 0.287 lies.

```
# normal distribution of Moran's I value from Moran I test under randomization
num_grocery_m_norm <- moran.test(chicago_sf$num_grocery,
  listw = chicago_nbw,
  zero.policy = TRUE)

# Monte-Carlo simulation of Moran I
num_grocery_mc <- moran.mc(chicago_sf$num_grocery,
  listw = chicago_nbw,
  nsim = 499,
  zero.policy = TRUE)
```

```
# Histogram of MC value from 499 simulations (randomized) with normal distribution overlaid
ggplot() +
  geom_histogram(aes(x = num_grocery_mc$res, after_stat(density))) +
  geom_vline(xintercept = num_grocery_mc$statistic, color = "red", size = 1) +
  geom_function(fun = function(x) dnorm(x, num_grocery_m_norm$estimate[2],
                                         sqrt(num_grocery_m_norm$estimate[3])),
               color = "blue", size = 1) +
  theme_bw() +
  labs(x = "Moran's I",
       title = "Figure 7. Distribution of Moran's I values under the null hypothesis")
```



The histogram illustrates that the observed value of 0.287 deviates significantly from what one would anticipate if the distribution of the number of grocery stores across each community area of Chicago were random. Furthermore, with a p-value of 0.002, we can confidently reject the null hypothesis, leading us to conclude that there exists spatial autocorrelation in the number of grocery stores among the community areas of Chicago.

Spatial Regression

To delve deeper into the potential associations between grocery store locations and other features, I incorporated a spatial regression component, which briefly introduces the utilization of the SAR (Simultaneous Autoregressive Model). To execute this regression, I employed the `lagsarlm` function, structured as follows:

$$Y = \beta_0 + \beta_1 X + \rho \sum w_i Y_i$$

Here, ρ characterizes the level of correlation with neighboring data points, w_i represents the weight assigned to neighbor i , and β_i denotes the regression coefficients for the variables of interest, akin to those in linear regression. If the ρ value approaches 1, it signifies a pronounced spatial autocorrelation among the variables of interest, necessitating its consideration in the analysis. Conversely, if the ρ value nears 0, it indicates minimal spatial autocorrelation among the variables, enabling reliance on the results of ordinary least squares (linear) regression for analysis purposes.

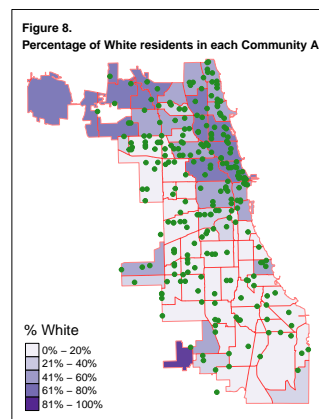
Exploratory Spatial Data Analysis

```

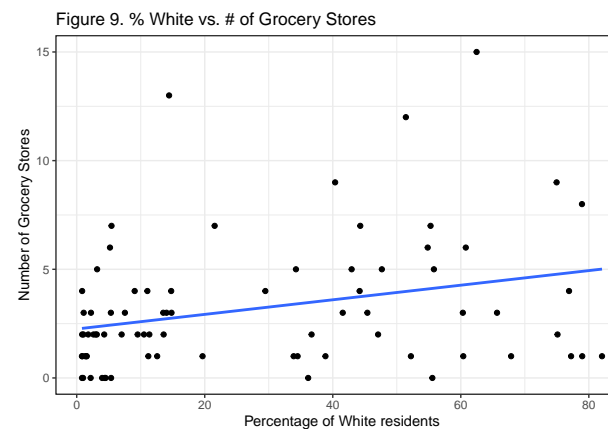
# White population choropleth
tm_shape(chicago_sf) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "Pct_white",
    palette = "Purples",
    legend.show = FALSE) +
  tm_shape(grocery_store) +
  tm_dots(col = "#228B22",
    size = .1,
    palette = color_status,
    legend.show = FALSE) +
  tm_layout(title = "Figure 8.\nPercentage of White residents in each Community Area",
    inner.margins = c(.05, .05, .12, .05),
    title.fontface = "bold",
    title.size = 1) +
  tm_add_legend(title = "% White",
    labels = c("0% - 20%",
      "21% - 40%",
      "41% - 60%",
      "61% - 80%",
      "81% - 100%"),
    col = RColorBrewer::brewer.pal(5, "Purples"))

ggplot(chicago_sf) +
  geom_point(aes(x = Pct_white, y = num_grocery)) +
  geom_smooth(aes(x = Pct_white, y = num_grocery), se = FALSE, method = "lm") +
  labs(x = "Percentage of White residents",
    y = "Number of Grocery Stores",
    title = "Figure 9. % White vs. # of Grocery Stores") +
  theme_bw()

```



Racial Factors



```

# African American population choropleth
tm_shape(chicago_sf) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "Pct_black",
    palette = "Purples",
    legend.show = FALSE) +

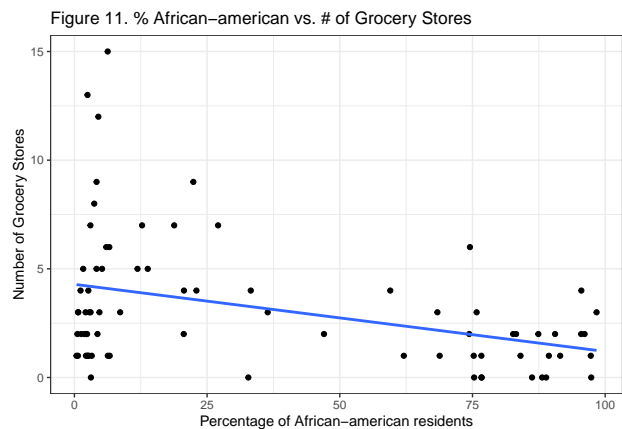
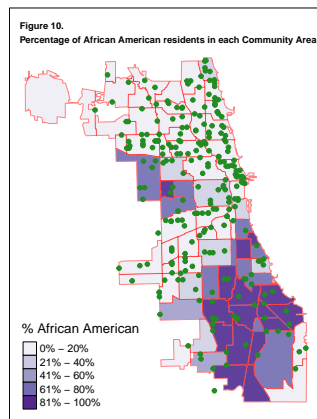
```

```

tm_shape(grocery_store) +
tm_dots(col = "#228B22",
        size = .1,
        palette = color_status,
        legend.show = FALSE) +
tm_layout(title = "Figure 10.\nPercentage of African American residents in each Community Area",
           inner.margins = c(.05, .05, .12, .05),
           title.fontface = "bold",
           title.size = .8) +
tm_add_legend(title = "% African American",
              labels = c("0% - 20%",
                         "21% - 40%",
                         "41% - 60%",
                         "61% - 80%",
                         "81% - 100%"),
              col = RColorBrewer::brewer.pal(5, "Purples"))

ggplot(chicago_sf) +
  geom_point(aes(x = Pct_black, y = num_grocery)) +
  geom_smooth(aes(x = Pct_black, y = num_grocery), se = FALSE, method = "lm") +
  labs(x = "Percentage of African-american residents",
       y = "Number of Grocery Stores",
       title = "Figure 11. % African-american vs. # of Grocery Stores") +
  theme_bw()

```



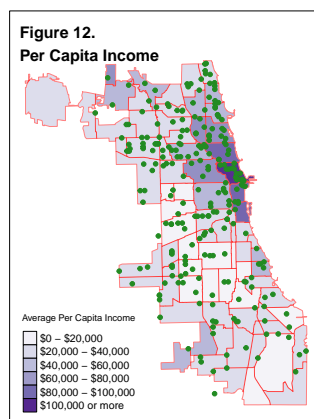
I specifically chose to analyze data regarding White and African American populations to maintain conciseness in this analysis, considering their starkly contrasting distribution across community areas. Notably, Figure 8 and Figure 10 highlight distinct spatial patterns: White residents predominantly inhabit the northern Chicago regions, constituting over 40% of the total population in these areas, while African American residents cluster predominantly in the southern parts, accounting for 60% to 80% of the population in those regions. This observation suggests a notable spatial correlation in residents' racial composition, wherein individuals of the same race tend to reside in closer proximity, as illustrated in the aforementioned figures. Upon examining the scatter plots in Figure 9 and 11, a subtle positive linear relationship between the number of grocery stores and the percentage of White residents, and a slight negative linear association with the percentage of African American residents in each community area, become apparent. Nonetheless, it's imperative to conduct significance testing before drawing any definitive conclusions.

```

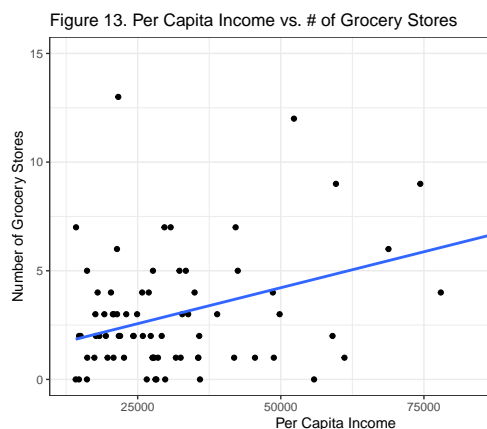
# per capita income
tm_shape(chicago_sf) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "Per_cap_income",
    palette = "Purples",
    legend.show = FALSE) +
  tm_shape(grocery_store) +
  tm_dots(col = "#228B22",
    size = .1,
    palette = color_status,
    legend.show = FALSE) +
  tm_layout(title = "Figure 12.\nPer Capita Income",
    inner.margins = c(.05, .05, .12, .05),
    title.fontface = "bold",
    title.size = 1) +
  tm_add_legend(title = "Average Per Capita Income",
    labels = c("$0 - $20,000",
      "$20,000 - $40,000",
      "$40,000 - $60,000",
      "$60,000 - $80,000",
      "$80,000 - $100,000",
      "$100,000 or more"),
    col = RColorBrewer::brewer.pal(6, "Purples"))

ggplot(chicago_sf) +
  geom_point(aes(x = Per_cap_income, y = num_grocery)) +
  geom_smooth(aes(x = Per_cap_income, y = num_grocery), se = FALSE, method = "lm") +
  labs(x = "Per Capita Income",
    y = "Number of Grocery Stores",
    title = "Figure 13. Per Capita Income vs. # of Grocery Stores") +
  theme_bw()

```



Socioeconomic Factors



```

# people with income less than $25,000
tm_shape(chicago_sf) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "Pct_poverty",
    palette = "Purples",

```

```

        legend.show = FALSE) +
tm_shape(grocery_store) +
  tm_dots(col = "#228B22",
    size = .1,
    palette = color_status,
    legend.show = FALSE) +
tm_layout(title = "Figure 14.\nPercentage of Population with Income less than $25,000",
  inner.margins = c(.05, .05, .12, .05),
  title.fontface = "bold",
  title.size = .9) +
tm_add_legend(title = "% less than $25,000",
  labels = c("0% - 5%",
    "5% - 10%",
    "10% - 15%",
    "15% - 20%",
    "20% - 25%",
    "25% - 30%",
    "30% or higher"),
  col = RColorBrewer::brewer.pal(7, "Purples"))

ggplot(chicago_sf) +
  geom_point(aes(x = Pct_poverty, y = num_grocery)) +
  geom_smooth(aes(x = Pct_poverty, y = num_grocery), se = FALSE, method = "lm") +
  labs(x = "Poverty Rate (income less than $25,000)",
    y = "Number of Grocery Stores",
    title = "Figure 15. Poverty rate vs. # of Grocery Stores") +
  theme_bw()

```

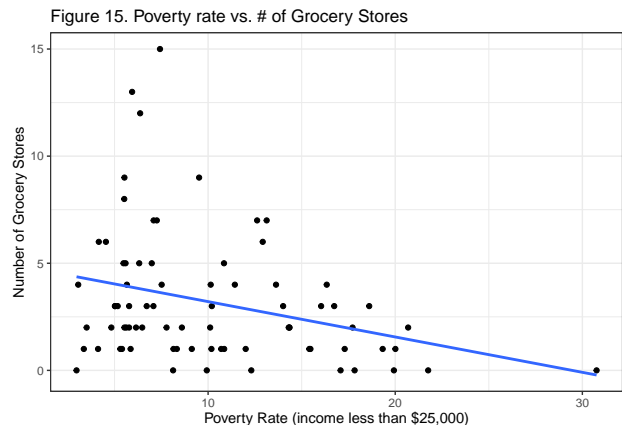
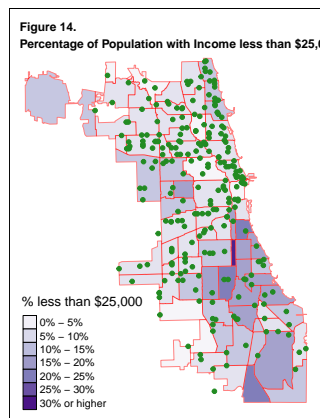


Figure 14 depicts the average per capita income for each community area, revealing a general trend of slightly higher incomes in the northern regions compared to the southern areas of Chicago. Notably, specific neighborhoods in the northeast exhibit considerably higher average per capita incomes, forming distinct clusters within the city. Meanwhile, Figure 15 outlines the poverty rate (individuals earning less than \$25,000 annually) across community areas. It's evident that areas with high poverty rates correspond to fewer grocery stores, with a predominant African American population in these regions. Examining the scatter plots in Figures 13 and 15, a positive linear relationship emerges between the number of grocery stores and per capita income, while a negative association is observed between the number of grocery stores and the poverty rate in each community area. However, it's crucial to refrain from drawing conclusions until significance testing is conducted.

```

tmap_mode("view")

# 1. Total population
pop_tot <- chicago_sf %>%
  select(Pop_2020)
# 2. White population
pct_white <- chicago_sf %>%
  select(Pct_white)
# 3. Asian population
pct_asian <- chicago_sf %>%
  select(Pct_asian)
# 4. African American population
pct_black <- chicago_sf %>%
  select(Pct_black)
# 5. Hispanic population
pct_hispanic <- chicago_sf %>%
  select(Pct_hispanic)
# 6. Population of other race
pct_other <- chicago_sf %>%
  select(Pct_other)
# 7. Unemployment rate
pct_unemployed <- chicago_sf %>%
  select(Pct_unemployed)
# 8. Median income
median_income <- chicago_sf %>%
  select(Med_income)
# 9. Per Capita income
per_capita_income <- chicago_sf %>%
  select(Per_cap_income)
# 10. % income less than $25,000
pct_poverty <- chicago_sf %>%
  select(Pct_poverty)
# 11. % no vehicle
pct_no_vehicle <- chicago_sf %>%
  select(Pct_no_vehicle)

tm_shape(chicago_sf) +
  tm_polygons() +
tm_shape(pop_tot) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "Pop_2020",
    palette = "Purples",
    legend.show = FALSE) +
  tm_shape(pct_white) +
  tm_polygons(col = "Pct_white",
    palette = "Purples",
    legend.show = FALSE) +
  tm_shape(pct_asian) +
  tm_borders(col = "red", alpha = .5) +
  tm_polygons(col = "Pct_asian",
    palette = "Purples",

```



```

        legend.show = FALSE) +
tm_shape(pct_black) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Pct_black",
            palette = "Purples",
            legend.show = FALSE) +
tm_shape(pct_hispanic) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Pct_hispanic",
            palette = "Purples",
            legend.show = FALSE) +
tm_shape(pct_other) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Pct_other",
            palette = "Purples",
            legend.show = FALSE) +
tm_shape(pct_unemployed) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Pct_unemployed",
            palette = "Purples",
            legend.show = FALSE) +
    tm_shape(median_income) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Med_income",
            palette = "Purples",
            legend.show = FALSE) +
tm_shape(per_capita_income) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Per_cap_income",
            palette = "Purples",
            legend.show = FALSE) +
tm_shape(pct_poverty) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Pct_poverty",
            palette = "Purples",
            legend.show = FALSE) +
    tm_shape(pct_no_vehicle) +
tm_borders(col = "red", alpha = .5) +
tm_polygons(col = "Pct_no_vehicle",
            palette = "Purples",
            legend.show = FALSE) +
tm_shape(grocery_store) +
tm_dots(col = "#228B22",
        size = .03,
        legend.show = FALSE)

```

Interactive Map This is an interactive map where the user can change the input of their interests and look into the distribution of demographic factors throughout the city of Chicago, overlaid with the grocery store locations, which could hint at the spatial association between the grocery store locations and other demographic factors that I did not include in this analysis. **Click one variable of interest at a time other than grocery store.**

Ordinary Least Squares Model (Linear Regression)

To test the significance of the independent variables in their relationship with the number of grocery stores in each community area, I started by fitting ordinary least squares regression models first.

1. Number of Grocery stores and the percentage of white residents in each community area

```
# changing the unit of per capita income so that the coefficients are not too small
chicago_sf <- chicago_sf %>%
  mutate(income1000 = Per_cap_income / 1000)

pct_white_lm <- lm(num_grocery ~ Pct_white, data = chicago_sf)
summary(pct_white_lm)
```

```
##
## Call:
## lm(formula = num_grocery ~ Pct_white, data = chicago_sf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.1192 -1.9137 -0.4893  1.2508 10.6488
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.25352    0.48603   4.637 1.47e-05 ***
## Pct_white    0.03358    0.01279   2.626  0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.927 on 75 degrees of freedom
## Multiple R-squared:  0.08419,    Adjusted R-squared:  0.07198
## F-statistic: 6.895 on 1 and 75 DF,  p-value: 0.01047
```

2. Number of Grocery stores and the percentage of African-american residents in each community area

```
pct_black_lm <- lm(num_grocery ~ Pct_black, data = chicago_sf)
summary(pct_black_lm)
```

```
##
## Call:
## lm(formula = num_grocery ~ Pct_black, data = chicago_sf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.1902 -1.9168 -0.5258  0.8746 10.9073
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.286280    0.447768   9.573 1.22e-14 ***
## Pct_black   -0.030881    0.008688  -3.555 0.000659 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.829 on 75 degrees of freedom
## Multiple R-squared:  0.1442, Adjusted R-squared:  0.1328
## F-statistic: 12.63 on 1 and 75 DF,  p-value: 0.0006594
```

3. Number of Grocery stores and the Per capita income (in thousands) in each community area

```
per_cap_income_lm <- lm(num_grocery ~ income1000, data = chicago_sf)
summary(per_cap_income_lm)
```

```
##
## Call:
## lm(formula = num_grocery ~ income1000, data = chicago_sf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.786 -1.850 -0.363   1.121  10.658
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.91187    0.62577   1.457   0.149
## income1000    0.06616    0.01578   4.192 7.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.753 on 75 degrees of freedom
## Multiple R-squared:  0.1898, Adjusted R-squared:  0.179
## F-statistic: 17.58 on 1 and 75 DF,  p-value: 7.474e-05
```

4. Number of Grocery stores and the percentage of residents earning less than \$25,000 annually

```
pct_poverty_lm <- lm(num_grocery ~ Pct_poverty, data = chicago_sf)
summary(pct_poverty_lm)
```

```
##
## Call:
## lm(formula = num_grocery ~ Pct_poverty, data = chicago_sf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.3695 -1.9306 -0.6913   1.0628  11.3665
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.8597    0.6982   6.960 1.1e-09 ***
## Pct_poverty   -0.1650    0.0604  -2.732 0.00784 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.917 on 75 degrees of freedom
## Multiple R-squared:  0.09053,    Adjusted R-squared:  0.07841
## F-statistic: 7.466 on 1 and 75 DF,  p-value: 0.007837
```

In summarizing the outcomes of a sequence of linear regressions, it's notable that all four independent variables of interests (% white, % African-American, per capita income, and poverty rate) emerge as significant predictors of the number of grocery stores within each community area. Below, I offer interpretations of the linear regression findings pertaining to each independent variable:

1. For every 1% increase in the percentage of White residents, the mean number of grocery stores increases by about 0.03 (p-value = 0.011).
2. For every 1% increase in the percentage of African-american residents, the mean number of grocery stores decreases by about 0.03 (p-value = 0.002).
3. For every \$1,000 increase in the per capita income, the mean number of grocery stores increases by about 1.07 (p-value = 0.0002).
4. For every 1% increase in the percentage of residents earning less than \$25,000, the mean number of grocery stores decreases by about 0.16 (p-value = 0.014).

Simultaneous Autoregressive Model

Although the significant results above may suggest that the job is complete, it's imperative to remember that we are working with spatial data. Hence, testing for the presence of spatial autocorrelation is essential. As previously noted, if spatial autocorrelation exists and is substantial, it must be addressed by employing a simultaneous autoregressive model. Failure to do so would violate the assumption of independence, a prerequisite for trusting the results of linear regression. Below, I present the findings from the simultaneous autoregressive models.

1. Number of Grocery stores and the percentage of white residents in each community area, accounting for spatial effects

```
sarlm_white <- lagsarlm(num_grocery ~ Pct_white, data = chicago_sf, listw = chicago_nbw)
summary(sarlm_white)
```

```
##
## Call:lagsarlm(formula = num_grocery ~ Pct_white, data = chicago_sf,
##      listw = chicago_nbw)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.18597 -1.79870 -0.40542  1.04918 10.04833
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.159508   0.541014  2.1432   0.0321
## Pct_white    0.018964   0.011918  1.5912   0.1116
##
## Rho: 0.45555, LR test value: 10.118, p-value: 0.0014684
## Asymptotic standard error: 0.13075
##      z-value: 3.4842, p-value: 0.00049358
## Wald statistic: 12.14, p-value: 0.00049358
##
## Log likelihood: -185.8734 for lag model
## ML residual variance (sigma squared): 6.973, (sigma: 2.6406)
## Number of observations: 77
```

```
## Number of parameters estimated: 4
## AIC: NA (not available for weighted model), (AIC for lm: 387.86)
## LM test for residual autocorrelation
## test value: 2.8694, p-value: 0.090278
```

2. Number of Grocery stores and the percentage of African-american residents in each community area, accounting for spatial effects

```
sarlm_black <- lagsarlm(num_grocery ~ Pct_black, data = chicago_sf, listw = chicago_nbw)
summary(sarlm_black)
```

```
##
## Call:lagsarlm(formula = num_grocery ~ Pct_black, data = chicago_sf,
##      listw = chicago_nbw)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.49268 -1.67312 -0.55633  1.04410  9.60536
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.5664129  0.6668712  3.8484 0.0001189
## Pct_black    -0.0202729  0.0085562 -2.3694 0.0178174
##
## Rho: 0.40763, LR test value: 7.8876, p-value: 0.0049775
## Asymptotic standard error: 0.13498
##      z-value: 3.0201, p-value: 0.0025273
## Wald statistic: 9.1207, p-value: 0.0025273
##
## Log likelihood: -184.3804 for lag model
## ML residual variance (sigma squared): 6.7779, (sigma: 2.6034)
## Number of observations: 77
## Number of parameters estimated: 4
## AIC: NA (not available for weighted model), (AIC for lm: 382.65)
## LM test for residual autocorrelation
## test value: 1.2581, p-value: 0.26201
```

3. Number of Grocery stores and the Per capita income (in thousands) in each community area, accounting for spatial effects

```
sarlm_per_cap_income <- lagsarlm(num_grocery ~ income1000, data = chicago_sf, listw = chicago_nbw)
summary(sarlm_per_cap_income)
```

```
##
## Call:lagsarlm(formula = num_grocery ~ income1000, data = chicago_sf,
##      listw = chicago_nbw)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.43027 -1.77474 -0.39477  0.92216 10.39496
##
```

```
## Type: lag
## Coefficients: (asymptotic standard errors)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.369195    0.630457  0.5856 0.558145
## income1000  0.046726    0.015754  2.9660 0.003017
##
## Rho: 0.36781, LR test value: 6.2974, p-value: 0.012091
## Asymptotic standard error: 0.1391
##      z-value: 2.6442, p-value: 0.0081876
## Wald statistic: 6.992, p-value: 0.0081876
##
## Log likelihood: -183.0641 for lag model
## ML residual variance (sigma squared): 6.5994, (sigma: 2.5689)
## Number of observations: 77
## Number of parameters estimated: 4
## AIC: NA (not available for weighted model), (AIC for lm: 378.43)
## LM test for residual autocorrelation
## test value: 5.5531, p-value: 0.018448
```

4. Number of Grocery stores and the percentage of residents earning less than \$25,000 annually, accounting for spatial effects

```
sarlm_poverty <- lagsarlm(num_grocery ~ Pct_poverty, data = chicago_sf, listw = chicago_nbw)
summary(sarlm_poverty)
```

```
##
## Call:lagsarlm(formula = num_grocery ~ Pct_poverty, data = chicago_sf,
##      listw = chicago_nbw)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.21852 -1.59665 -0.63306  1.11119  9.74928
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.707736    0.821263  3.2970 0.0009771
## Pct_poverty -0.099946    0.055843 -1.7898 0.0734934
##
## Rho: 0.45318, LR test value: 10.202, p-value: 0.0014027
## Asymptotic standard error: 0.13026
##      z-value: 3.479, p-value: 0.0005033
## Wald statistic: 12.103, p-value: 0.0005033
##
## Log likelihood: -185.5636 for lag model
## ML residual variance (sigma squared): 6.9209, (sigma: 2.6308)
## Number of observations: 77
## Number of parameters estimated: 4
## AIC: NA (not available for weighted model), (AIC for lm: 387.33)
## LM test for residual autocorrelation
## test value: 2.9642, p-value: 0.085129
```

Each of the four models yields ρ values ranging from 0.33 to 0.42, all with p-values less than 0.05. These moderately positive ρ values indicate that there is a discernible difference, albeit slight, in the outcomes between

the SAR model and the linear regression model. Thus, it is advisable to consider spatial autocorrelation when interpreting the regression results and drawing conclusions.

Interpretations of SAR models specific to each independent variable are as follows:

1. After accounting for spatial autocorrelation between neighboring community areas, the p-value for the percentage of white residents is greater than 0.05. Therefore, we fail to reject the null hypothesis and conclude that this is not a statistically significant predictor for the number of grocery stores in each community area of Chicago. (p-value = 0.135)
2. After accounting for spatial autocorrelation between neighboring community areas, for every 1% increase in the percentage of African-american residents, the mean number of grocery stores decreases by about 0.02 (p-value = 0.03).
3. After accounting for spatial autocorrelation between neighboring community areas, for every \$1,000 increase in the per capita income, the mean number of grocery stores increases by about 0.05 (p-value = 0.004).
4. After accounting for spatial autocorrelation between neighboring community areas, the p-value for the poverty rate is greater than 0.05. Therefore, we fail to reject the null hypothesis and conclude that this is not a statistically significant predictor for the number of grocery stores in each community area of Chicago. (p-value = 0.09).

In summary, the introduction of the SAR model, which addresses spatial correlations, has resulted in changes to the analysis outcomes. Notably, two variables, namely the percentage of white residents and the percentage of residents earning less than \$25,000, which previously showed significance in the linear regression, are now rendered non-significant in the SAR model. Conversely, the percentage of African-American residents and per capita income remain statistically significant even after accounting for spatial autocorrelation. This suggests that these two variables can serve as robust predictors of the number of grocery stores in each community area of Chicago, underscoring their importance in the analysis.

About Ecological Fallacy

The Ecological Fallacy is a formal error in statistical interpretation wherein relationships observed at aggregate (group) levels are erroneously attributed to individual levels. This fallacy can lead to inaccurate conclusions and misguided assumptions, particularly in understanding social phenomena. Furthermore, it can perpetuate false beliefs or assumptions if not addressed.

In the context of this analysis, demographic data and the number of grocery stores are aggregated at the community area level. Consequently, any conclusions drawn regarding spatial autocorrelation and the significance of demographic factors concerning the number of grocery stores should not be extrapolated to smaller neighborhoods, households, or individual levels. It's important to recognize that the results may vary significantly if the same analysis were conducted with data collected at different aggregate levels.