Simultaneous Autoregressive Model

Sam Song

Setup

Simultaneous Autoregressive Model

$$Y = \beta_0 + \beta_1 X + \rho \sum w_i (Y_i - \beta_0 - \beta_1 X_i)$$
$$Y = \beta_0 + \beta_1 X + \rho \sum w_i Y_i$$

ρ describes the degree of correlation with neighbors; if ρ value is close to 1, it weights heavily and if ρ value is close to 0, not much weight w_i is the weight on neighbor i.

$Y_i - \beta_0 - \beta_1 X_i$ is the residual!!

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

Data setup

```
# 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/ReferenceCCAProfiles20162020.csv") %>
  select(GEOID, GEOG, `2020_POP`, WHITE, ASIAN, BLACK, HISP, OTHER, UNEMP, NO_VEH, MEDINC, INCPERCAP, I
```

```r
  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)
```

```
## Rows: 77 Columns: 258
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (21): GEOG, RES_NAICS1_TYPE, RES_NAICS2_TYPE, RES_NAICS3_TYPE, RES_NAIC...
## dbl (201): GEOID, 2000_POP, 2010_POP, 2020_POP, 2020_HH, 2020_HH_SIZE, TOT_P...
## lgl  (36): IDES_START_EMP, IDES_MID_EMP, IDES_CURR_EMP, RET_SALES, GEN_MERCH...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# 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/Hea
grocery_store <- read_csv("~/Documents/Data/grocery_chicago.csv")
```

```
## Rows: 264 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Store Name, Address, Zip, New status, Last updated, Location
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
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') %>%
  # transform the dataset to a sf object
  st_as_sf(coords = c("x", "y")) %>%
  # assign the Coordinate Reference System (WGS 84)
  st_set_crs(4236)

# transorm 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)
```

```
## Joining with ‘by = join_by(ComAreaID)‘
```

```
# 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)
```

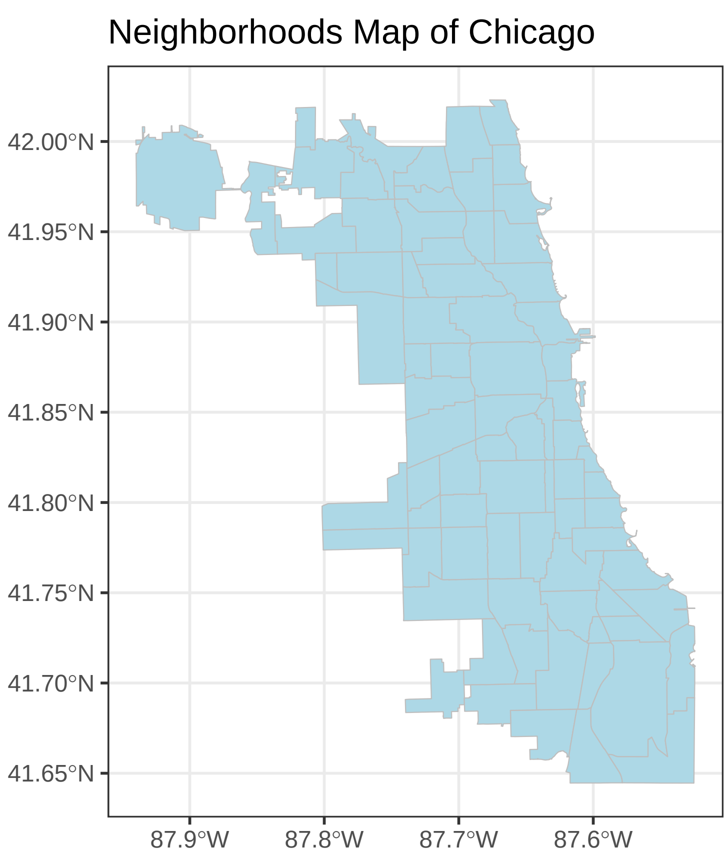
- Data tidying process

```r
# refined dataset
chicago_sf
```

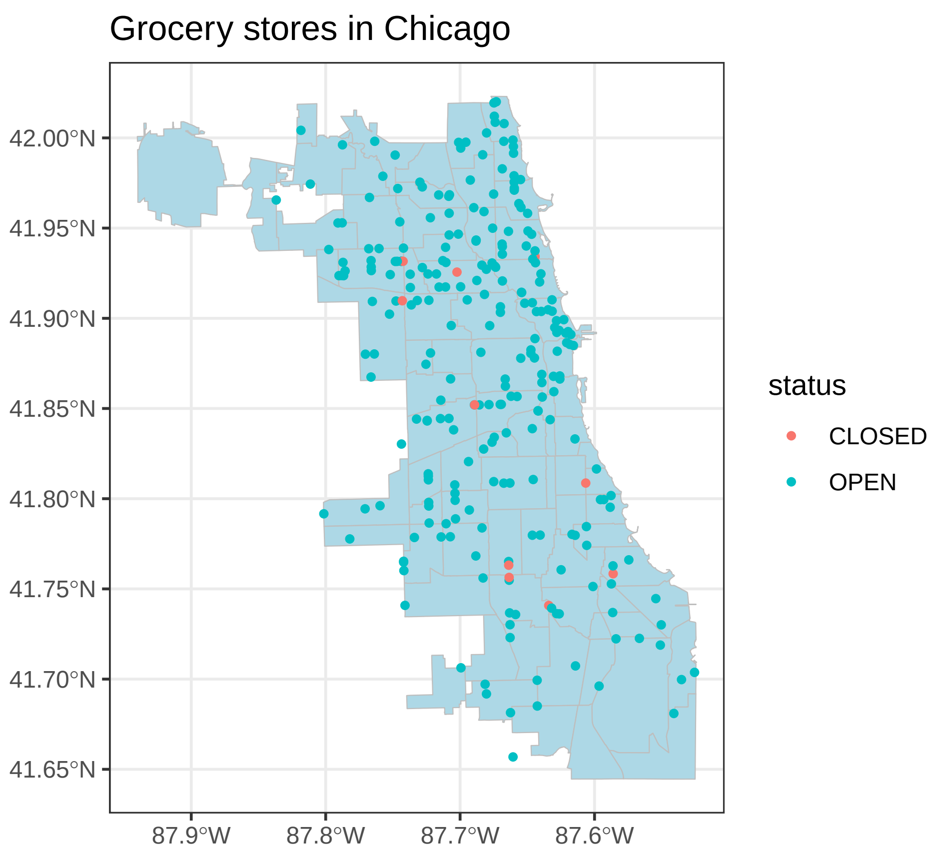
```
## Simple feature collection with 77 features and 27 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -87.94011 ymin: 41.64454 xmax: -87.52414 ymax: 42.02304
## Geodetic CRS:  WGS84(DD)
## First 10 features:
##    ComAreaID       community shape_area shape_len Pop_2020      White
## 1         35         DOUGLAS   46004621  31027.05    20291  2270.0000
## 2         36         OAKLAND   16913961  19565.51     6799   307.0000
## 3         37     FULLER PARK   19916705  25339.09     2567   112.1807
## 4         38 GRAND BOULEVARD   48492503  28196.84    24589   976.0000
## 5         39         KENWOOD   29071742  23325.17    19116  3759.0000
## 6          4  LINCOLN SQUARE   71352328  36624.60    40494 26590.0000
## 7         40 WASHINGTON PARK   42373881  28175.32    12707   191.0000
## 8         41       HYDE PARK   45105380  29746.71    29456 13019.0000
## 9         42        WOODLAWN   57815180  46936.96    24425  1838.0000
## 10         1     ROGERS PARK   51259902  34052.40    55628 24644.0000
##         Asian     Black   Hispanic Other Unemployed No_vehicle Med_income
## 1  3121.00000 13964.000  1152.0000   943   1372.000  4266.0000   35796.12
## 2   114.00000  6043.000   407.0000   102    730.000  1071.0000   36837.61
## 3    10.39759  1933.133   145.6024    31    184.702   700.7779   17216.79
## 4   172.00000 21200.000   771.0000   550   1645.000  3924.0000   39110.74
## 5  1081.00000 11864.000   404.0000   864    860.000  3135.0000   52336.45
## 6  3771.00000  1234.000  7373.0000  2382   1468.000  4149.0000   80899.84
## 7     0.00000  9559.000   233.0000   444    792.000  2240.0000   23351.06
## 8  3946.00000  6773.000  2083.0000  1901   1002.000  5666.0000   52422.57
## 9   766.00000 18510.000   652.0000   736   1881.615  3713.8542   27540.98
## 10 3018.00000 15059.000 10537.0000  2385   1484.000  9961.0000   46244.10
##    Per_cap_income Income_under_25K Pct_bad_transit Pct_not_walkable Pct_white
## 1        28546.89        4060.0000               0            0.000     11.19
## 2        28101.44        1355.0000               0            0.207      4.52
## 3        14738.42         789.7167               0            0.000      4.37
## 4        29807.92        4201.0000               0            0.000      3.97
## 5        48781.33        2957.0000               0            0.000     19.66
## 6        49797.25        2031.0000               0            0.000     65.66
## 7        16193.13        2456.0000               0            0.000      1.50
## 8        48606.07        4014.0000               0            0.000     44.20
## 9        20824.42        4545.0000               0            0.021      7.53
## 10       29682.50        7019.0000               0            0.000     44.30
##    Pct_asian Pct_black Pct_hispanic Pct_other Pct_unemployed Pct_poverty
## 1      15.38     68.82         5.68      4.65           6.76       20.01
## 2       1.68     88.88         5.99      1.50          10.74       19.93
## 3       0.41     75.31         5.67      1.21           7.20       30.76
## 4       0.70     86.22         3.14      2.24           6.69       17.08
## 5       5.65     62.06         2.11      4.52           4.50       15.47
## 6       9.31      3.05        18.21      5.88           3.63        5.02
## 7       0.00     75.23         1.83      3.49           6.23       19.33
## 8      13.40     22.99         7.07      6.45           3.40       13.63
## 9       3.14     75.78         2.67      3.01           7.70       18.61
## 10      5.43     27.07        18.94      4.29           2.67       12.62
```

```
##    Pct_no_vehicle num_grocery                       geometry grocery_100k
## 1           21.02           1 MULTIPOLYGON (((-87.60914 4...     4.928293
## 2           15.75           0 MULTIPOLYGON (((-87.59215 4...     0.000000
## 3           27.30           0 MULTIPOLYGON (((-87.6288 41...     0.000000
## 4           15.96           0 MULTIPOLYGON (((-87.60671 4...     0.000000
## 5           16.40           2 MULTIPOLYGON (((-87.59215 4...    10.462440
## 6           10.25           3 MULTIPOLYGON (((-87.67441 4...     7.408505
## 7           17.63           1 MULTIPOLYGON (((-87.60604 4...     7.869678
## 8           19.24           4 MULTIPOLYGON (((-87.58038 4...    13.579576
## 9           15.21           3 MULTIPOLYGON (((-87.57714 4...    12.282497
## 10          17.91           7 MULTIPOLYGON (((-87.65456 4...    12.583591
```

```r
# create a visualization to see if all of the neighborhoods of Chicago are included in the dataset
ggplot(chicago_sf) +
  geom_sf(color = "grey", fill = "lightblue") +
  theme_bw() +
  labs(title = "Neighborhoods Map of Chicago")
```




```r
# overlay the locations of grecery store
ggplot(chicago_sf) +
  geom_sf(color = "grey", fill = "lightblue") +
  # grocery store locations in points
  geom_sf(data = grocery_store, size = 1, aes(color = status)) +
  theme_bw() +
  labs(title = "Grocery stores in Chicago")
```



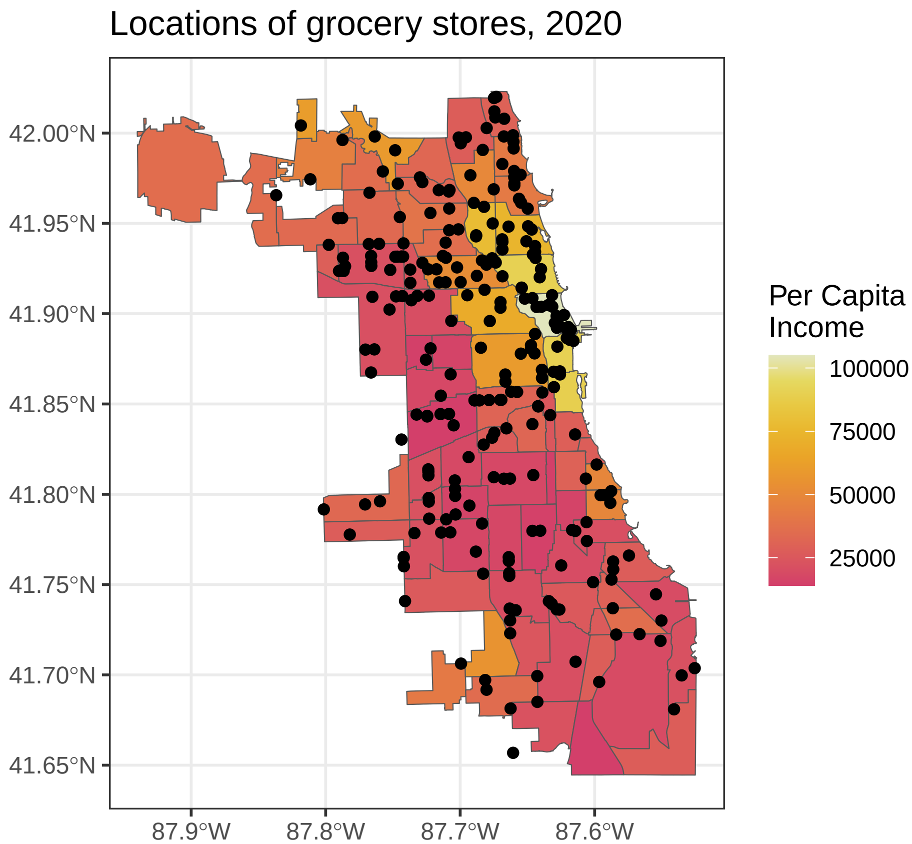

```
tmap_mode("view")

tm_shape(chicago_sf) +
  tm_borders(col = "red",
             alpha = 0.5) +
  tm_shape(grocery_store) +
  tm_dots(col = "status",
          popup.vars = c("Address", "status"))
```

2. What type of geometry does chicago_sup have? Would we consider this area or point pattern data?

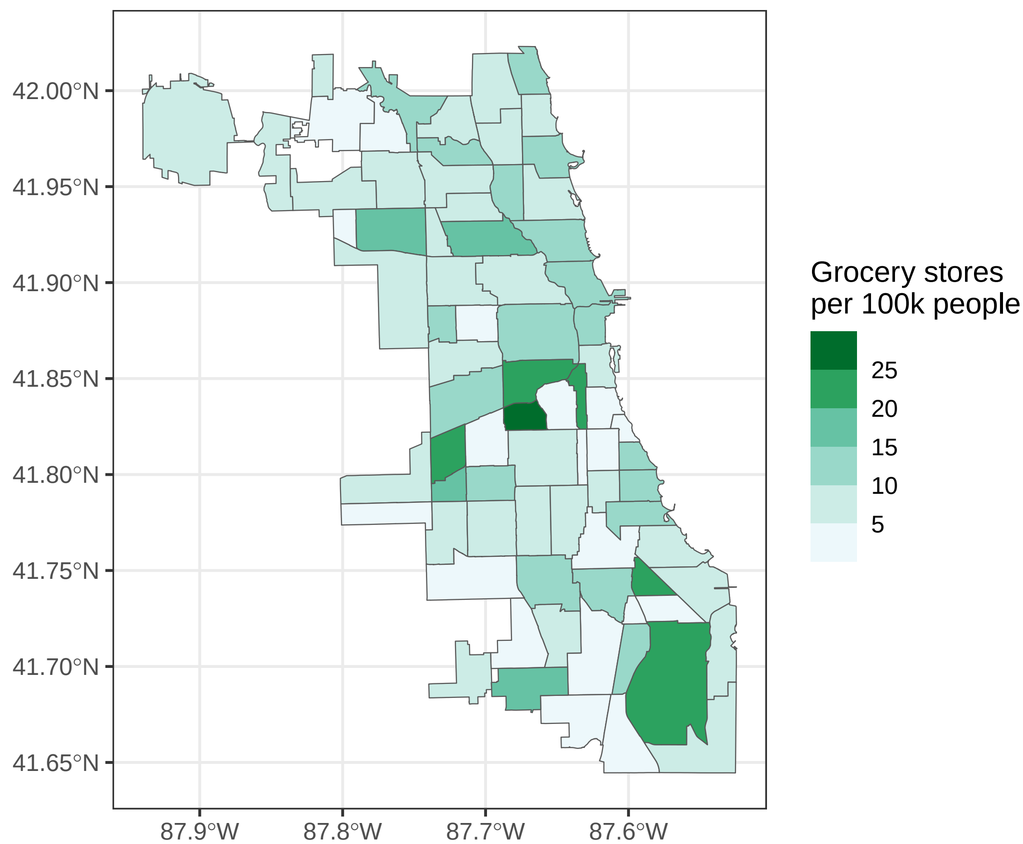
- point

Visualize

```
# left one is point pattern, and the right one is areal, at this point can't really directly compare th
ggplot(chicago_sf) +
  geom_sf(aes(fill = Per_cap_income)) +
  scale_fill_gradientn(colours = colorspace::heat_hcl(10)) +
  geom_sf(data = grocery_store) +
  theme_bw() +
  labs(fill = "Per Capita\nIncome",
       title = "Locations of grocery stores, 2020")
```




```r
# converted the point pattern data to areal data so that the comparison can be done
ggplot(chicago_sf) +
  geom_sf(aes(fill = grocery_100k)) +
  scale_fill_fermenter(palette = 2, direction = 1) +
  labs(fill = "Grocery stores\nper 100k people") +
  theme_bw()
```



Moran's I review

3. Comparing the plots of Per capita income and Grocery stores per 100k, which variable do you think has stronger spatial autocorrelation?

Whether we are looking at per capita income or at number of grocery stores, we start by creating the neighbors (nb) and the neighbor weights (nbw).

```
# Create neigbors
chicago_nb <- poly2nb(chicago_sf, queen = TRUE)
# Create neighbor weights
chicago_nbw <- nb2listw(chicago_nb, style = "W", zero.policy = TRUE)
```

4. What does the code below do? Interpret the result.

```
moran.mc(chicago_sf$num_grocery, chicago_nbw, nsim = 499)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  chicago_sf$num_grocery
## weights: chicago_nbw
## number of simulations + 1: 500
```

```
##
## statistic = 0.28664, observed rank = 500, p-value = 0.002
## alternative hypothesis: greater
```

There is a moderately strong spatial autocorrelation (I = .51)

```
moran.mc(chicago_sf$num_grocery, chicago_nbw, nsim = 499)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  chicago_sf$num_grocery
## weights: chicago_nbw
## number of simulations + 1: 500
##
## statistic = 0.28664, observed rank = 499, p-value = 0.002
## alternative hypothesis: greater
```

There is a strong spatial autocorrelation (I = .69) in the percentage of residents of a neighborhood that identify as White. Neighborhoods tend to have similar percentage of white residents as their neighbors.

5. Repeat #4 using the grocery_100k variable. Interpret the result.

```
moran.mc(chicago_sf$grocery_100k, chicago_nbw, nsim = 499)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  chicago_sf$grocery_100k
## weights: chicago_nbw
## number of simulations + 1: 500
##
## statistic = -0.095379, observed rank = 55, p-value = 0.89
## alternative hypothesis: greater
```

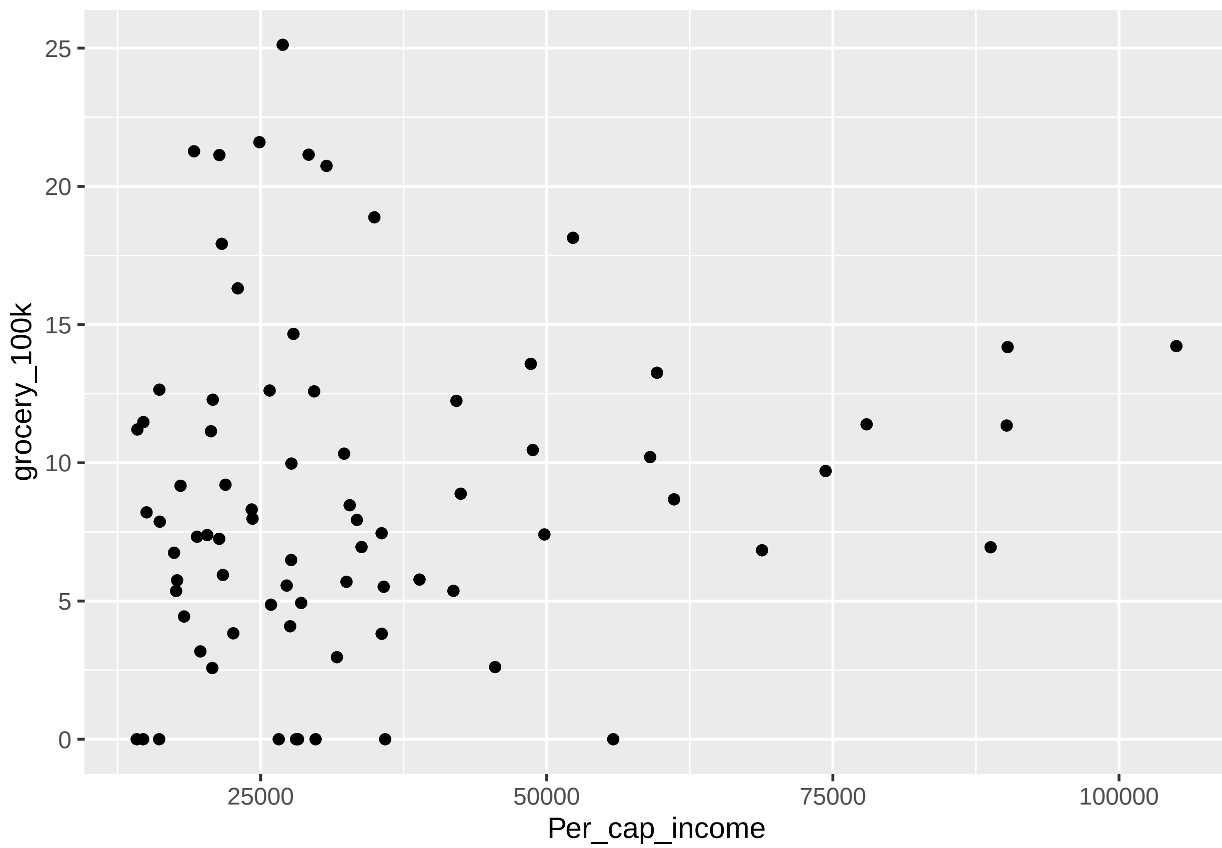
There is a weaker spatial autocorrelation (I = .13) in the

6. Comparing Per capita income and Grocery stores per 100k in #4 and #5, which variable do you think has stronger spatial autocorrelation?

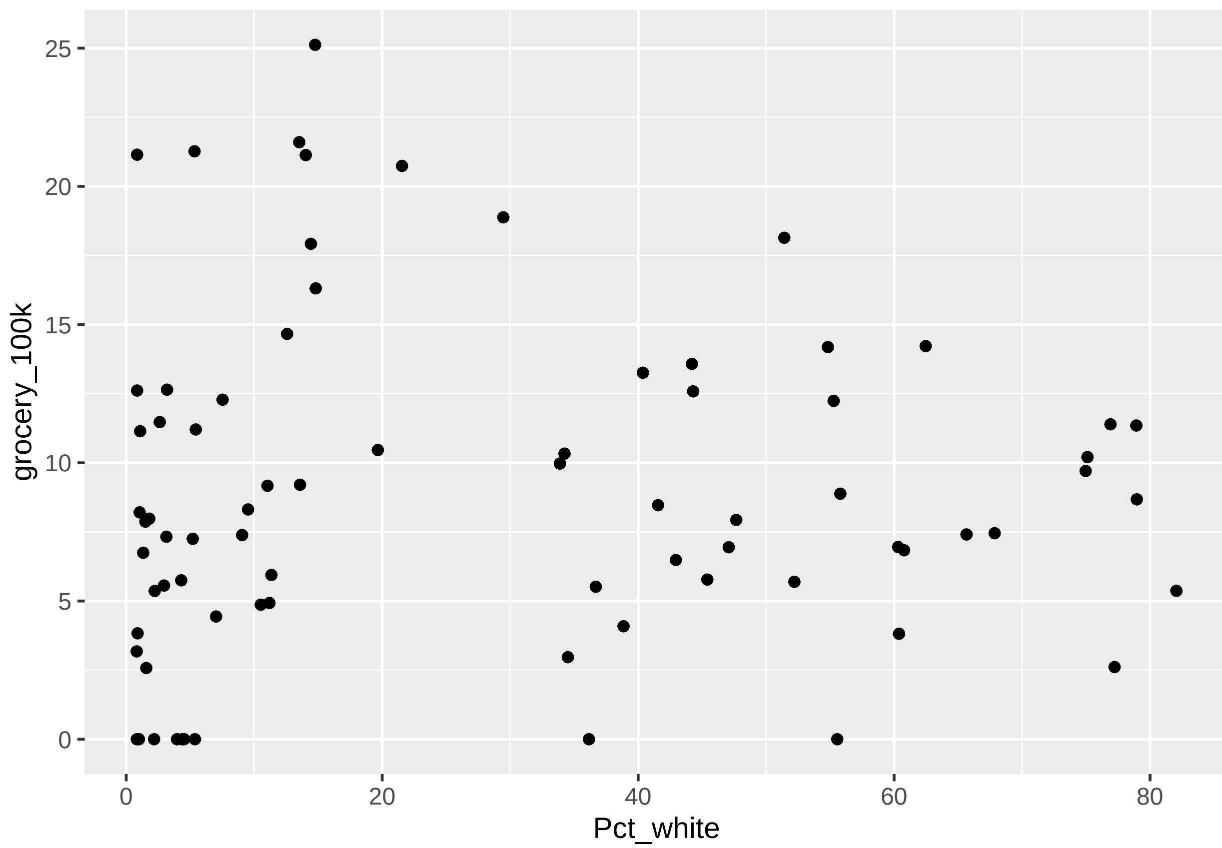
Regression

Start by exploring relationships with other variables:

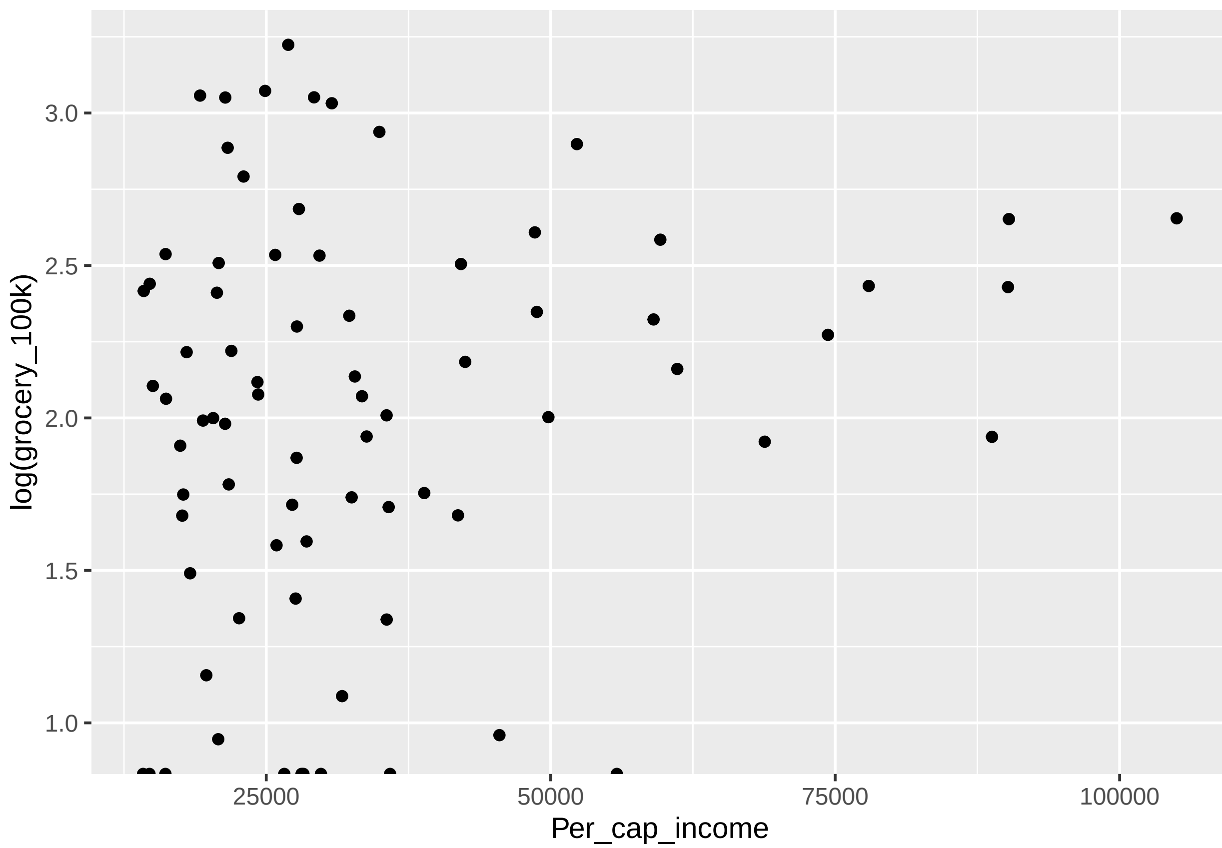
```
ggplot(chicago_sf) +
  geom_point(aes(Per_cap_income, grocery_100k))
```



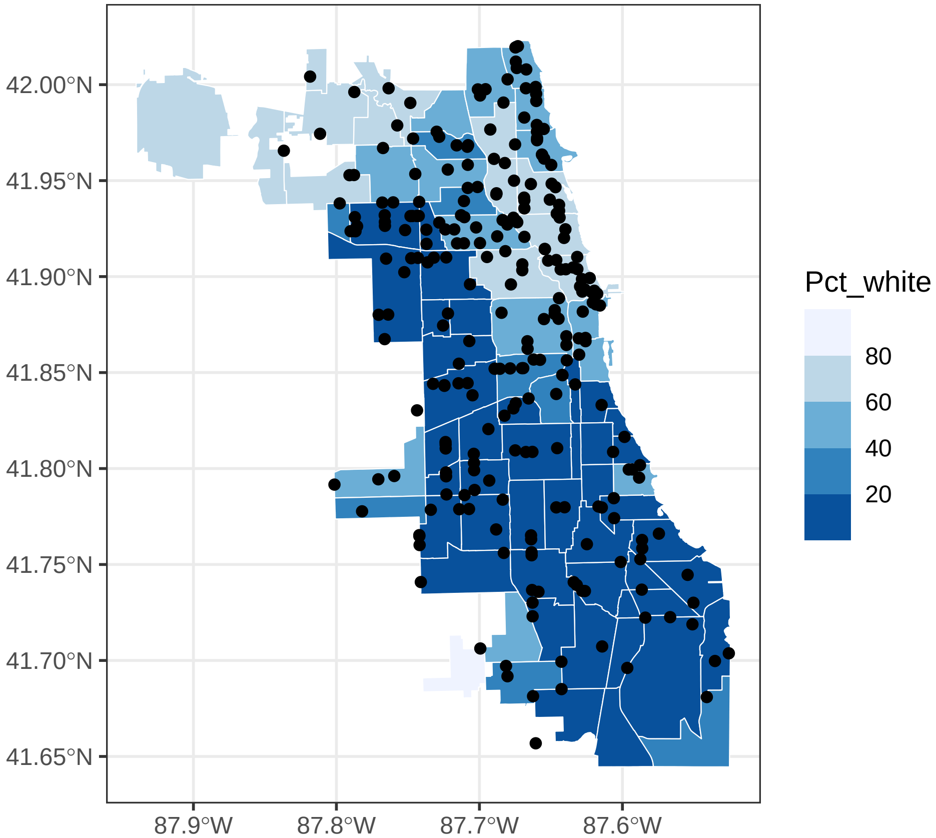

```
ggplot(chicago_sf) +
  geom_point(aes(Pct_white, grocery_100k))
```




```r
ggplot(chicago_sf) +
  geom_point(aes(Per_cap_income, log(grocery_100k)))
```

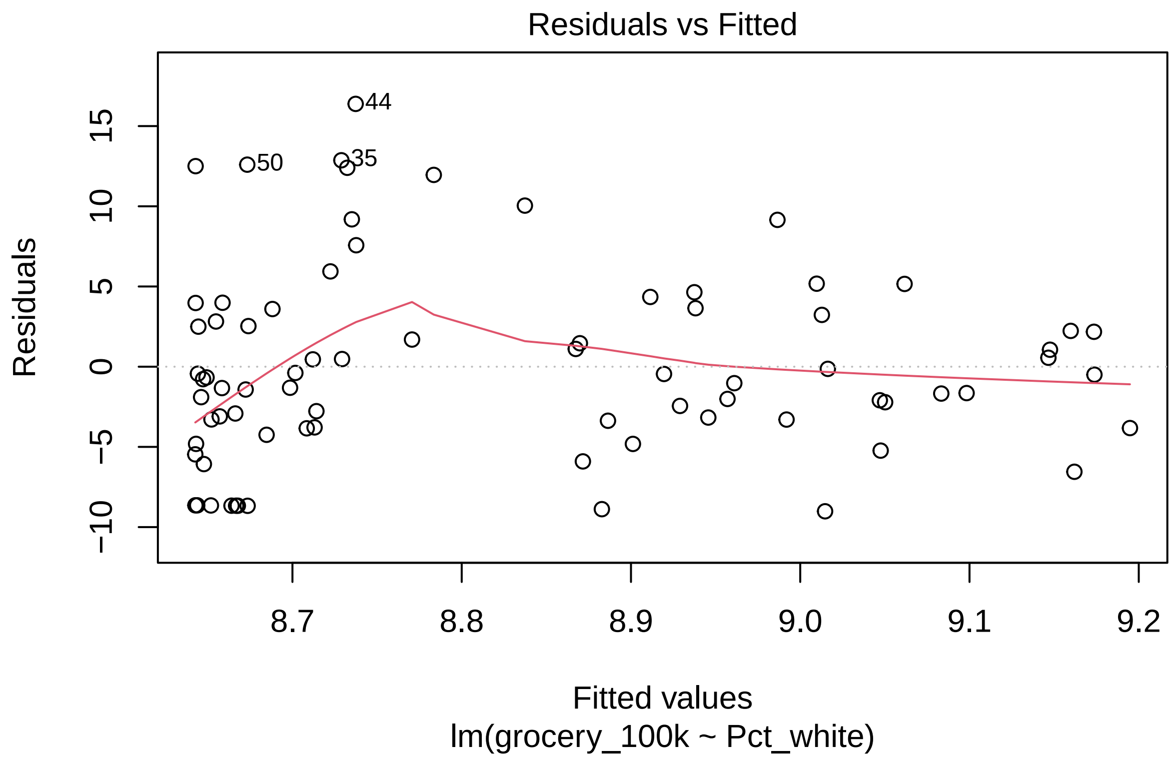



```r
ggplot(chicago_sf) +
  geom_sf(aes(fill = Pct_white), color = "white") +
  scale_fill_fermenter() +
  geom_sf(data = grocery_store) +
  theme_bw()
```



Start by fitting a linear regression model:

```
lm1 <- lm(grocery_100k ~ Pct_white, data= chicago_sf)
plot(lm1, 1)
```




```r
summary(lm1)
```

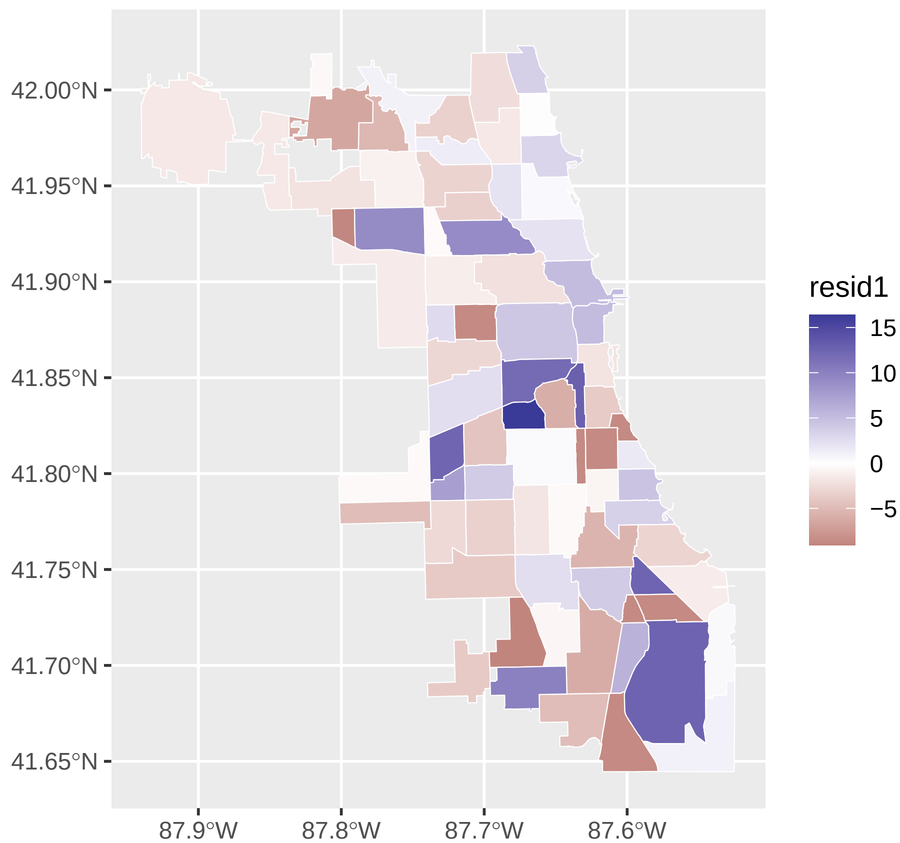
```
##
## Call:
## lm(formula = grocery_100k ~ Pct_white, data = chicago_sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0147 -3.7848 -0.7775  3.2288 16.3836
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.637000   0.992477   8.702 5.48e-13 ***
## Pct_white   0.006798   0.026114   0.260    0.795
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.976 on 75 degrees of freedom
## Multiple R-squared:  0.0009026, Adjusted R-squared:  -0.01242
## F-statistic: 0.06776 on 1 and 75 DF,  p-value: 0.7953
```

Join residuals to sf and Plot Residuals:

```r
chicago_sf$resid1 <- residuals(lm1)
```

7. Using ggplot, make a chloropleth map of the residuals.

```r
# looking at the colors of graph, the assumption of independent is violated, so ordinary linear regress
ggplot(chicago_sf) +
  geom_sf(aes(fill = resid1), color = "white") +
  scale_fill_gradient2()
```



8. Check moran's I:

```r
moran(chicago_sf$resid1,
                        chicago_nbw,
                        n = length(chicago_nb),
                        S0 = Szero(chicago_nbw))
```

```
## $I
## [1] -0.09631616
##
## $K
## [1] 3.074718
```

Fit spatial regression:

```r
sarlm1 <- lagsarlm(grocery_100k ~ Pct_white, data = chicago_sf, listw = chicago_nbw)

summary(sarlm1)
```

```
##
## Call:lagsarlm(formula = grocery_100k ~ Pct_white, data = chicago_sf,
##     listw = chicago_nbw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6186 -3.4426 -1.0290  3.2958 16.5806
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##              Estimate Std. Error z value  Pr(>|z|)
## (Intercept) 10.882297   1.861657  5.8455 5.051e-09
## Pct_white    0.007775   0.025319  0.3071    0.7588
##
## Rho: -0.25317, LR test value: 1.8758, p-value: 0.17081
## Asymptotic standard error: 0.1827
##     z-value: -1.3858, p-value: 0.16582
## Wald statistic: 1.9204, p-value: 0.16582
##
## Log likelihood: -244.9676 for lag model
## ML residual variance (sigma squared): 33.537, (sigma: 5.7911)
## Number of observations: 77
## Number of parameters estimated: 4
## AIC: NA (not available for weighted model), (AIC for lm: 497.81)
## LM test for residual autocorrelation
## test value: 0.0876, p-value: 0.76725
```

- $\rho = 0.06$; spatial autocorrelation in the number of grocery stores in neighboring communities is pretty low
- if ρ is small, OLS is a good model. if ρ is big, OLS is not to be trusted.

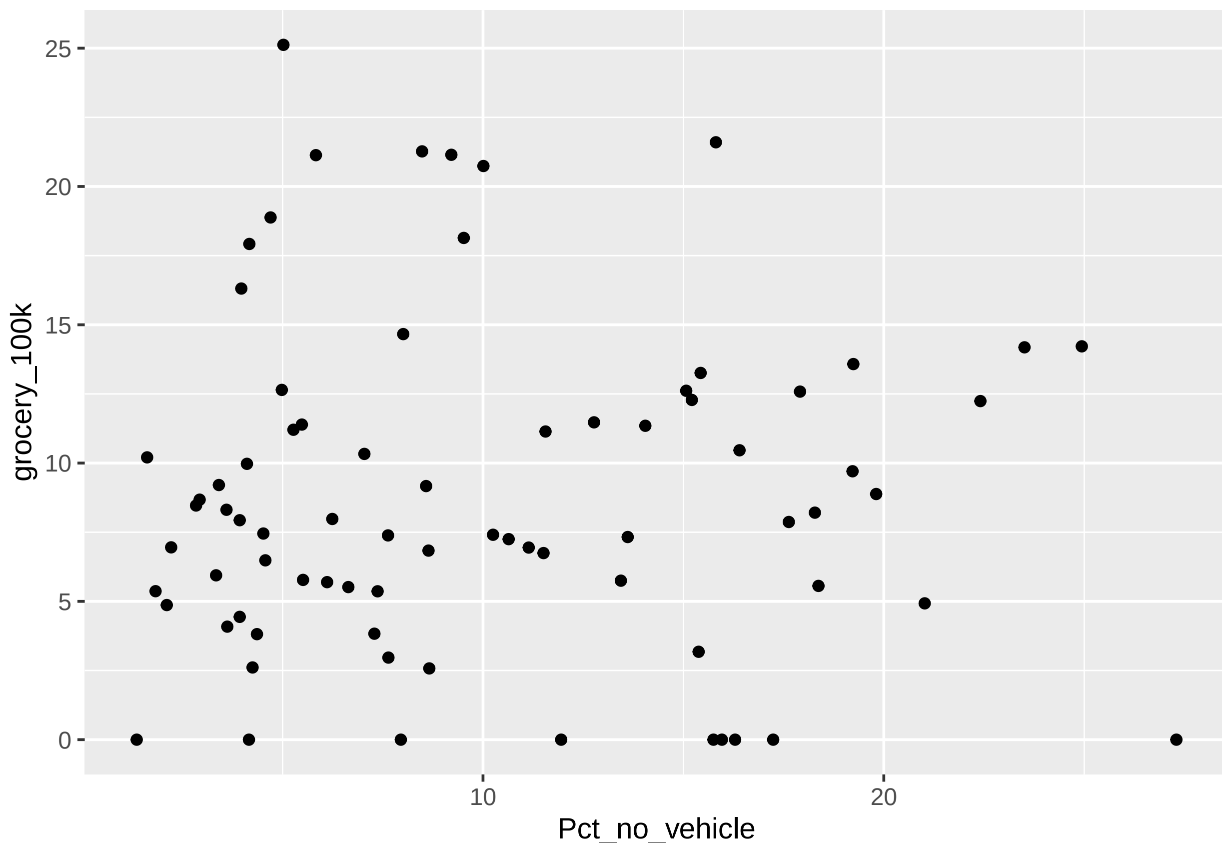
9. Compare the SAR (lagsarlm) and OLS (lm) models. Look at estimates of slope and intercept, the standard error, and p-value.
10. Write 2-3 concluding sentences about what you learned of the distribution of grocery stores throughout Chicago. Consider including ideas from your background readings.

- we caanot conclude satistical significance that regions with white people are correlated with more grocery store. Yet, aggregation and confounding variables exists in our analysis. It is also important to note that the graphs paint a different story than our original statistical analysis.

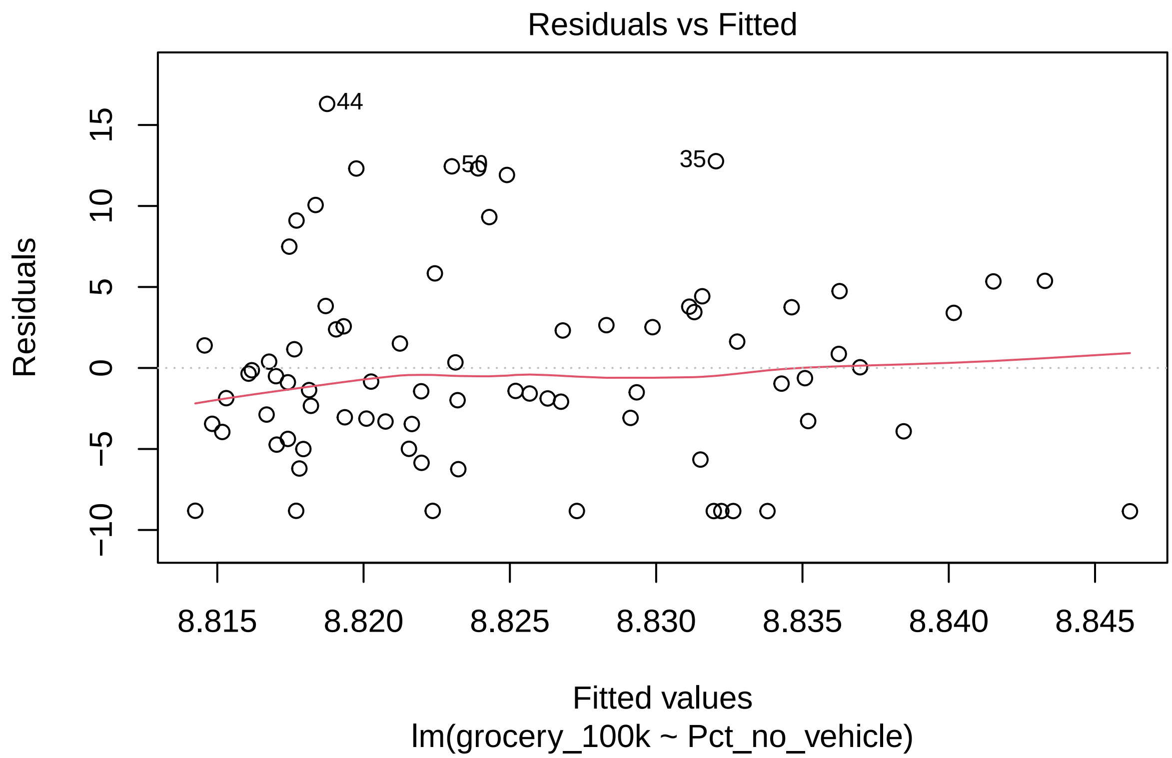
_ **Interpretation**: For every % point increase in white residents, number of groceries per 100,000 residents is predicted to increase 1.002 times (or by .2%).

Tries

```r
# Percent no vehicle
ggplot(data = chicago_sf) +
  geom_point(aes(x = Pct_no_vehicle, y = grocery_100k))
```



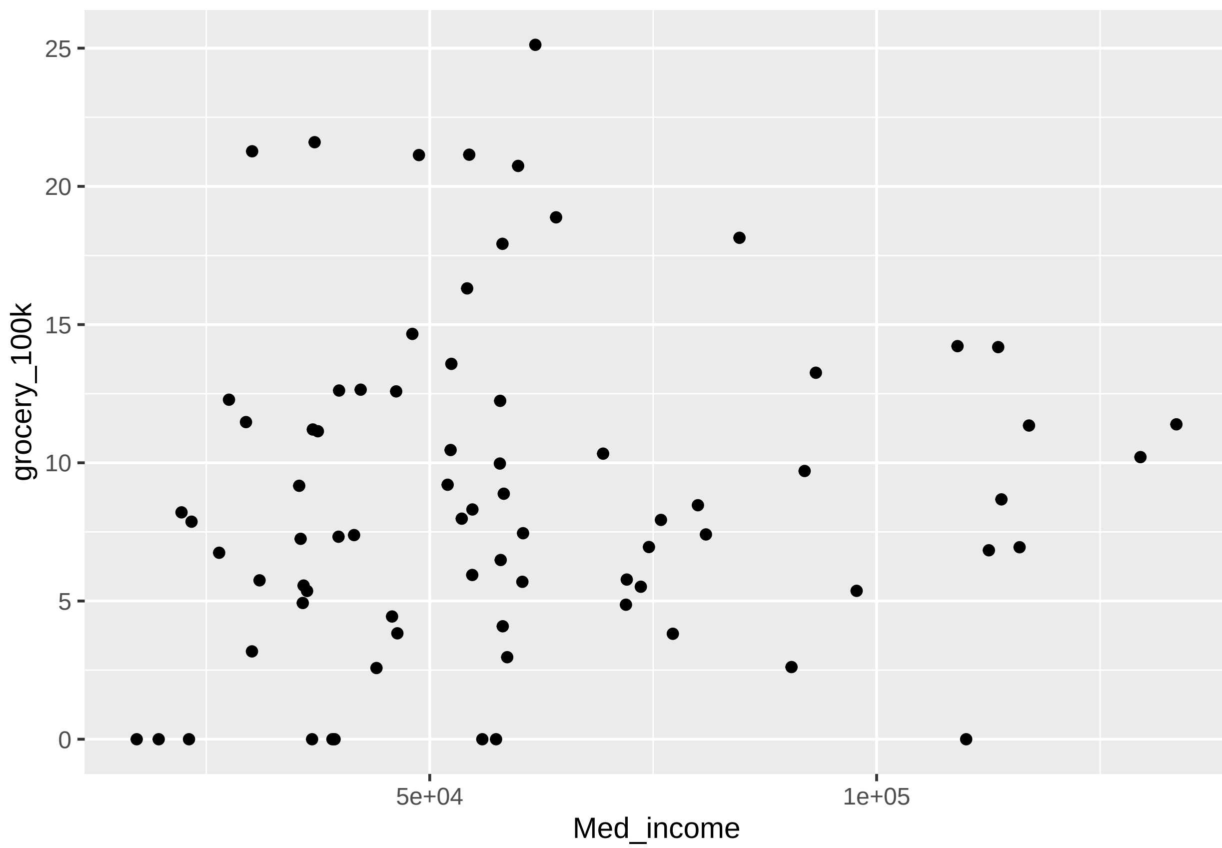

```
lm_vehicle <- lm(grocery_100k ~ Pct_no_vehicle, data = chicago_sf)
plot(lm_vehicle, 1)
```



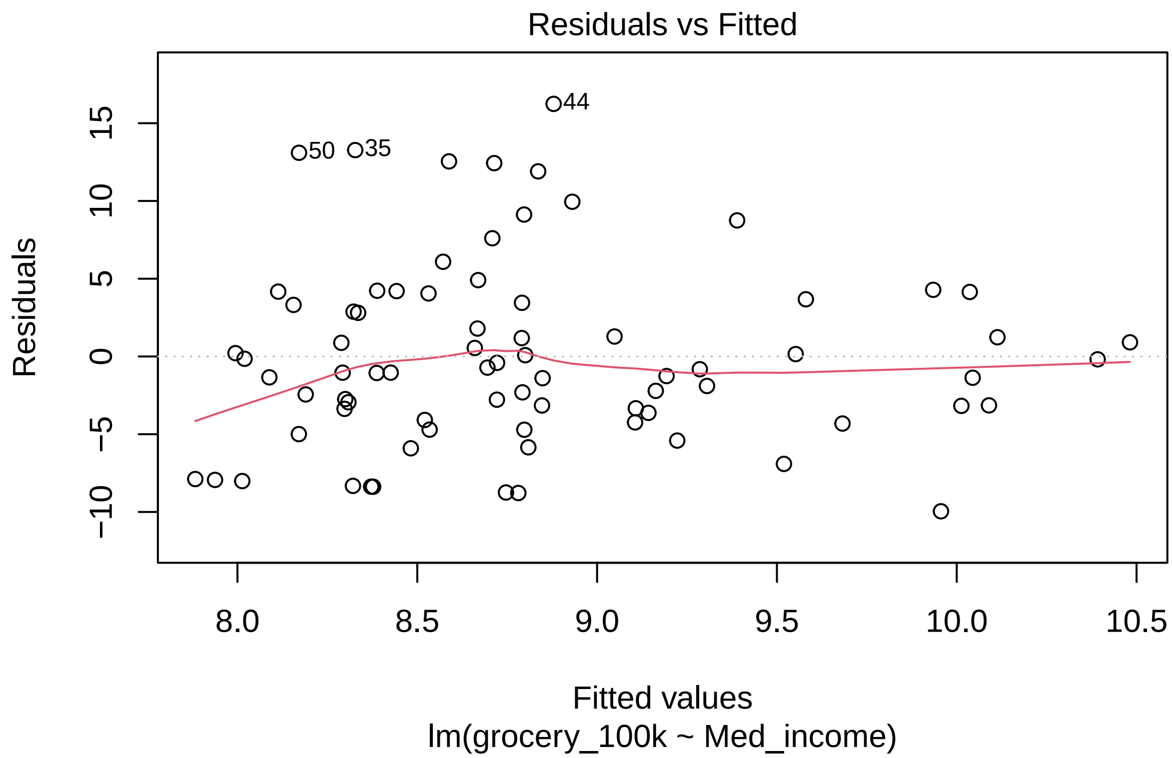

```r
summary(lm_vehicle)
```

```
##
## Call:
## lm(formula = grocery_100k ~ Pct_no_vehicle, data = chicago_sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8462 -3.4579 -0.8834  3.4014 16.3021
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.812572   1.273156   6.922  1.3e-09 ***
## Pct_no_vehicle 0.001232   0.107303   0.011    0.991
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.979 on 75 degrees of freedom
## Multiple R-squared:  1.756e-06,  Adjusted R-squared:  -0.01333
## F-statistic: 0.0001317 on 1 and 75 DF,  p-value: 0.9909
```

```r
# Median Income
ggplot(data = chicago_sf) +
  geom_point(aes(x = Med_income, y = grocery_100k))
```



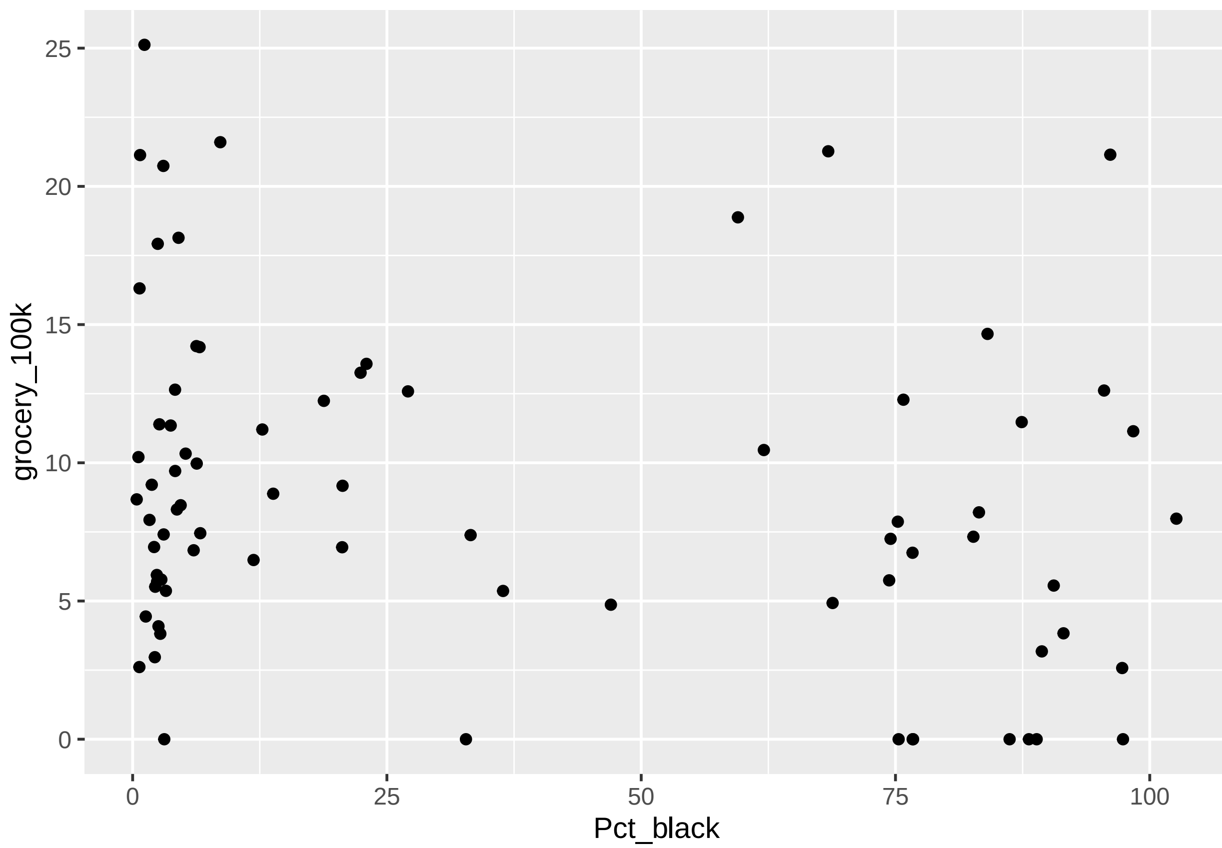

```r
lm_income <- lm(grocery_100k ~ Med_income, data = chicago_sf)
plot(lm_income, 1)
```



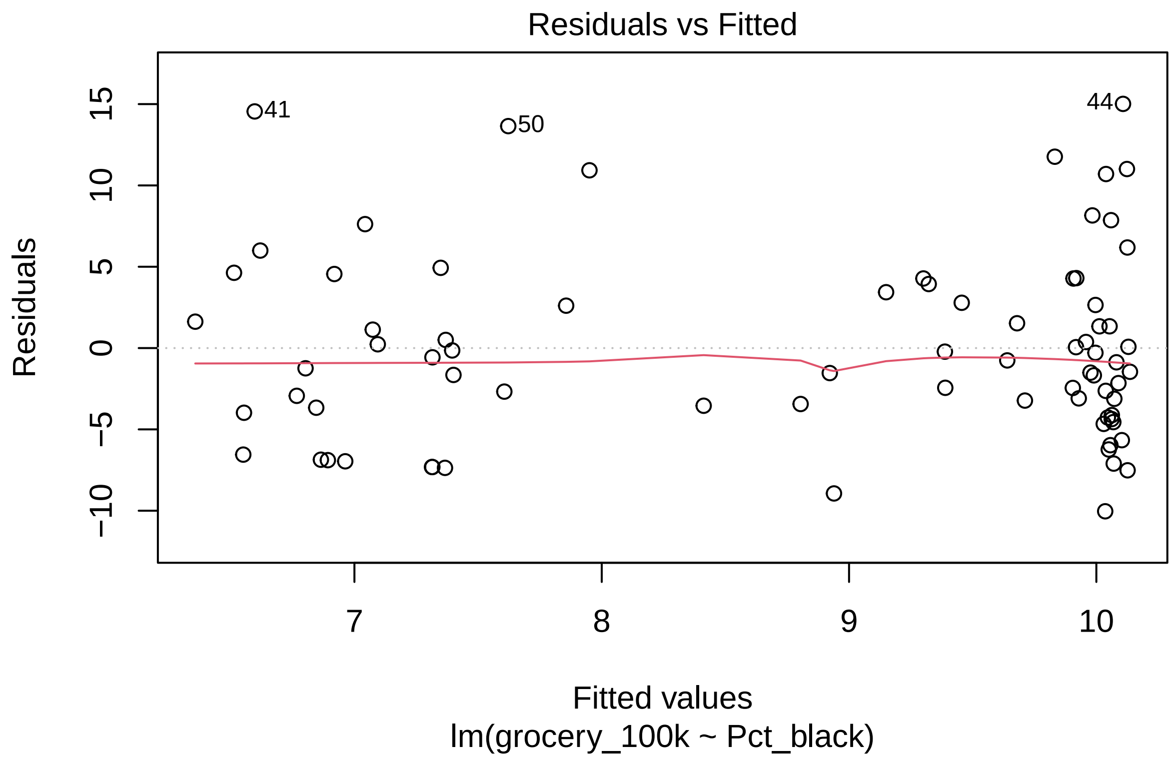

```r
summary(lm_income)
```

```
##
## Call:
## lm(formula = grocery_100k ~ Med_income, data = chicago_sf)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.956 -3.627 -1.041  3.450 16.242
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.498e+00  1.598e+00   4.691 1.19e-05 ***
## Med_income  2.234e-05  2.437e-05   0.917    0.362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.946 on 75 degrees of freedom
## Multiple R-squared:  0.01108,    Adjusted R-squared:  -0.002105
## F-statistic: 0.8403 on 1 and 75 DF,  p-value: 0.3622
```

```r
# Percent African Americans
ggplot(data = chicago_sf) +
  geom_point(aes(x = Pct_black, y = grocery_100k))
```



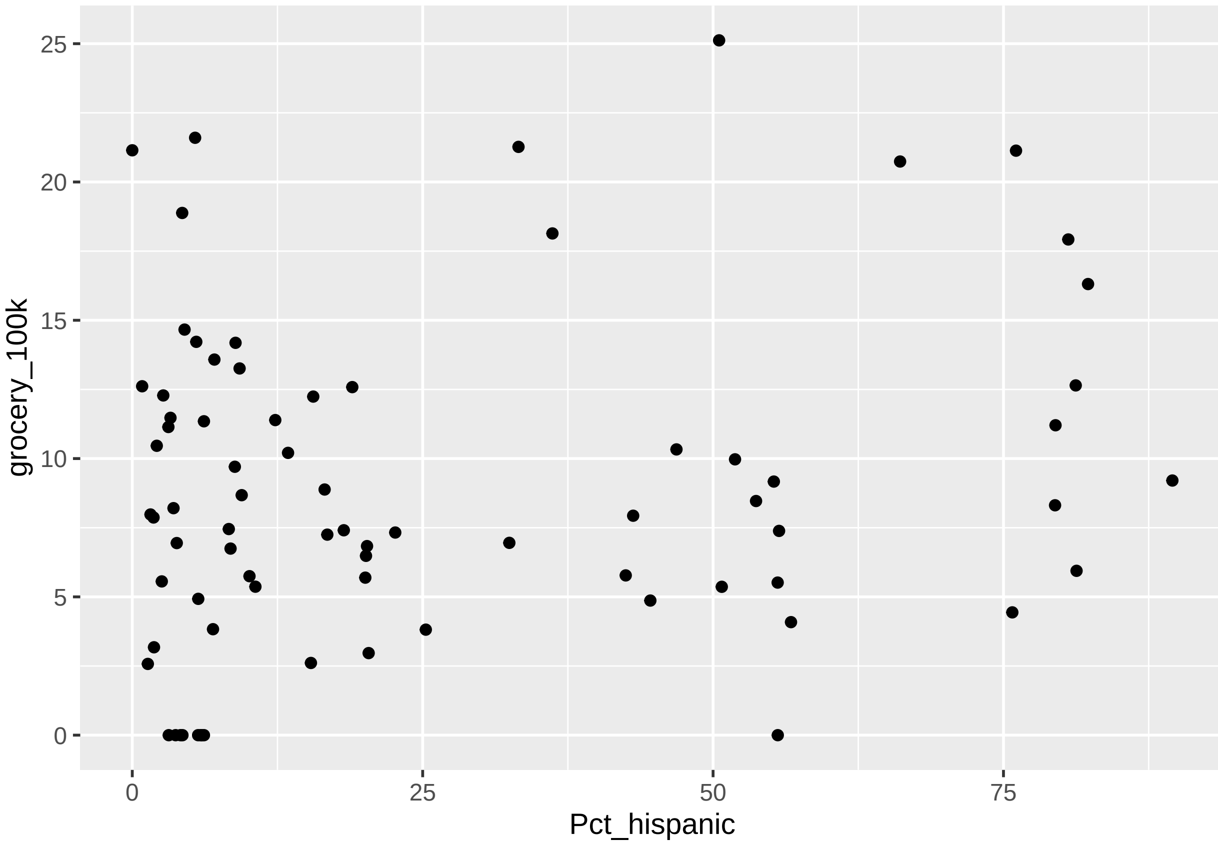

```
lm_black <- lm(grocery_100k ~ Pct_black, data = chicago_sf)
plot(lm_black, 1)
```



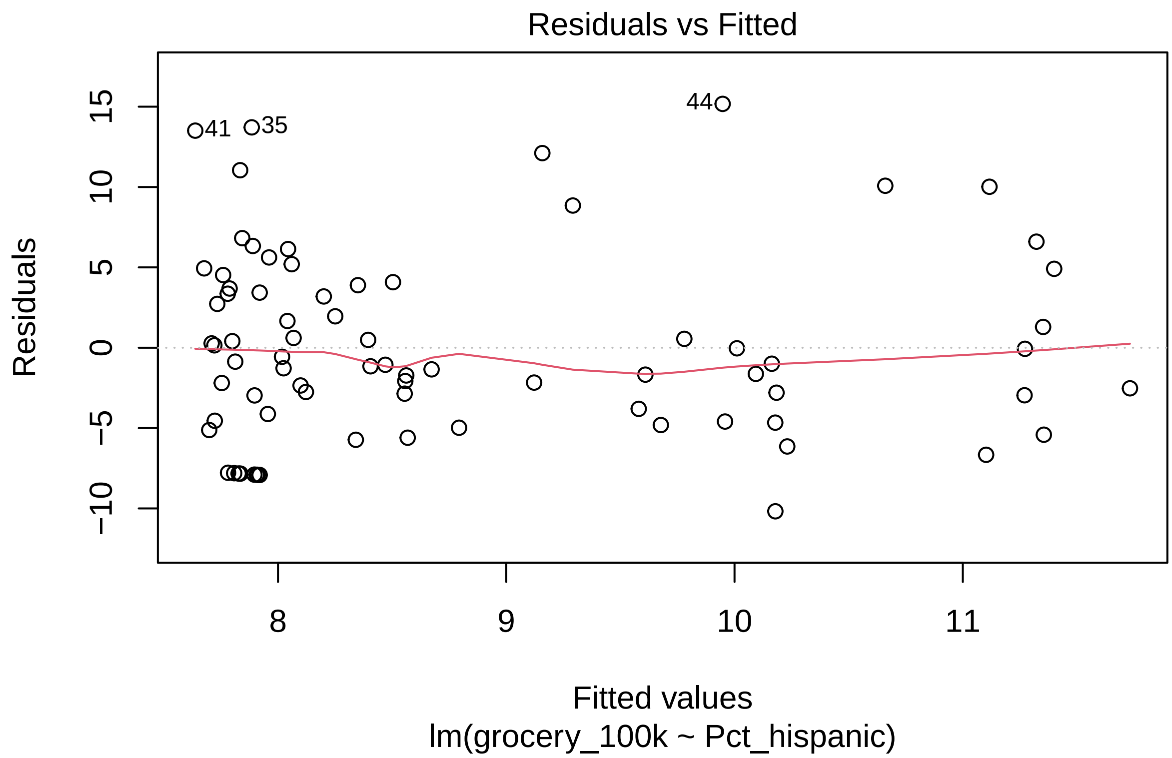

```r
summary(lm_black)
```

```
##
## Call:
## lm(formula = grocery_100k ~ Pct_black, data = chicago_sf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0358  -3.9772  -0.8749   3.4337  15.0130
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.15079    0.91938   11.04   <2e-16 ***
## Pct_black   -0.03698    0.01778   -2.08   0.0409 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.814 on 75 degrees of freedom
## Multiple R-squared:  0.05454,    Adjusted R-squared:  0.04193
## F-statistic: 4.326 on 1 and 75 DF,  p-value: 0.04094
```

```r
# Percent Hispanics
ggplot(data = chicago_sf) +
  geom_point(aes(x = Pct_hispanic, y = grocery_100k))
```



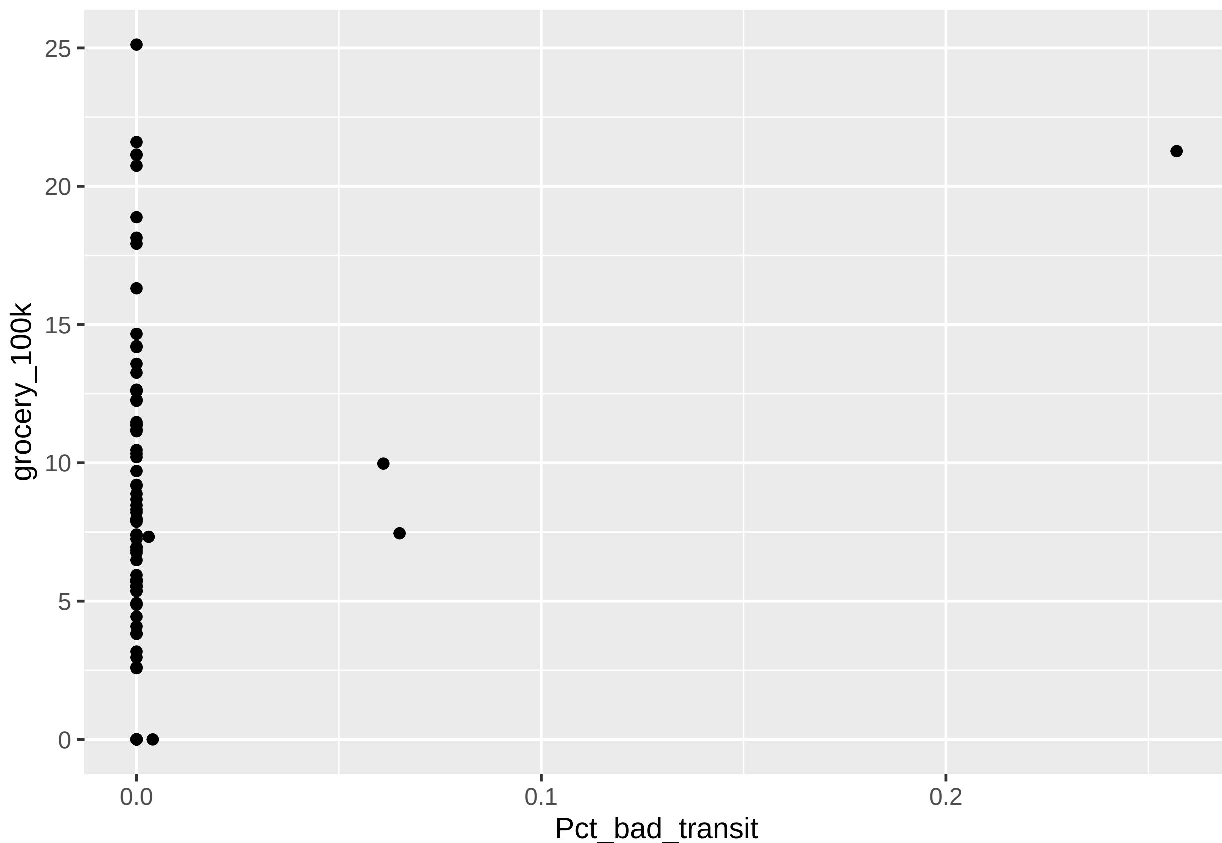

```
lm_hisp <- lm(grocery_100k ~ Pct_hispanic, data = chicago_sf)
plot(lm_hisp, 1)
```



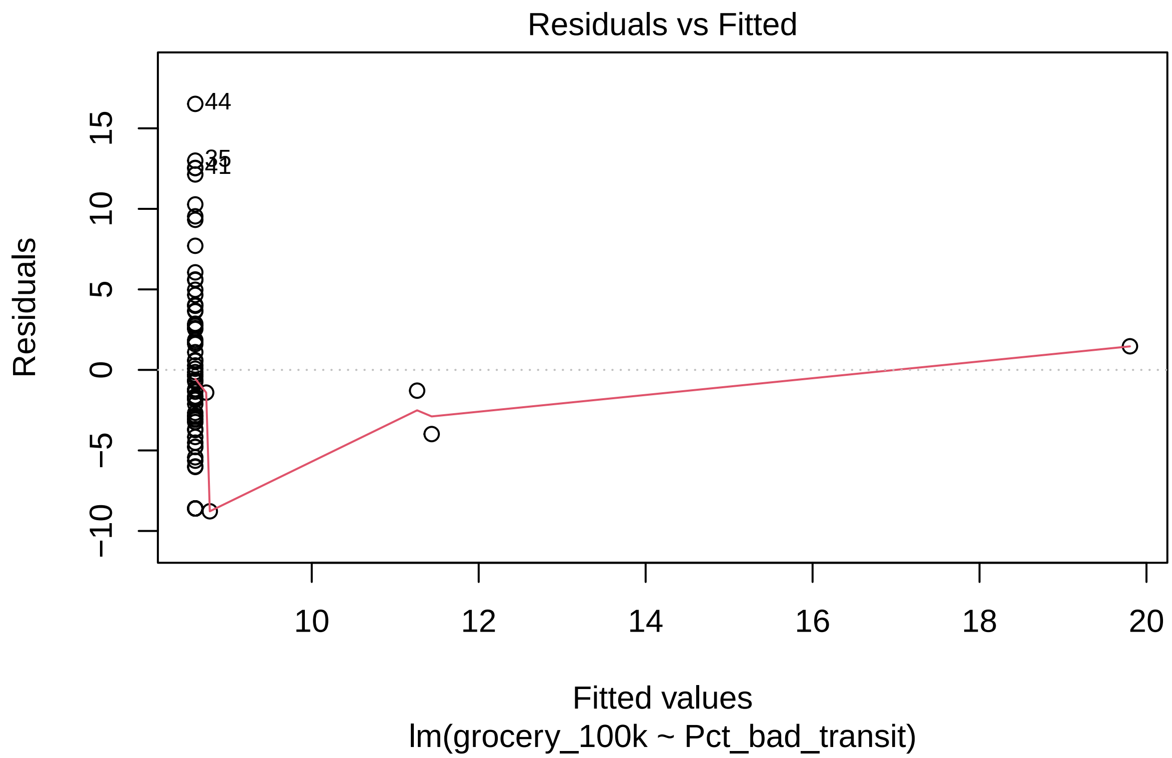

```r
summary(lm_hisp)
```

```
##
## Call:
## lm(formula = grocery_100k ~ Pct_hispanic, data = chicago_sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.179  -4.546  -1.062   3.684  15.173
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.63761    0.93055   8.208  4.8e-12 ***
## Pct_hispanic  0.04573    0.02500   1.829   0.0714 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.85 on 75 degrees of freedom
## Multiple R-squared:  0.04269,    Adjusted R-squared:  0.02993
## F-statistic: 3.345 on 1 and 75 DF,  p-value: 0.0714
```

```r
ggplot(data = chicago_sf) +
  geom_point(aes(x = Pct_bad_transit, y = grocery_100k))
```




```
lm_transit <- lm(grocery_100k ~ Pct_bad_transit, data = chicago_sf)
plot(lm_transit, 1)
```




```r
summary(lm_transit)
```

```
##
## Call:
## lm(formula = grocery_100k ~ Pct_bad_transit, data = chicago_sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7785 -3.6759 -0.7345  2.8683 16.5167
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.6042     0.6728  12.789   <2e-16 ***
## Pct_bad_transit  43.5728    21.6998   2.008   0.0482 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.824 on 75 degrees of freedom
## Multiple R-squared:  0.05102,    Adjusted R-squared:  0.03836
## F-statistic: 4.032 on 1 and 75 DF,  p-value: 0.04825
```

Variable Selection

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
model0 <- lm(grocery_100k ~ 1, data = chicago_sf)
step.for <- stepAIC(model0, direction = "forward", trace = FALSE)
summary(step.for)
```

```
##
## Call:
## lm(formula = grocery_100k ~ 1, data = chicago_sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8249 -3.4612 -0.8909  3.4167 16.2960
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.8249     0.6769   13.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.94 on 76 degrees of freedom
```

```r
model1 <- lm(grocery_100k ~ Pop_2020 + Med_income + Per_cap_income + Pct_bad_transit + Pct_not_walkable
summary(model1)
```

```
##
## Call:
## lm(formula = grocery_100k ~ Pop_2020 + Med_income + Per_cap_income +
##     Pct_bad_transit + Pct_not_walkable + Pct_white + Pct_asian +
##     Pct_black + Pct_hispanic + Pct_other + Pct_unemployed + Pct_poverty +
##     Pct_no_vehicle, data = chicago_sf)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.957 -3.300 -0.238  3.021 12.793
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.258e+01  1.462e+01   0.861  0.39273
## Pop_2020         -1.151e-05  3.425e-05  -0.336  0.73787
## Med_income        1.045e-04  1.154e-04   0.906  0.36831
## Per_cap_income   -1.283e-04  1.497e-04  -0.857  0.39458
## Pct_bad_transit   7.415e+01  2.429e+01   3.052  0.00332 **
## Pct_not_walkable  2.111e+00  6.980e+00   0.302  0.76334
```

```
## Pct_white        -1.441e-01  1.456e-01  -0.990  0.32617
## Pct_asian         1.099e-01  1.422e-01   0.773  0.44229
## Pct_black        -2.454e-02  1.353e-01  -0.181  0.85667
## Pct_hispanic      9.267e-03  1.421e-01   0.065  0.94820
## Pct_other         3.633e-01  6.134e-01   0.592  0.55582
## Pct_unemployed   -2.533e-01  5.536e-01  -0.458  0.64878
## Pct_poverty      -7.835e-01  3.756e-01  -2.086  0.04102 *
## Pct_no_vehicle    6.707e-01  3.063e-01   2.190  0.03222 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.406 on 63 degrees of freedom
## Multiple R-squared:  0.3132, Adjusted R-squared:  0.1715
## F-statistic:  2.21 on 13 and 63 DF,  p-value: 0.01904
```

```
step.both <- stepAIC(model0, direction = "both", trace = FALSE)
summary(step.both)
```

```
##
## Call:
## lm(formula = grocery_100k ~ 1, data = chicago_sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8249 -3.4612 -0.8909  3.4167 16.2960
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.8249     0.6769   13.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.94 on 76 degrees of freedom
```

```
step(model1, direction = "both")
```

```
## Start:  AIC=272.44
## grocery_100k ~ Pop_2020 + Med_income + Per_cap_income + Pct_bad_transit +
##     Pct_not_walkable + Pct_white + Pct_asian + Pct_black + Pct_hispanic +
##     Pct_other + Pct_unemployed + Pct_poverty + Pct_no_vehicle
##
##                    Df Sum of Sq    RSS    AIC
## - Pct_hispanic      1     0.124 1841.5 270.44
## - Pct_black         1     0.961 1842.4 270.48
## - Pct_not_walkable  1     2.673 1844.1 270.55
## - Pop_2020          1     3.303 1844.7 270.57
## - Pct_unemployed    1     6.122 1847.5 270.69
## - Pct_other         1    10.252 1851.7 270.86
## - Pct_asian         1    17.474 1858.9 271.16
## - Per_cap_income    1    21.478 1862.9 271.33
## - Med_income        1    24.000 1865.4 271.43
## - Pct_white         1    28.623 1870.0 271.62
## <none>                          1841.4 272.44
```

```
## - Pct_poverty       1   127.208 1968.6 275.58
## - Pct_no_vehicle    1   140.206 1981.6 276.09
## - Pct_bad_transit   1   272.313 2113.7 281.06
##
## Step:  AIC=270.44
## grocery_100k ~ Pop_2020 + Med_income + Per_cap_income + Pct_bad_transit +
##     Pct_not_walkable + Pct_white + Pct_asian + Pct_black + Pct_other +
##     Pct_unemployed + Pct_poverty + Pct_no_vehicle
##
##                    Df Sum of Sq    RSS    AIC
## - Pct_not_walkable  1     2.979 1844.5 268.56
## - Pop_2020          1     3.184 1844.7 268.57
## - Pct_unemployed    1     6.043 1847.6 268.69
## - Pct_other         1    10.129 1851.7 268.86
## - Pct_black         1    18.967 1860.5 269.23
## - Per_cap_income    1    25.804 1867.3 269.51
## - Med_income        1    25.961 1867.5 269.52
## <none>                          1841.5 270.44
## - Pct_asian         1    60.821 1902.4 270.94
## + Pct_hispanic      1     0.124 1841.4 272.44
## - Pct_poverty       1   140.075 1981.6 274.09
## - Pct_white         1   144.600 1986.1 274.26
## - Pct_no_vehicle    1   144.953 1986.5 274.27
## - Pct_bad_transit   1   292.891 2134.4 279.81
##
## Step:  AIC=268.57
## grocery_100k ~ Pop_2020 + Med_income + Per_cap_income + Pct_bad_transit +
##     Pct_white + Pct_asian + Pct_black + Pct_other + Pct_unemployed +
##     Pct_poverty + Pct_no_vehicle
##
##                    Df Sum of Sq    RSS    AIC
## - Pop_2020          1      4.09 1848.6 266.74
## - Pct_unemployed    1      4.15 1848.7 266.74
## - Pct_other         1      7.96 1852.5 266.90
## - Pct_black         1     20.39 1864.9 267.41
## - Per_cap_income    1     22.86 1867.4 267.51
## - Med_income        1     23.03 1867.5 267.52
## <none>                          1844.5 268.56
## - Pct_asian         1     62.32 1906.8 269.12
## + Pct_not_walkable  1      2.98 1841.5 270.44
## + Pct_hispanic      1      0.43 1844.1 270.55
## - Pct_poverty       1    137.31 1981.8 272.09
## - Pct_no_vehicle    1    145.42 1989.9 272.41
## - Pct_white         1    150.42 1995.0 272.60
## - Pct_bad_transit   1    335.81 2180.3 279.44
##
## Step:  AIC=266.74
## grocery_100k ~ Med_income + Per_cap_income + Pct_bad_transit +
##     Pct_white + Pct_asian + Pct_black + Pct_other + Pct_unemployed +
##     Pct_poverty + Pct_no_vehicle
##
##                    Df Sum of Sq    RSS    AIC
## - Pct_unemployed    1      3.43 1852.0 264.88
## - Pct_other         1      8.20 1856.8 265.08
```

```
## - Pct_black         1     20.14 1868.8 265.57
## - Per_cap_income    1     23.71 1872.3 265.72
## - Med_income        1     26.23 1874.8 265.82
## <none>                          1848.6 266.74
## - Pct_asian         1     71.17 1919.8 267.64
## + Pop_2020          1      4.09 1844.5 268.56
## + Pct_not_walkable  1      3.89 1844.7 268.57
## + Pct_hispanic      1      0.15 1848.5 268.73
## - Pct_poverty       1    142.83 1991.5 270.47
## - Pct_no_vehicle    1    145.38 1994.0 270.56
## - Pct_white         1    151.84 2000.5 270.81
## - Pct_bad_transit   1    336.35 2185.0 277.61
##
## Step:  AIC=264.88
## grocery_100k ~ Med_income + Per_cap_income + Pct_bad_transit +
##     Pct_white + Pct_asian + Pct_black + Pct_other + Pct_poverty +
##     Pct_no_vehicle
##
##                    Df Sum of Sq    RSS    AIC
## - Pct_other         1     10.39 1862.4 263.31
## - Per_cap_income    1     21.64 1873.7 263.77
## - Med_income        1     26.18 1878.2 263.96
## - Pct_black         1     41.95 1894.0 264.60
## <none>                          1852.0 264.88
## - Pct_asian         1     76.73 1928.8 266.00
## + Pct_unemployed    1      3.43 1848.6 266.74
## + Pop_2020          1      3.38 1848.7 266.74
## + Pct_not_walkable  1      1.71 1850.3 266.81
## + Pct_hispanic      1      0.04 1852.0 266.88
## - Pct_poverty       1    141.29 1993.3 268.54
## - Pct_no_vehicle    1    142.24 1994.3 268.58
## - Pct_white         1    149.65 2001.7 268.86
## - Pct_bad_transit   1    334.45 2186.5 275.66
##
## Step:  AIC=263.31
## grocery_100k ~ Med_income + Per_cap_income + Pct_bad_transit +
##     Pct_white + Pct_asian + Pct_black + Pct_poverty + Pct_no_vehicle
##
##                    Df Sum of Sq    RSS    AIC
## - Per_cap_income    1     19.88 1882.3 262.13
## - Med_income        1     21.99 1884.4 262.21
## - Pct_black         1     32.67 1895.1 262.65
## <none>                          1862.4 263.31
## - Pct_asian         1     83.63 1946.1 264.69
## + Pct_other         1     10.39 1852.0 264.88
## + Pct_unemployed    1      5.62 1856.8 265.08
## + Pop_2020          1      3.43 1859.0 265.17
## + Pct_hispanic      1      0.18 1862.2 265.30
## + Pct_not_walkable  1      0.07 1862.4 265.31
## - Pct_no_vehicle    1    150.31 2012.7 267.29
## - Pct_white         1    154.98 2017.4 267.46
## - Pct_poverty       1    155.42 2017.9 267.48
## - Pct_bad_transit   1    324.07 2186.5 273.66
##
```

```
## Step:  AIC=262.13
## grocery_100k ~ Med_income + Pct_bad_transit + Pct_white + Pct_asian +
##     Pct_black + Pct_poverty + Pct_no_vehicle
##
##                   Df Sum of Sq    RSS    AIC
## - Med_income       1     2.974 1885.3 260.25
## - Pct_black        1    38.461 1920.8 261.68
## <none>                         1882.3 262.13
## - Pct_asian        1    78.983 1961.3 263.29
## + Per_cap_income   1    19.877 1862.4 263.31
## + Pct_other        1     8.618 1873.7 263.77
## + Pop_2020         1     4.385 1877.9 263.95
## + Pct_unemployed   1     2.784 1879.5 264.01
## + Pct_hispanic     1     1.103 1881.2 264.08
## + Pct_not_walkable  1     0.509 1881.8 264.11
## - Pct_poverty      1   137.547 2019.9 265.56
## - Pct_white        1   150.443 2032.8 266.05
## - Pct_no_vehicle   1   207.023 2089.3 268.16
## - Pct_bad_transit  1   305.044 2187.3 271.69
##
## Step:  AIC=260.25
## grocery_100k ~ Pct_bad_transit + Pct_white + Pct_asian + Pct_black +
##     Pct_poverty + Pct_no_vehicle
##
##                   Df Sum of Sq    RSS    AIC
## - Pct_black        1    35.847 1921.1 259.70
## <none>                         1885.3 260.25
## - Pct_asian        1    80.978 1966.3 261.49
## + Pct_other        1     6.660 1878.6 261.98
## + Pop_2020         1     5.538 1879.8 262.02
## + Pct_unemployed   1     3.831 1881.5 262.09
## + Med_income       1     2.974 1882.3 262.13
## + Per_cap_income   1     0.863 1884.4 262.21
## + Pct_not_walkable  1     0.674 1884.6 262.22
## + Pct_hispanic     1     0.497 1884.8 262.23
## - Pct_no_vehicle   1   240.112 2125.4 267.48
## - Pct_white        1   260.061 2145.3 268.20
## - Pct_poverty      1   275.631 2160.9 268.75
## - Pct_bad_transit  1   305.978 2191.3 269.83
##
## Step:  AIC=259.7
## grocery_100k ~ Pct_bad_transit + Pct_white + Pct_asian + Pct_poverty +
##     Pct_no_vehicle
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         1921.1 259.70
## + Pct_black        1     35.85 1885.3 260.25
## + Pct_hispanic     1     35.73 1885.4 260.25
## + Pct_unemployed   1     20.92 1900.2 260.86
## + Per_cap_income   1      9.34 1911.8 261.32
## + Pop_2020         1      2.09 1919.0 261.62
## + Pct_not_walkable  1      0.59 1920.5 261.68
## + Med_income       1      0.36 1920.8 261.68
## + Pct_other        1      0.35 1920.8 261.69
```

```
## - Pct_asian         1    156.47 2077.6 263.73
## - Pct_white         1    224.50 2145.6 266.21
## - Pct_no_vehicle    1    226.51 2147.6 266.28
## - Pct_bad_transit   1    318.01 2239.1 269.49
## - Pct_poverty       1    417.52 2338.7 272.84
```

```
##
## Call:
## lm(formula = grocery_100k ~ Pct_bad_transit + Pct_white + Pct_asian +
##     Pct_poverty + Pct_no_vehicle, data = chicago_sf)
##
## Coefficients:
##     (Intercept)  Pct_bad_transit        Pct_white        Pct_asian
##        14.23351         69.90720         -0.09268          0.14381
##     Pct_poverty   Pct_no_vehicle
##        -0.83536          0.43693
```