

Mini Project 1

Nicolas Fairweather

What is R? What are the basics of R?

R is a free software that allows data scientists to compute statistics and analyze data. Anyone is able to use R by simply downloading the program on their laptop or desktop. Through R, users have the ability to use various functions and codes to perform data wrangling and data visualization. The basics of R include learning what functions and data types to use to get the desired results. R basics also teaches how to create vectors, and sort the dataset being used. The basics of R are the building blocks of communicating statistical data.

Motivation

It is vital for all humans to eat fruits, vegetables, and food with quality ingredients in order to sustain a healthy lifestyle. However, U.S. residents who live in urban and rural areas have less access to healthy food from grocery stores. Many African Americans live in urban communities that are considered “food deserts.” According to the Center for American progress (<https://www.americanprogress.org/issues/race/reports/2020/05/07/484742/health-disparities-race-ethnicity/>) (<https://www.americanprogress.org/issues/race/reports/2020/05/07/484742/health-disparities-race-ethnicity/>), 80% of African Americans are considered overweight or obese and 13.8% of African Americans live with poor health. The leading causes of death among African Americans are heart disease, cancer, and other. Could African American's poor health be due to their eating habits and access to healthy food? This statistical analysis aims to find a correlation between race, location, and access to healthy food. ## Research Questions/Hypothesis “Are African Americans more likely to live in food deserts?” For this project, my hypothesis is that there will be more African Americans that live in “food deserts” ## Data Visualization Data visualization is an art of how to turn numbers into useful knowledge. R Programming lets you learn this art by offering a set of inbuilt functions and libraries to build visualizations and present data. In this project we used the bar and pie graph to display the comparison between the black population in food deserts to that of white, asian, and hispanic. ## Data Wrangling Data wrangling is the process of transforming a “raw” dataset to one that is more organized and easier to work with. The steps to data wrangling included: discovering, cleaning, enriching, and validating. 1. Discovering Data: studying data set and understanding various abbreviations found in the columns and rows is necessary in this process. 2. Structuring: Secondly, the user must know how to organize the data. This includes reshaping and joining tables. 3. Cleaning data: For this step, users must eliminate null variables (N/A) or errors. 4. Enriching: This step refers to the process of adding a new data set to support research/hypothesis. 5. Validating: This step is necessary to confirm the user correctly cleaned the data set, that the findings are consistent, and ensures that the everything is distributed normally.

```
setwd("~/food_access_research_atlas.csv")
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyrr   1.1.3     v stringr 1.4.0
## v readr    1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
## 
##     src, summarize
```

```
## The following objects are masked from 'package:base':
## 
##     format.pval, units
```

```
library(readr)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:Hmisc':
## 
##     subplot
```

```
## The following object is masked from 'package:ggplot2':
## 
##     last_plot
```

```
## The following object is masked from 'package:stats':  
##  
##     filter
```

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

```
library(tidyr)  
library(xts)
```

```
## Loading required package: zoo
```

```
##  
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':  
##  
##     as.Date, as.Date.numeric
```

```
##  
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##     first, last
```

```
library(highcharter)
```

```
## Registered S3 method overwritten by 'quantmod':  
##   method           from  
##   as.zoo.data.frame zoo
```

```
library(knitr)  
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(viridisLite)
```

All the packages needed to be downloaded in order to perform the functions used below.

```
foodacc <- read_csv("food_access_research_atlas.csv")
```

```
##  
## -- Column specification -----  
## cols(  
##   .default = col_double(),  
##   CensusTract = col_character(),  
##   State = col_character(),  
##   County = col_character()  
## )  
## i Use `spec()` for the full column specifications.
```

Reading the data set and assigning it to the variable “foodacc”

```
glimpse(foodacc)
```

```

## Rows: 72,864
## Columns: 147
## $ CensusTract <chr> "01001020100", "01001020200", "01001020300", "010~  

## $ State <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alab~  

## $ County <chr> "Autauga", "Autauga", "Autauga", "Autauga", "Autua~  

## $ Urban <dbl> 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1~  

## $ POP2010 <dbl> 1912, 2170, 3373, 4386, 10766, 3668, 2891, 3081, ~  

## $ OHU2010 <dbl> 693, 743, 1256, 1722, 4082, 1311, 1188, 1074, 369~  

## $ GroupQuartersFlag <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ NUMGQTRS <dbl> 0, 181, 0, 0, 181, 0, 36, 0, 0, 14, 10, 33, 31, 6~  

## $ PCTGQTRS <dbl> 0.0000000, 0.0834101, 0.0000000, 0.0000000, 0.016~  

## $ LILATracts_1And10 <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0~  

## $ LILATracts_halfAnd10 <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0~  

## $ LILATracts_1And20 <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ LILATracts_Vehicle <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0~  

## $ HUNVFlag <dbl> 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1~  

## $ LowIncomeTracts <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0~  

## $ PovertyRate <dbl> 10.0, 18.2, 19.1, 3.3, 8.5, 14.1, 26.4, 13.~  

## $ MedianFamilyIncome <dbl> 74750, 51875, 52905, 68079, 77819, 67218, 43646, ~  

## $ LA1and10 <dbl> 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1~  

## $ LAhalfand10 <dbl> 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1~  

## $ LA1and20 <dbl> 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1~  

## $ LATracts_half <dbl> 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1~  

## $ LATracts1 <dbl> 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1~  

## $ LATracts10 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0~  

## $ LATracts20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ LATractsVehicle_20 <dbl> 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1~  

## $ LAPOP1_10 <dbl> 1357.48094, 483.42968, 1417.87489, 1363.46689, 26~  

## $ LAPOP05_10 <dbl> 1732.22547, 1410.37483, 2764.60413, 3651.06102, 7~  

## $ LAPOP1_20 <dbl> 1357.4809, 483.4297, 1417.8749, 1363.4669, 2643.0~  

## $ LALOWI1_10 <dbl> 322.08461, 145.14142, 696.63680, 409.58784, 623.0~  

## $ LALOWI05_10 <dbl> 411.69075, 475.30603, 1350.17655, 1067.84450, 191~  

## $ LALOWI1_20 <dbl> 322.0846, 145.1414, 696.6368, 409.5878, 623.0766, ~  

## $ lapophalf <dbl> 1732.225, 1410.375, 2764.604, 3651.061, 7778.396, ~  

## $ lapophalfshare <dbl> 0.9059757, 0.6499423, 0.8196277, 0.8324353, 0.722~  

## $ lalowihalf <dbl> 411.6908, 475.3060, 1350.1765, 1067.8445, 1913.45~  

## $ lalowihalfshare <dbl> 0.2153194, 0.2190350, 0.4002895, 0.2434666, 0.177~  

## $ lakidshalf <dbl> 466.4264, 448.1635, 744.8916, 846.6138, 2309.4757~  

## $ lakidshalfshare <dbl> 0.2439469, 0.2065270, 0.2208395, 0.1930264, 0.214~  

## $ laseniorshalf <dbl> 198.8282, 139.3054, 346.2031, 766.5215, 839.8855, ~  

## $ laseniorshalfshare <dbl> 0.10398966, 0.06419603, 0.10263952, 0.17476550, 0~  

## $ lawhitehalf <dbl> 1482.8814, 412.2913, 2114.8514, 3395.0619, 6298.7~  

## $ lawhitehalfshare <dbl> 0.7755656, 0.1899960, 0.6269942, 0.7740679, 0.585~  

## $ lablackhalf <dbl> 184.3191, 945.3178, 528.0501, 170.0655, 1000.6785~  

## $ lablackhalfshare <dbl> 0.09640121, 0.43563032, 0.15655206, 0.03877461, 0~  

## $ laasianhalf <dbl> 12.718121, 4.000000, 9.031083, 14.874754, 208.982~  

## $ laasianhalfshare <dbl> 0.006651737, 0.001843318, 0.002677463, 0.00339141~  

## $ lanhopihalf <dbl> 0.000000, 0.000000, 1.000000, 2.638384, 5.196510, ~  

## $ lanhopihalfshare <dbl> 0.000000000, 0.000000000, 0.000296472, 0.00060154~  

## $ laaianhalf <dbl> 13.969933, 4.717582, 9.964775, 8.033406, 37.99405~  

## $ laaianhalfshare <dbl> 0.007306450, 0.002174001, 0.002954277, 0.00183160~  

## $ laomultirhalf <dbl> 38.33690, 44.04810, 101.70680, 60.38711, 226.8340~
```

```

## $ laomultirhalfshare <dbl> 0.020050678, 0.020298664, 0.030153216, 0.01376815~
## $ lahisphalf <dbl> 39.69502, 34.58791, 76.40862, 61.43778, 276.91459~
## $ lahisphalfshare <dbl> 0.020760994, 0.015939129, 0.022653014, 0.01400770~
## $ lahunvhalf <dbl> 21.556248, 58.628965, 49.139711, 17.525112, 129.5~
## $ lahunvhalfshare <dbl> 0.031105696, 0.078908433, 0.039123973, 0.01017718~
## $ lasnaphalf <dbl> 101.87740, 127.36774, 100.17839, 67.71320, 339.07~
## $ lasnaphalfshare <dbl> 0.14700923, 0.17142361, 0.07975986, 0.03932241, 0~
## $ lapop1 <dbl> 1357.4809, 483.4297, 1417.8749, 1363.4669, 2643.0~
## $ lapop1share <dbl> 0.7099796, 0.2227787, 0.4203602, 0.3108680, 0.245~
## $ lalowi1 <dbl> 322.0846, 145.1414, 696.6368, 409.5878, 623.0766, ~
## $ lalowi1share <dbl> 0.16845430, 0.06688545, 0.20653329, 0.09338528, 0~
## $ lakids1 <dbl> 363.6384, 174.7705, 377.1281, 345.9669, 714.5536, ~
## $ lakids1share <dbl> 0.19018744, 0.08053939, 0.11180793, 0.07887982, 0~
## $ laseniors1 <dbl> 162.49725, 50.97682, 190.00148, 236.52644, 361.67~
## $ laseniors1share <dbl> 0.08498810, 0.02349162, 0.05633012, 0.05392760, 0~
## $ lawhite1 <dbl> 1161.8026, 128.1610, 1168.0237, 1233.2744, 2167.7~
## $ lawhite1share <dbl> 0.60763732, 0.05906037, 0.34628632, 0.28118432, 0~
## $ labblack1 <dbl> 147.48767, 335.44772, 202.48852, 80.94742, 343.21~
## $ labblack1share <dbl> 0.07713790, 0.15458420, 0.06003217, 0.01845586, 0~
## $ laasian1 <dbl> 11.015106, 1.590927, 7.823386, 6.875032, 47.49454~
## $ laasian1share <dbl> 0.005761039, 0.000733146, 0.002319415, 0.00156749~
## $ lanhopi1 <dbl> 0.0000000, 0.0000000, 0.0000000, 2.0000000, 0.953~
## $ lanhopi1share <dbl> 0.0000000000, 0.0000000000, 0.0000000000, 0.00045~
## $ laaian1 <dbl> 9.9965922, 0.2635107, 1.6564795, 3.5399402, 13.56~
## $ laaian1share <dbl> 0.005228343, 0.000121434, 0.000491100, 0.00080710~
## $ laomultir1 <dbl> 27.17903, 17.96653, 37.88277, 36.83005, 70.10428, ~
## $ laomultir1share <dbl> 0.014214974, 0.008279507, 0.011231181, 0.00839718~
## $ lahisp1 <dbl> 29.71554, 11.17359, 31.22827, 29.85954, 85.81376, ~
## $ lahisp1share <dbl> 0.015541598, 0.005149120, 0.009258306, 0.00680792~
## $ lahunv1 <dbl> 9.772855, 21.638337, 13.305612, 8.782160, 44.6570~
## $ lahunv1share <dbl> 0.014102243, 0.029122929, 0.010593640, 0.00509997~
## $ lasnap1 <dbl> 79.54684, 41.70096, 50.26342, 24.40175, 119.94647~
## $ lasnap1share <dbl> 0.11478621, 0.05612512, 0.04001865, 0.01417058, 0~
## $ lapop10 <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00~
## $ lapop10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ lalowi10 <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00~
## $ lalowi10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ lakids10 <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00~
## $ lakids10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ laseniors10 <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00~
## $ laseniors10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ lawhite10 <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00~
## $ lawhite10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ labblack10 <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00~
## $ labblack10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ laasian10 <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00~
## $ laasian10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ lanhopi10 <dbl> 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, ~
## $ lanhopi10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ laaian10 <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ laaian10share <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0~
## $ laomultir10 <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.000~
```

```

## $ laomultir10share <dbl> 0.0000000000, 0.0000000000, 0.0000000000, 0.00000~  

## $ lahisp10 <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.000~  

## $ lahisp10share <dbl> 0.0000000000, 0.0000000000, 0.0000000000, 0.00000000~  

## $ lahunv10 <dbl> 0.0000000000, 0.0000000000, 0.0000000000, 0.00000000~  

## $ lahunv10share <dbl> 0.0000000000, 0.0000000000, 0.0000000000, 0.00000000~  

## $ lasnap10 <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.000~  

## $ lasnap10share <dbl> 0.0000000000, 0.0000000000, 0.0000000000, 0.00000000~  

## $ lapop20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lapop20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lalowi20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lalowi20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lakids20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lakids20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laseniors20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laseniors20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lawhite20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lawhite20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lablack20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lablack20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laasian20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laasian20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lanhopi20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lanhopi20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laaian20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laaian20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laomultir20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ laomultir20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lahisp20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lahisp20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lahunv20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lahunv20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lasnap20 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ lasnap20share <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ TractLOWI <dbl> 448, 763, 1578, 1241, 2692, 1504, 1240, 649, 3762~  

## $ TractKids <dbl> 507, 606, 894, 1015, 3162, 1008, 686, 815, 2909, ~  

## $ TractSeniors <dbl> 221, 214, 439, 904, 1126, 411, 360, 341, 1060, 63~  

## $ TractWhite <dbl> 1622, 888, 2576, 4086, 8666, 2751, 2333, 2690, 87~  

## $ TractBlack <dbl> 217, 1217, 647, 193, 1437, 740, 435, 278, 1387, 6~  

## $ TractAsian <dbl> 14, 5, 17, 18, 296, 9, 13, 16, 61, 20, 3, 2, 6, 1~  

## $ TractNHOPI <dbl> 0, 0, 5, 4, 9, 1, 3, 0, 7, 3, 0, 0, 0, 0, 1, 0, 1~  

## $ TractAIAN <dbl> 14, 5, 11, 11, 48, 10, 11, 27, 49, 14, 18, 14, 64~  

## $ TractOMultir <dbl> 45, 55, 117, 74, 310, 157, 96, 70, 187, 124, 47, ~  

## $ TractHispanic <dbl> 44, 75, 87, 85, 355, 176, 98, 57, 159, 96, 39, 39~  

## $ TractHUNV <dbl> 26, 87, 108, 19, 198, 49, 134, 126, 82, 32, 24, 1~  

## $ TractSNAP <dbl> 112, 202, 120, 82, 488, 118, 218, 115, 367, 394, ~

```

Took a little glimpse to verify the data (There aren't any NA's) if there were the function is.na() could be used to eliminate those answers.

```

foodacc_ <- select(foodacc, c(State, County, POP2010, lapop1, lawhite1, lablack1, laasian1, lahi
sp1, TractWhite, TractBlack, TractAsian, TractHispanic))
foodacc_AL <- filter(foodacc_, State == "Alabama")
foodacc_GA <- filter(foodacc_, State == "Georgia")
foodacc_AL <- filter(foodacc_AL, County == "Jefferson")
foodacc_GA <- filter(foodacc_GA, County == "Fulton")
summary(foodacc_AL)

```

	State	County	POP2010	lapop1
##	Length:163	Length:163	Min. : 931	Min. : 0.0
##	Class :character	Class :character	1st Qu.: 2777	1st Qu.: 272.7
##	Mode :character	Mode :character	Median : 3737	Median :1362.8
##			Mean : 4040	Mean :1848.4
##			3rd Qu.: 4925	3rd Qu.:2829.5
##			Max. :12502	Max. :9780.7
##	lawhite1	lablack1	laasian1	lahisp1
##	Min. : 0.00	Min. : 0.00	Min. : 0.00	Min. : 0.000
##	1st Qu.: 36.24	1st Qu.: 28.53	1st Qu.: 0.00	1st Qu.: 3.322
##	Median : 280.92	Median : 313.93	Median : 3.00	Median : 20.079
##	Mean :1067.82	Mean : 706.16	Mean : 18.52	Mean : 55.169
##	3rd Qu.:1484.85	3rd Qu.: 947.37	3rd Qu.: 15.22	3rd Qu.: 68.393
##	Max. :7174.93	Max. :5104.00	Max. :315.18	Max. :703.000
##	TractWhite	TractBlack	TractAsian	TractHispanic
##	Min. : 6.0	Min. : 8.0	Min. : 0.00	Min. : 3.0
##	1st Qu.: 315.5	1st Qu.: 591.5	1st Qu.: 3.50	1st Qu.: 40.0
##	Median :1707.0	Median :1583.0	Median : 14.00	Median : 94.0
##	Mean :2142.1	Mean :1696.5	Mean : 56.18	Mean : 156.4
##	3rd Qu.:3424.5	3rd Qu.:2528.5	3rd Qu.: 53.00	3rd Qu.: 182.5
##	Max. :9203.0	Max. :5954.0	Max. :738.00	Max. :1498.0

```
summary(foodacc_GA)
```

```

##      State          County        POP2010       lapop1
## Length:204      Length:204      Min.   : 0    Min.   : 0.0
## Class :character Class :character  1st Qu.: 2554  1st Qu.: 0.0
## Mode  :character Mode  :character Median : 4084  Median : 454.1
##                                         Mean   : 4513  Mean   : 1366.0
##                                         3rd Qu.: 5442  3rd Qu.: 1880.2
##                                         Max.   :15841  Max.   :14582.4
##      lawhite1      lablack1      laasian1      lahisp1
## Min.   : 0.00  Min.   : 0.00  Min.   : 0.000  Min.   : 0.00
## 1st Qu.: 0.00  1st Qu.: 0.00  1st Qu.: 0.000  1st Qu.: 0.00
## Median : 40.11  Median : 67.02  Median : 3.129  Median : 13.15
## Mean   : 497.64  Mean   : 758.33  Mean   : 53.733  Mean   : 67.15
## 3rd Qu.: 646.74  3rd Qu.: 424.89  3rd Qu.: 39.127  3rd Qu.: 64.96
## Max.   :9049.62  Max.   :12111.67  Max.   :1165.065  Max.   :1252.00
##      TractWhite     TractBlack     TractAsian     TractHispanic
## Min.   : 0.0  Min.   : 0  Min.   : 0.00  Min.   : 0.00
## 1st Qu.: 131.8 1st Qu.: 534  1st Qu.: 12.75  1st Qu.: 72.75
## Median : 1467.5 Median : 1252  Median : 85.00  Median : 183.00
## Mean   : 2008.3 Mean   : 1988  Mean   : 252.79  Mean   : 355.72
## 3rd Qu.: 3533.2 3rd Qu.: 2489  3rd Qu.: 222.25  3rd Qu.: 379.00
## Max.   :11249.0  Max.   :13399  Max.   :4305.00  Max.   :5268.00

```

The summary function provides summary data related to the data set we selected. The which(is.na) function searches through the data set, finding the outliers with an na value that might affect the calculation. In our code, We are selecting the variables of importance, and filtering the data for only information in Jefferson County, AL and Fulton County, GA.

```
which(is.na(foodacc_AL))
```

```
## integer(0)
```

```
which(is.na(foodacc_GA))
```

```
## integer(0)
```

```
class(foodacc_AL$lawhite1)
```

```
## [1] "numeric"
```

```

foodaccAL_long <- foodacc_AL %>%
  gather(key = "Race",
         value = "la1",
         c(lawhite1, lablack1, laasian1, lahispl))

```

```

foodaccGA_long <- foodacc_GA %>%
  gather(key = "Race",
         value = "la1",
         c(lawhite1, lablack1, laasian1, lahispl))

```

```

foodaccAL_long$Race[foodaccAL_long$Race == "lawhite1"] <- "White"
foodaccAL_long$Race[foodaccAL_long$Race == "lablack1"] <- "AA"
foodaccAL_long$Race[foodaccAL_long$Race == "laasian1"] <- "Asian"
foodaccAL_long$Race[foodaccAL_long$Race == "lahisp1"] <- "Hispanic"
foodaccGA_long$Race[foodaccGA_long$Race == "lawhite1"] <- "White"
foodaccGA_long$Race[foodaccGA_long$Race == "lablack1"] <- "AA"
foodaccGA_long$Race[foodaccGA_long$Race == "laasian1"] <- "Asian"
foodaccGA_long$Race[foodaccGA_long$Race == "lahisp1"] <- "Hispanic"
head(foodaccAL_long)

```

```

## # A tibble: 6 x 10
##   State    County  POP2010 lapop1 TractWhite TractBlack TractAsian TractHispanic
##   <chr>    <chr>     <dbl>   <dbl>      <dbl>      <dbl>      <dbl>
## 1 Alabama Jeffers~    3042    817.      530       2384      25        82
## 2 Alabama Jeffers~    2735   1479.      498       1710      33       623
## 3 Alabama Jeffers~    3338   3311.      256       2927      10       162
## 4 Alabama Jeffers~    2864   2864       83       2684      1         89
## 5 Alabama Jeffers~    2577   1363.      12       2538      1         12
## 6 Alabama Jeffers~    3859    604.       54       3722      0         74
## # ... with 2 more variables: Race <chr>, la1 <dbl>

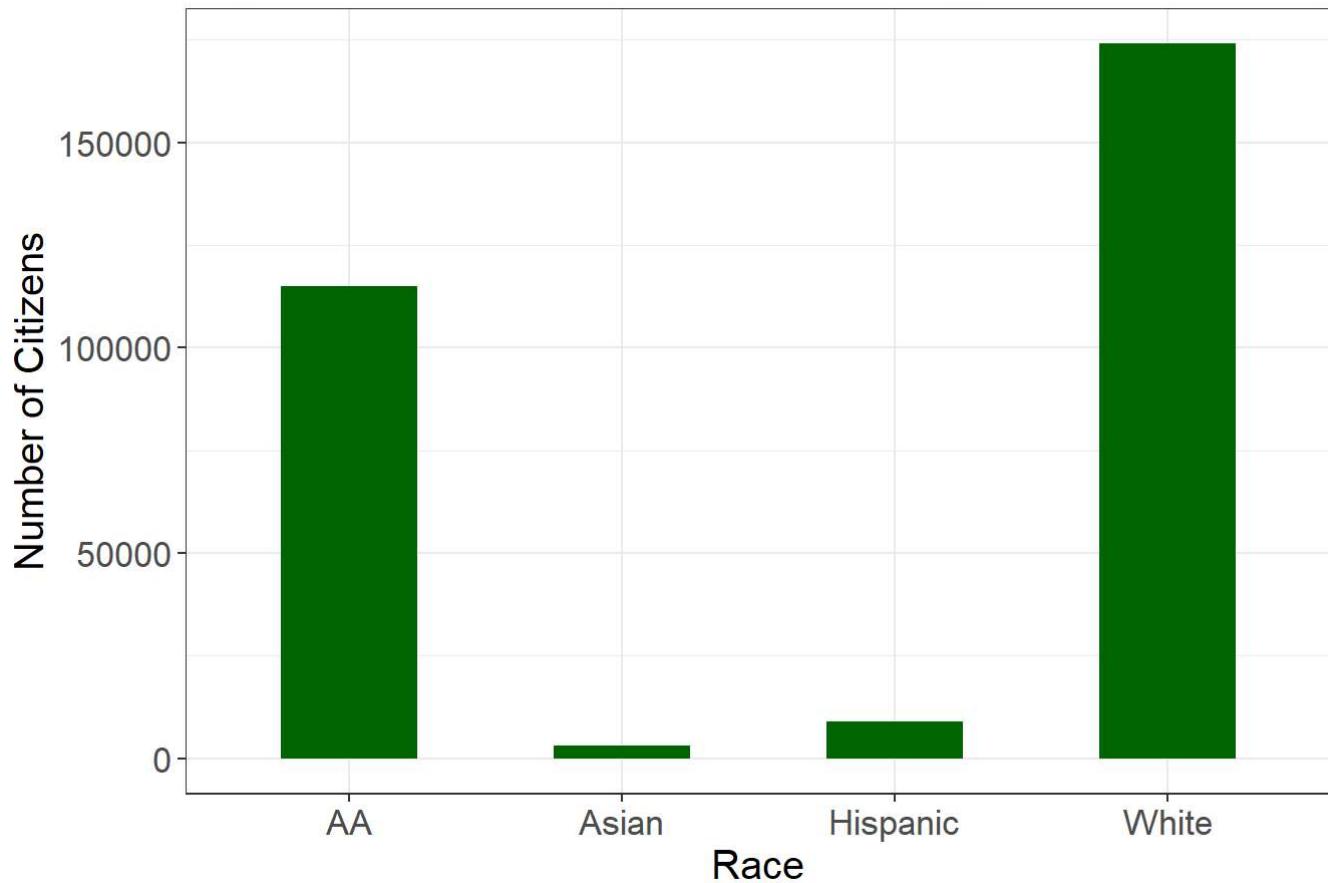
```

```

ggplot(foodaccAL_long, aes(x=Race, y=la1)) +
  geom_bar(stat="identity", width=.5, fill="darkgreen") +
  labs(x="Race",
       y="Number of Citizens",
       title="Populations of Food Desert Residents by Race Jeff. County AL")+
  theme_bw()+
  theme(plot.title = element_text(size=16),
        axis.text.x= element_text(size=13),
        axis.text.y= element_text(size=13),
        axis.title=element_text(size=15))

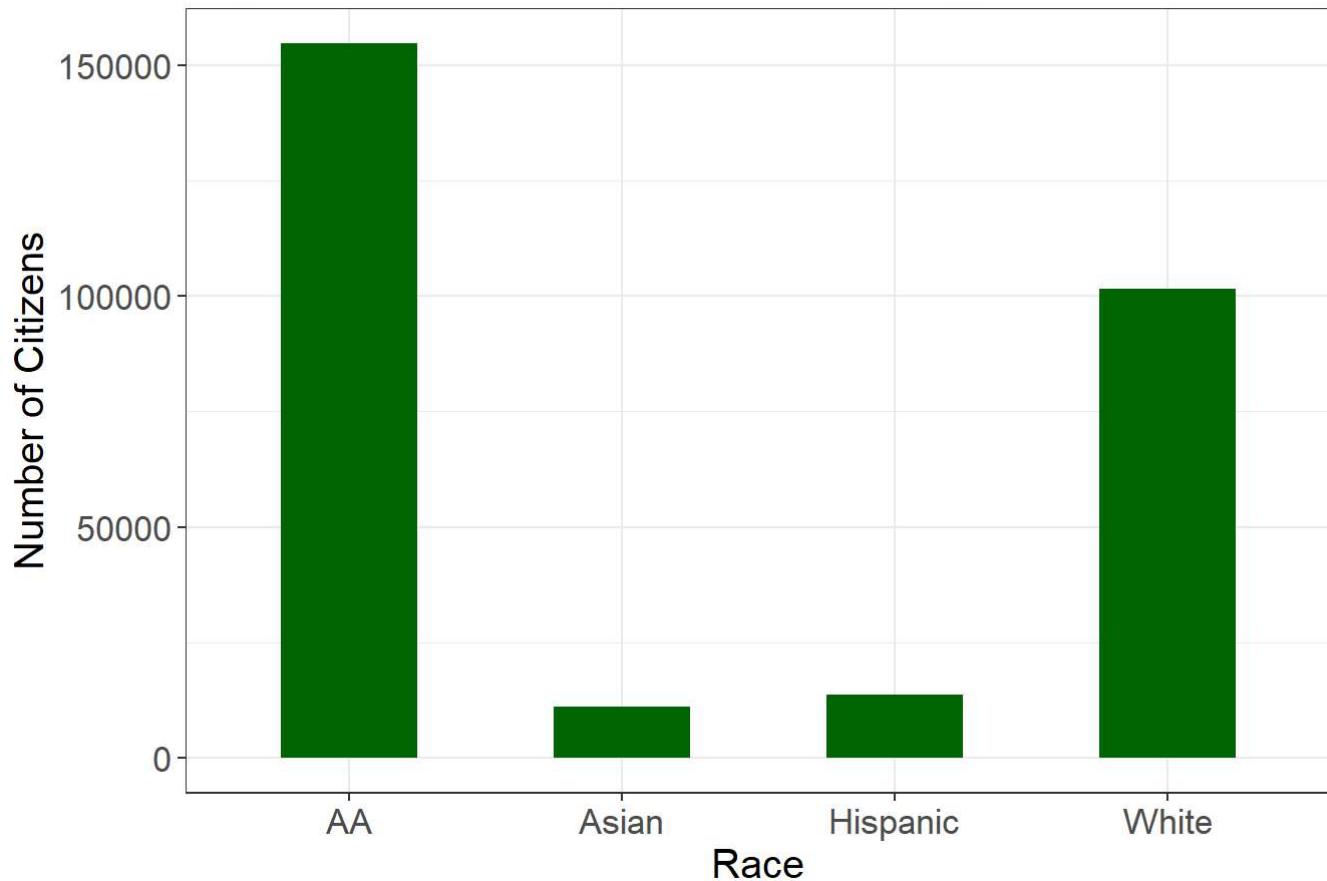
```

Populations of Food Desert Residents by Race Jeff. County



```
ggplot(foodaccGA_long, aes(x=Race, y=la1)) +  
  geom_bar(stat="identity", width=.5, fill="darkgreen") +  
  labs(x="Race",  
       y="Number of Citizens",  
       title="Populations of Food Desert Residents by Race Fult. County GA") +  
  theme_bw() +  
  theme(plot.title = element_text(size=16),  
        axis.text.x = element_text(size=13),  
        axis.text.y = element_text(size=13),  
        axis.title = element_text(size=15))
```

Populations of Food Desert Residents by Race Fult. County



###Pearson Correlation first we will install the packages to make a correlation matrix and perform the test. We have to make another subset that excludes character values (State, County).

```
library(corrplot)
```

```
## corrplot 0.89 loaded
```

```
AL <- foodacc_AL[,c(-1,-2)]  
AL
```

```

## # A tibble: 163 x 10
##   POP2010 lapop1 lawhite1 lablack1 laasian1 lahisp1 TractWhite TractBlack
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 3042     817.    119.     650.     5.80    23.0     530     2384
## 2 2735    1479.    332.     753.    16.7     469.     498     1710
## 3 3338    3311.    253.    2903.     9.86    160.     256     2927
## 4 2864    2864.    83.0    2684.     1       89.0     83      2684
## 5 2577    1363.    6.19    1344.     0       10.4     12      2538
## 6 3859     604.    5.50     589.     0       8.10     54      3722
## 7 5355    5355.    197.    5104.     7.00    40.0     197     5104
## 8 2876    2328.    119.    2002.     0       255.     127     2526
## 9 2181     653.    49.7     543.     2.95    99.5     90      1936
## 10 3189     0       0       0       0       0       169     2921
## # ... with 153 more rows, and 2 more variables: TractAsian <dbl>,
## #   TractHispanic <dbl>

```

```

GA <- foodacc_GA[,c(-1,-2)]
GA

```

```

## # A tibble: 204 x 10
##   POP2010 lapop1 lawhite1 lablack1 laasian1 lahisp1 TractWhite TractBlack
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 4413     546.    514.    11.4     5.35    14.8     4090     115
## 2 5449      0       0       0       0       0       4907     271
## 3 1715      0       0       0       0       0       1456     172
## 4 4687      0       0       0       0       0       3144     960
## 5 5203      0       0       0       0       0       2492    1323
## 6 2794    17.7    12.1    3.32    0.302   1.06      710    1969
## 7 2272      0       0       0       0       0       1144     188
## 8 6315      0       0       0       0       0       4064     582
## 9 3633      0       0       0       0       0       2721     494
## 10 3479     0       0       0       0       0       2529     565
## # ... with 194 more rows, and 2 more variables: TractAsian <dbl>,
## #   TractHispanic <dbl>

```

##Here we will make sure the populations of each race are properly distributed. We know that Caucasian people tend to be the majority in most places and want to make sure we are accounting for that.

```

cor(AL, method = "pearson")

```

```

##          POP2010      lapop1     lawhite1    lablack1   laasian1
## POP2010  1.0000000  0.636937026  0.641743717  0.15328599  0.5197686
## lapop1    0.6369370  1.000000000  0.850419873  0.49842916  0.4776036
## lawhite1   0.6417437  0.850419873  1.000000000 -0.02995085  0.5177609
## lablack1   0.1532860  0.498429157 -0.029950855  1.000000000  0.0193747
## laasian1   0.5197686  0.477603607  0.517760943  0.01937470  1.0000000
## lahisp1    0.1556900  0.410812961  0.271719051  0.26180157  0.2327617
## TractWhite 0.7832196  0.564253944  0.795749146 -0.23895694  0.5194150
## TractBlack  0.1069964 -0.004031546 -0.370960581  0.61129044 -0.1535863
## TractAsian  0.4570751  0.109299328  0.220173557 -0.18000690  0.6742399
## TractHispanic 0.2292807 -0.016125795  0.008584481 -0.08642172  0.1129936
##          lahisp1 TractWhite TractBlack TractAsian TractHispanic
## POP2010    0.15568997  0.7832196  0.106996416  0.45707507  0.229280745
## lapop1     0.41081296  0.5642539 -0.004031546  0.10929933 -0.016125795
## lawhite1   0.27171905  0.7957491 -0.370960581  0.22017356  0.008584481
## lablack1   0.26180157 -0.2389569  0.611290443 -0.18000690 -0.086421724
## laasian1   0.23276165  0.5194150 -0.153586300  0.67423991  0.112993571
## lahisp1    1.00000000  0.1160115 -0.023909102  0.04388892  0.438572965
## TractWhite 0.11601150  1.0000000 -0.527617713  0.48779705  0.136208643
## TractBlack -0.02390910 -0.5276177  1.000000000 -0.23862444 -0.033297323
## TractAsian  0.04388892  0.4877971 -0.238624441  1.000000000  0.301155718
## TractHispanic 0.43857296  0.1362086 -0.033297323  0.30115572  1.000000000

```

```
cor(GA, method = "pearson")
```

```

##          POP2010      lapop1     lawhite1    lablack1   laasian1
## POP2010  1.0000000  0.69118909  0.40860493  0.559359476  0.429952346
## lapop1    0.6911891  1.00000000  0.51706373  0.874605476  0.253360967
## lawhite1   0.4086049  0.51706373  1.000000000 0.044188126  0.393528082
## lablack1   0.5593595  0.87460548  0.04418813  1.000000000  0.004279498
## laasian1   0.4299523  0.25336097  0.39352808  0.004279498  1.000000000
## lahisp1    0.4142620  0.64720110  0.38867160  0.499858426  0.246426465
## TractWhite 0.5375544  0.11651246  0.63288957 -0.230862698  0.424435492
## TractBlack  0.5140730  0.67311828 -0.13555482  0.868798149 -0.092892267
## TractAsian  0.4515157  0.04265868  0.19098269 -0.121584107  0.862905896
## TractHispanic 0.3838432  0.04689120  0.04673888  0.014659962  0.070560188
##          lahisp1 TractWhite TractBlack TractAsian TractHispanic
## POP2010    0.41426201  0.53755440  0.51407297  0.45151572  0.38384319
## lapop1     0.64720110  0.11651246  0.67311828  0.04265868  0.04689120
## lawhite1   0.38867160  0.63288957 -0.13555482  0.19098269  0.04673888
## lablack1   0.49985843 -0.23086270  0.86879815 -0.12158411  0.01465996
## laasian1   0.24642647  0.42443549 -0.09289227  0.86290590  0.07056019
## lahisp1    1.00000000  0.09529925  0.34025918  0.06363516  0.22834068
## TractWhite 0.09529925  1.00000000 -0.42204706  0.51240417  0.30086766
## TractBlack  0.34025918 -0.42204706  1.00000000 -0.18738823  0.01301854
## TractAsian  0.06363516  0.51240417 -0.18738823  1.00000000  0.12897793
## TractHispanic 0.22834068  0.30086766  0.01301854  0.12897793  1.00000000

```

```
library(psych)
```

```
##  
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:Hmisc':  
##  
##     describe
```

```
## The following objects are masked from 'package:ggplot2':  
##  
##     %+%, alpha
```

```
corr.test(AL,method = "pearson")
```

```

## Call:corr.test(x = AL, method = "pearson")
## Correlation matrix
##          POP2010 lapop1 lawhite1 lablack1 laasian1 lahisp1 TractWhite
## POP2010      1.00   0.64    0.64    0.15    0.52    0.16    0.78
## lapop1       0.64   1.00    0.85    0.50    0.48    0.41    0.56
## lawhite1     0.64   0.85    1.00   -0.03    0.52    0.27    0.80
## lablack1     0.15   0.50   -0.03   1.00    0.02    0.26   -0.24
## laasian1     0.52   0.48    0.52    0.02    1.00    0.23    0.52
## lahisp1      0.16   0.41    0.27    0.26    0.23    1.00    0.12
## TractWhite   0.78   0.56    0.80   -0.24    0.52    0.12    1.00
## TractBlack   0.11   0.00   -0.37    0.61   -0.15   -0.02   -0.53
## TractAsian   0.46   0.11    0.22   -0.18    0.67    0.04    0.49
## TractHispanic 0.23  -0.02    0.01   -0.09    0.11    0.44    0.14
##          TractBlack TractAsian TractHispanic
## POP2010      0.11      0.46      0.23
## lapop1       0.00      0.11   -0.02
## lawhite1     -0.37      0.22      0.01
## lablack1     0.61   -0.18   -0.09
## laasian1     -0.15      0.67      0.11
## lahisp1      -0.02      0.04      0.44
## TractWhite   -0.53      0.49      0.14
## TractBlack    1.00   -0.24   -0.03
## TractAsian   -0.24      1.00      0.30
## TractHispanic -0.03      0.30      1.00
## Sample Size
## [1] 163
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##          POP2010 lapop1 lawhite1 lablack1 laasian1 lahisp1 TractWhite
## POP2010      0.00    0.00    0.00    0.80    0.00    0.80    0.00
## lapop1       0.00    0.00    0.00    0.00    0.00    0.00    0.00
## lawhite1     0.00    0.00    0.00    1.00    0.00    0.01    0.00
## lablack1     0.05    0.00    0.70    0.00    1.00    0.02    0.05
## laasian1     0.00    0.00    0.00    0.81    0.00    0.06    0.00
## lahisp1      0.05    0.00    0.00    0.00    0.00    0.00    1.00
## TractWhite   0.00    0.00    0.00    0.00    0.00    0.14    0.00
## TractBlack   0.17    0.96    0.00    0.00    0.05    0.76    0.00
## TractAsian   0.00    0.16    0.00    0.02    0.00    0.58    0.00
## TractHispanic 0.00    0.84    0.91    0.27    0.15    0.00    0.08
##          TractBlack TractAsian TractHispanic
## POP2010      1.00      0.00      0.06
## lapop1       1.00      1.00      1.00
## lawhite1     0.00      0.09      1.00
## lablack1     0.00      0.39      1.00
## laasian1     0.80      0.00      1.00
## lahisp1      1.00      1.00      0.00
## TractWhite   0.00      0.00      1.00
## TractBlack   0.00      0.05      1.00
## TractAsian   0.00      0.00      0.00
## TractHispanic 0.67      0.00      0.00

```

```
##  
## To see confidence intervals of the correlations, print with the short=FALSE option
```

```
corr.test(GA,method = "pearson")
```

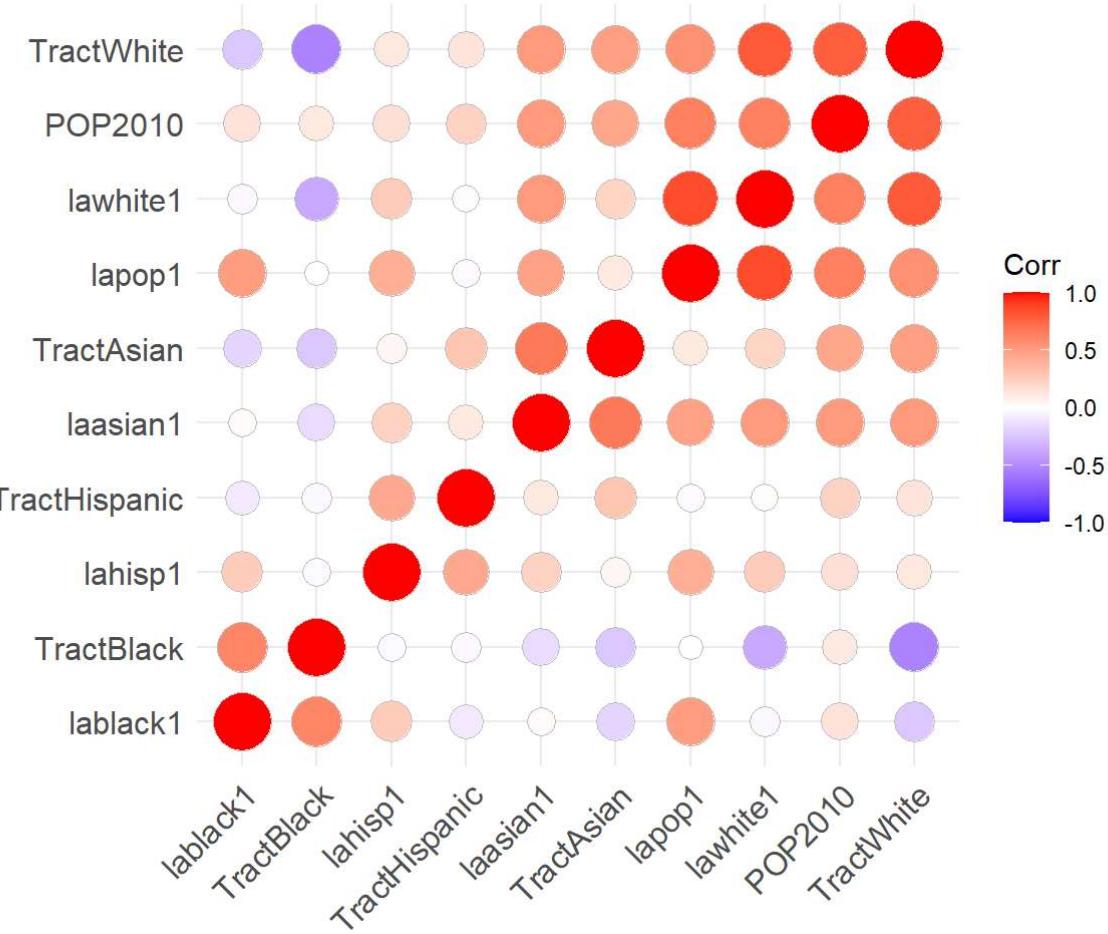
```

## Call:corr.test(x = GA, method = "pearson")
## Correlation matrix
##          POP2010 lapop1 lawhite1 lablack1 laasian1 lahisp1 TractWhite
## POP2010      1.00   0.69    0.41    0.56    0.43   0.41    0.54
## lapop1       0.69   1.00    0.52    0.87    0.25   0.65    0.12
## lawhite1     0.41   0.52    1.00    0.04    0.39   0.39    0.63
## lablack1     0.56   0.87    0.04    1.00    0.00   0.50   -0.23
## laasian1     0.43   0.25    0.39    0.00    1.00   0.25    0.42
## lahisp1      0.41   0.65    0.39    0.50    0.25   1.00    0.10
## TractWhite   0.54   0.12    0.63   -0.23    0.42   0.10    1.00
## TractBlack   0.51   0.67   -0.14    0.87   -0.09   0.34   -0.42
## TractAsian   0.45   0.04    0.19   -0.12    0.86   0.06    0.51
## TractHispanic 0.38   0.05    0.05    0.01    0.07   0.23    0.30
##          TractBlack TractAsian TractHispanic
## POP2010      0.51      0.45      0.38
## lapop1       0.67      0.04      0.05
## lawhite1     -0.14      0.19      0.05
## lablack1     0.87     -0.12      0.01
## laasian1     -0.09      0.86      0.07
## lahisp1      0.34      0.06      0.23
## TractWhite   -0.42      0.51      0.30
## TractBlack   1.00     -0.19      0.01
## TractAsian   -0.19      1.00      0.13
## TractHispanic 0.01      0.13      1.00
## Sample Size
## [1] 204
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##          POP2010 lapop1 lawhite1 lablack1 laasian1 lahisp1 TractWhite
## POP2010      0  0.00   0.00   0.00   0.00   0.00   0.00
## lapop1       0  0.00   0.00   0.00   0.01   0.00   1.00
## lawhite1     0  0.00   0.00   1.00   0.00   0.00   0.00
## lablack1     0  0.00   0.53   0.00   1.00   0.00   0.02
## laasian1     0  0.00   0.00   0.95   0.00   0.01   0.00
## lahisp1      0  0.00   0.00   0.00   0.00   0.00   1.00
## TractWhite   0  0.10   0.00   0.00   0.00   0.18   0.00
## TractBlack   0  0.00   0.05   0.00   0.19   0.00   0.00
## TractAsian   0  0.54   0.01   0.08   0.00   0.37   0.00
## TractHispanic 0  0.51   0.51   0.84   0.32   0.00   0.00
##          TractBlack TractAsian TractHispanic
## POP2010      0.00      0.00      0.00
## lapop1       0.00      1.00      1.00
## lawhite1     0.80      0.11      1.00
## lablack1     0.00      1.00      1.00
## laasian1     1.00      0.00      1.00
## lahisp1      0.00      1.00      0.02
## TractWhite   0.00      0.00      0.00
## TractBlack   0.00      0.12      1.00
## TractAsian   0.01      0.00      0.92
## TractHispanic 0.85      0.07      0.00

```

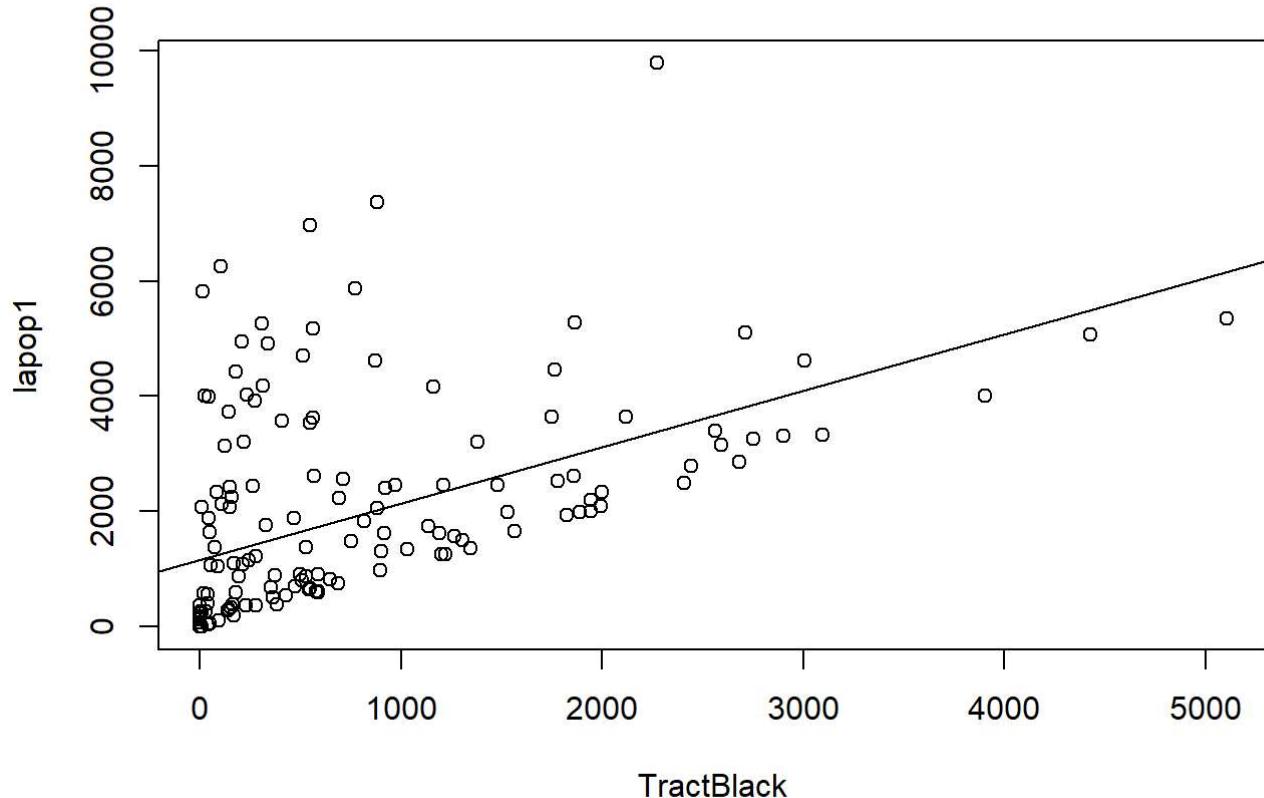
```
##  
## To see confidence intervals of the correlations, print with the short=FALSE option
```

```
library(ggcorrplot)  
ggcorrplot(cor(AL), hc.order = TRUE, method = "circle")
```



```
plot(AL$lablack1, AL$lapop1, main = "Regression for total AA Population on Total Pop 1 mile > from Supermarket",  
     xlab= "TractBlack", ylab= "lapop1") +  
     abline(lm(lapop1 ~ lablack1, data = AL))
```

Regression for total AA Population on Total Pop 1 mile > from Supermarket

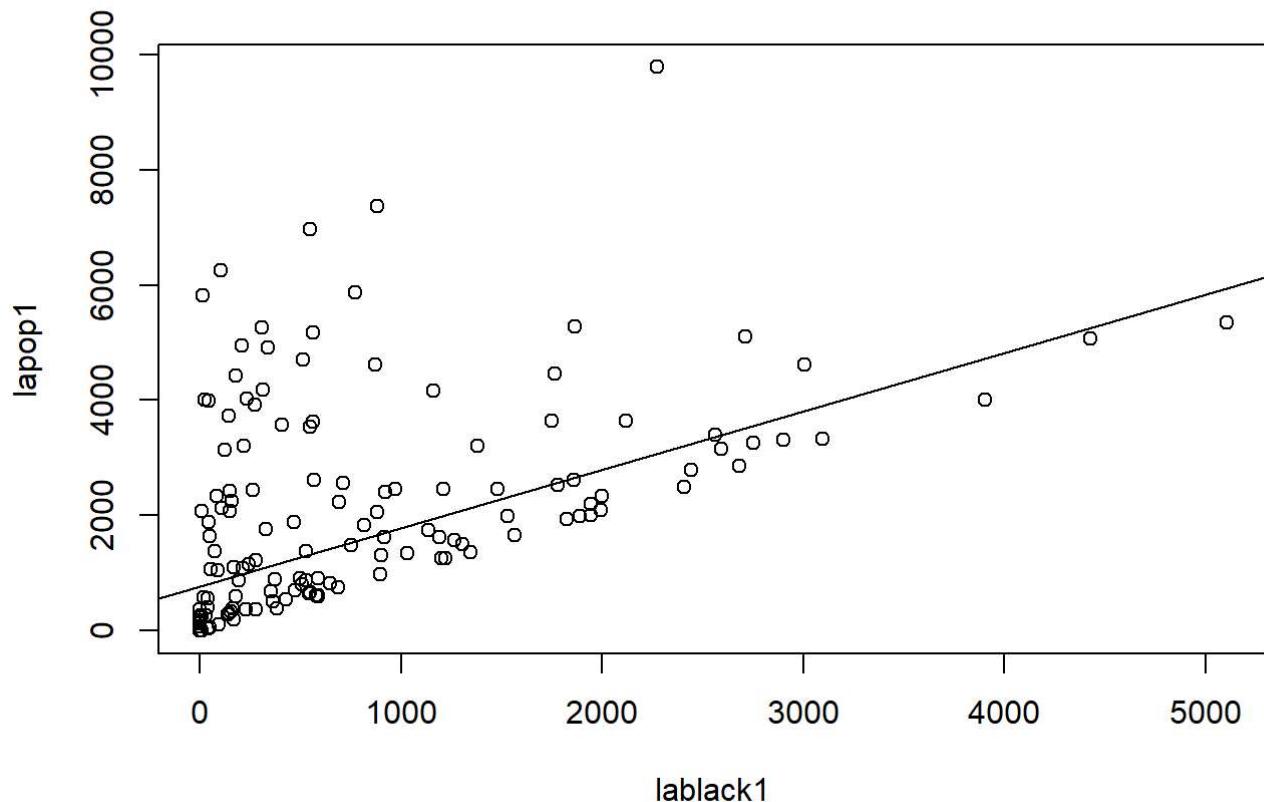


```
## integer(0)
```

```
plot(AL$lablack1, AL$lapop1, main = "Regression for AA population in Food Deserts compared to total population in Food Deserts",
     xlab= "lablack1", ylab= "lapop1")

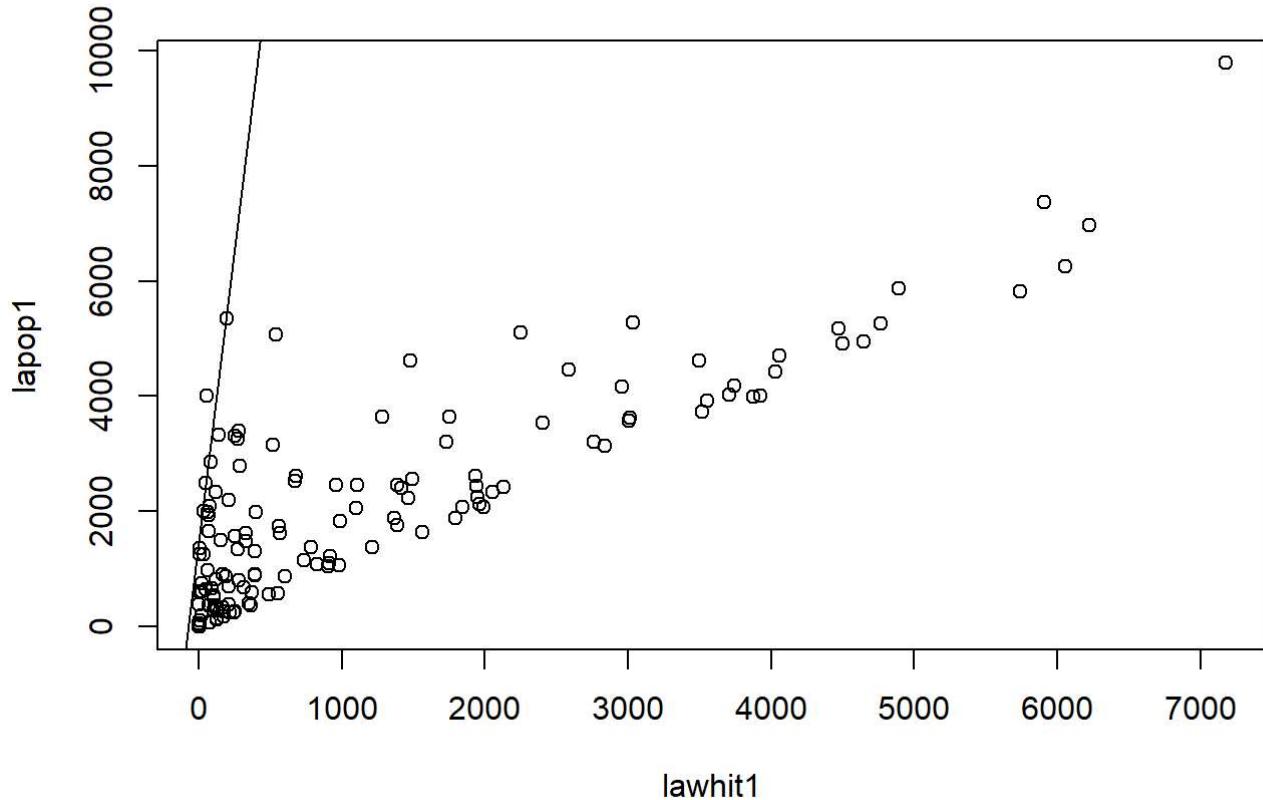
abline(lm(lapop1 ~ lawhite1, data = AL))
```

tion for AA population in Food Deserts compared to total population in Fo



```
plot(AL$lawhite1, AL$lapop1, main = "Regression for White population in Food Deserts compared to  
total population in Food Deserts",  
      xlab= "lawhit1", ylab= "lapop1")  
  
abline(lm(lapop1 ~ laasian1, data = AL))
```

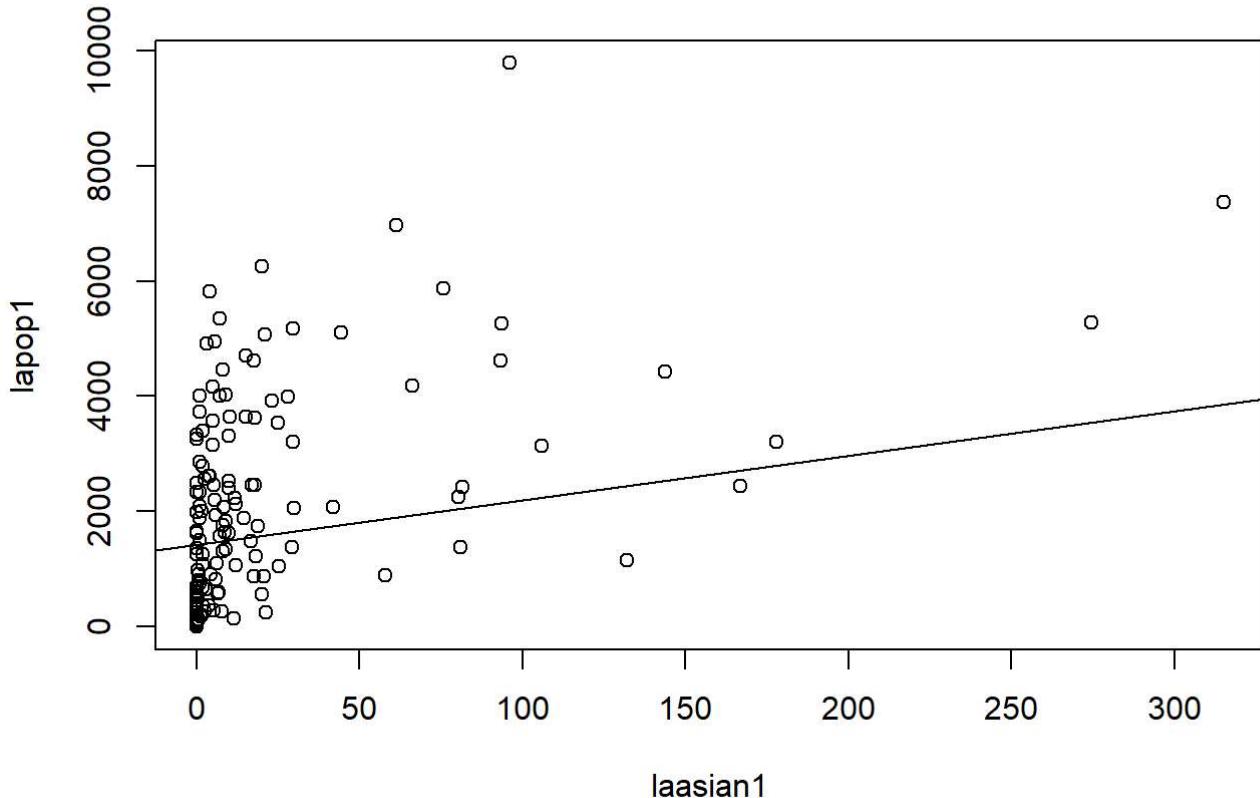
ion for White population in Food Deserts compared to total population in F



```
plot(AL$laasian1, AL$lapop1, main = "Regression for Asian population in Food Deserts compared to total population in Food Deserts",
     xlab= "laasian1", ylab= "lapop1")

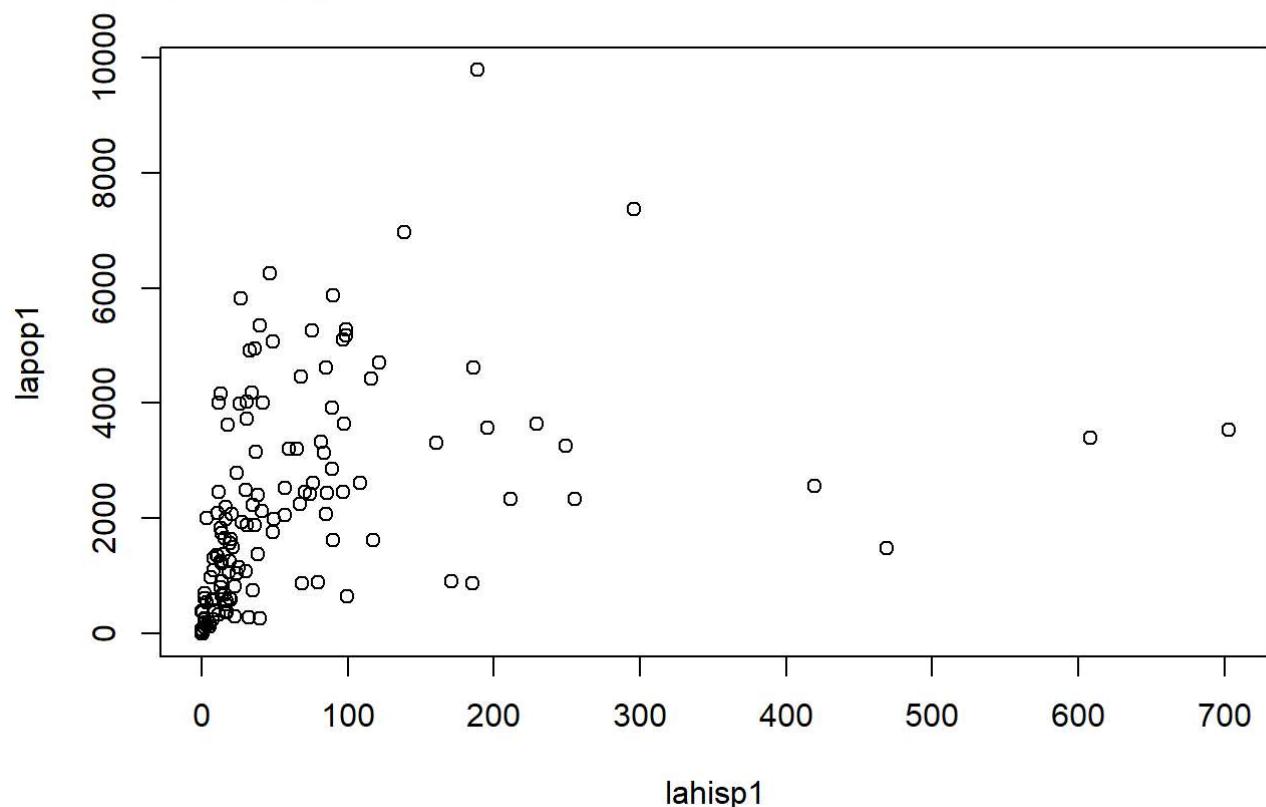
abline(lm(lapop1 ~ lahisp1, data = AL))
```

ion for Asian population in Food Deserts compared to total population in F

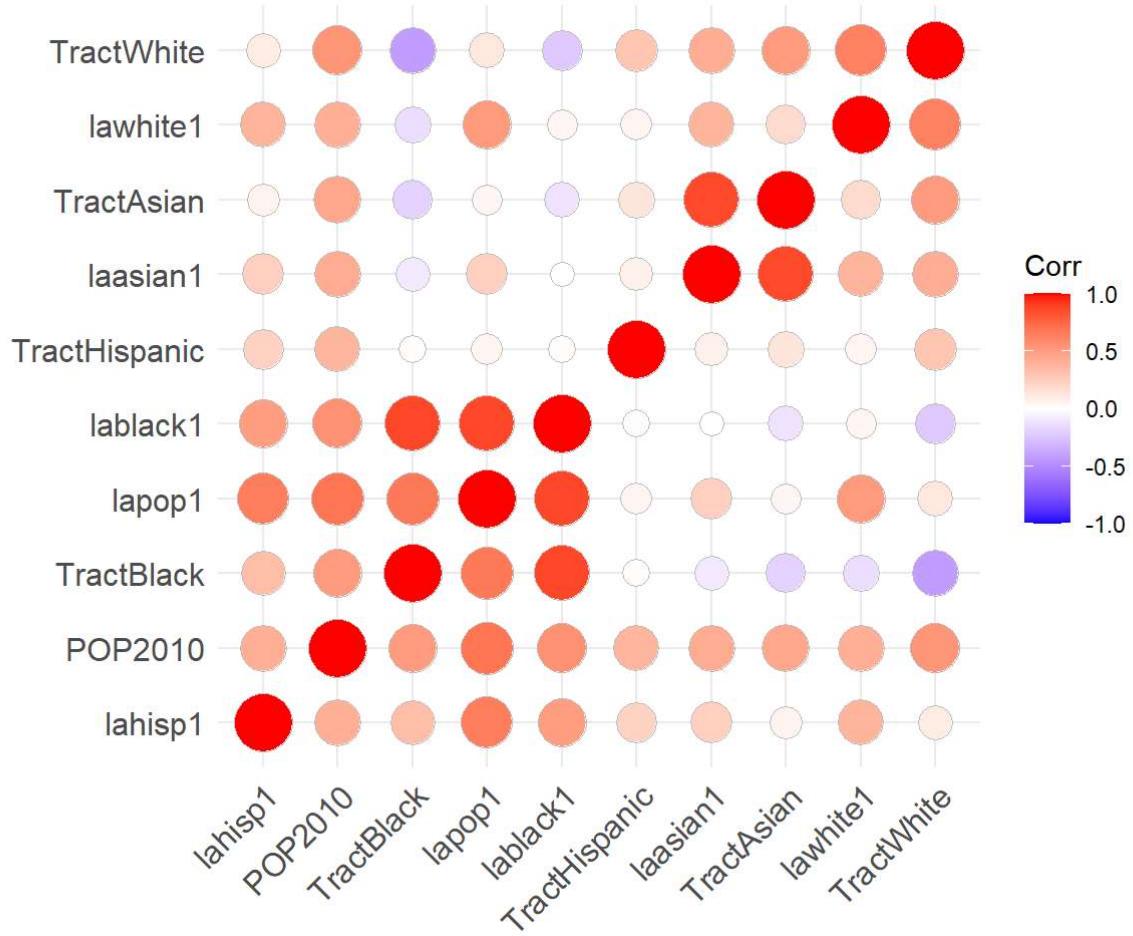


```
plot(AL$lahisp1, AL$lapop1, main = "Regression for Hispanic population in Food Deserts compared  
to total population in Food Deserts",  
      xlab= "lahisp1", ylab= "lapop1")
```

in for Hispanic population in Food Deserts compared to total population in

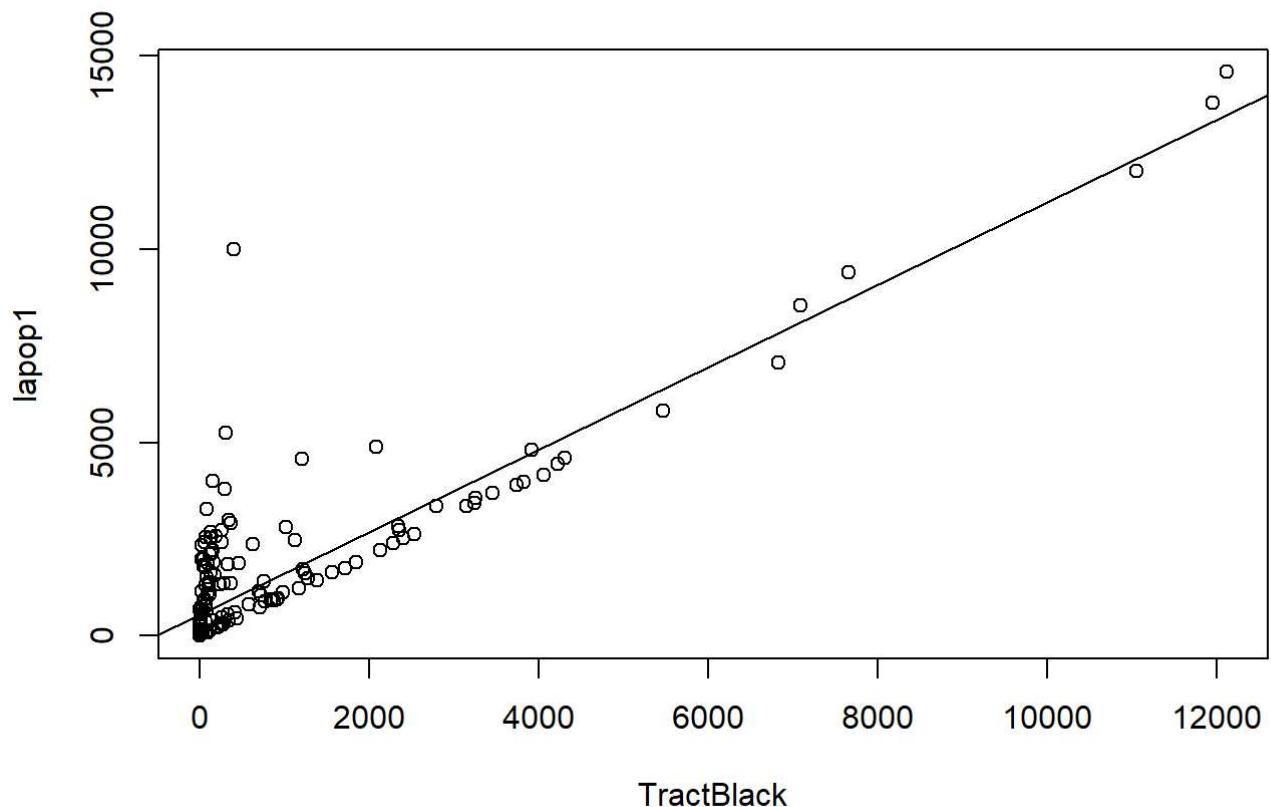


```
ggcorrplot(cor(GA), hc.order = TRUE, method = "circle")
```



```
plot(GA$lablack1, GA$lapop1, main = "Total AA Population on Total Pop 1 mile > from Supermarket  
(Fult.Co)",  
     xlab= "TractBlack", ylab= "lapop1") +  
abline(lm(lapop1 ~ lablack1, data = GA))
```

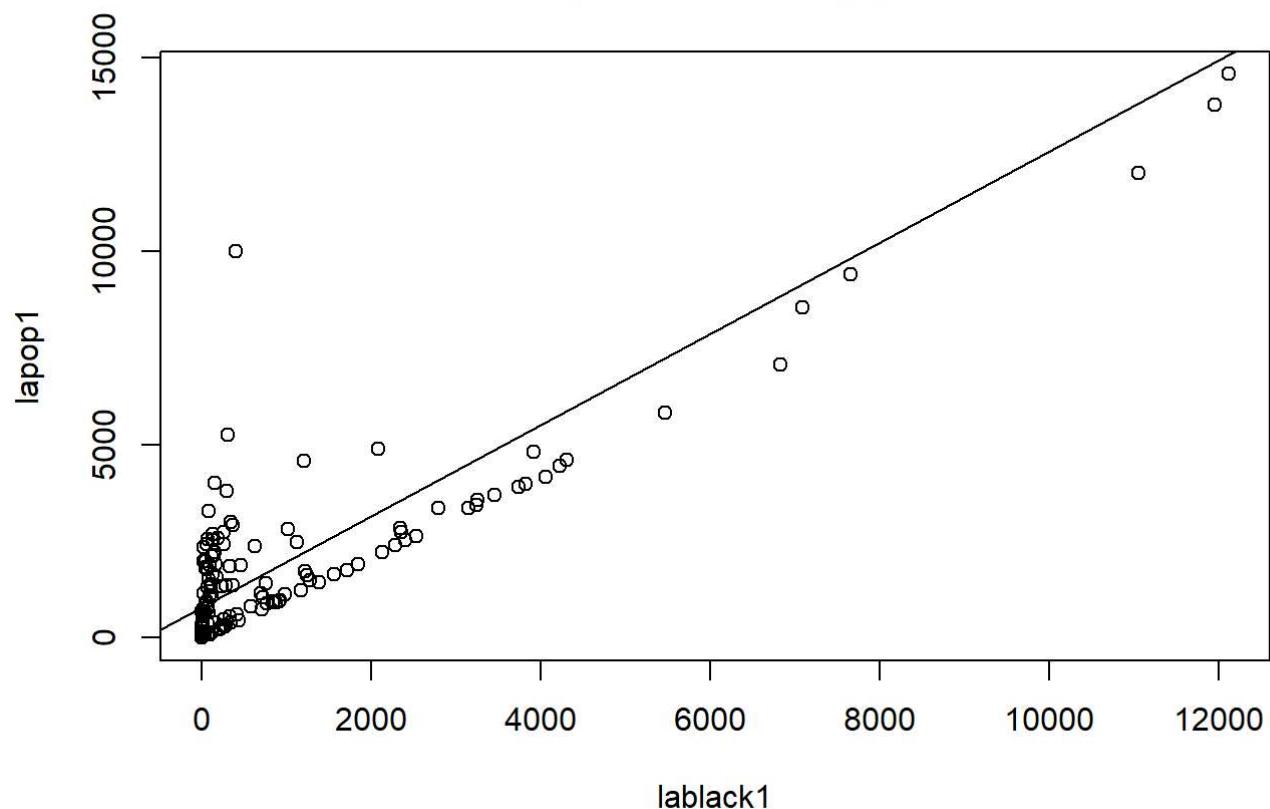
Total AA Population on Total Pop 1 mile > from Supermarket(Fult.Co)



```
## integer(0)
```

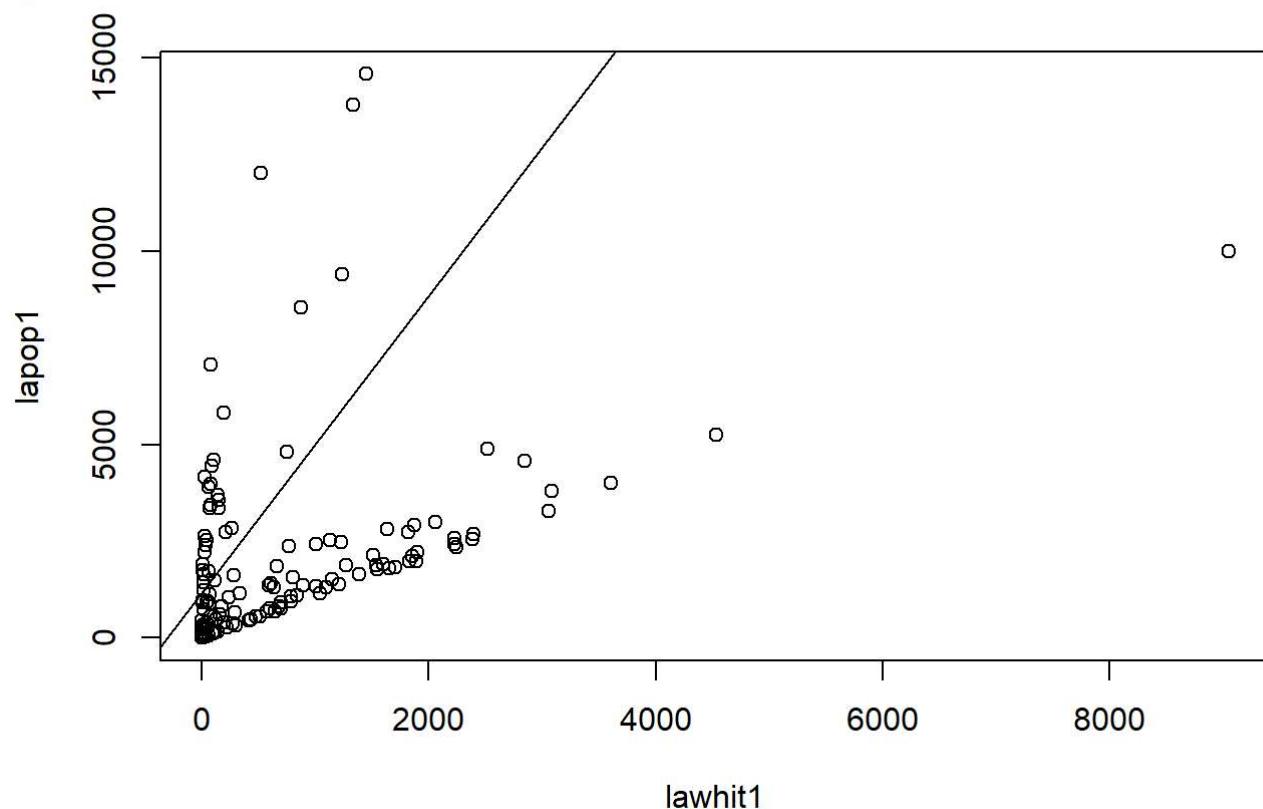
```
plot(GA$lablack1, GA$lapop1, main = "AA population in Food Deserts compared to total population in Food Deserts (Fult.CO)",  
     xlab= "lablack1", ylab= "lapop1")  
  
abline(lm(lapop1 ~ lawhite1, data = GA))
```

Population in Food Deserts compared to total population in Food Deserts (



```
plot(GA$lawhite1, GA$lapop1, main = "White population in Food Deserts compared to total population in Food Deserts (Fult.Co)",  
     xlab= "lawhit1", ylab= "lapop1")  
  
abline(lm(lapop1 ~ laasian1, data = GA))
```

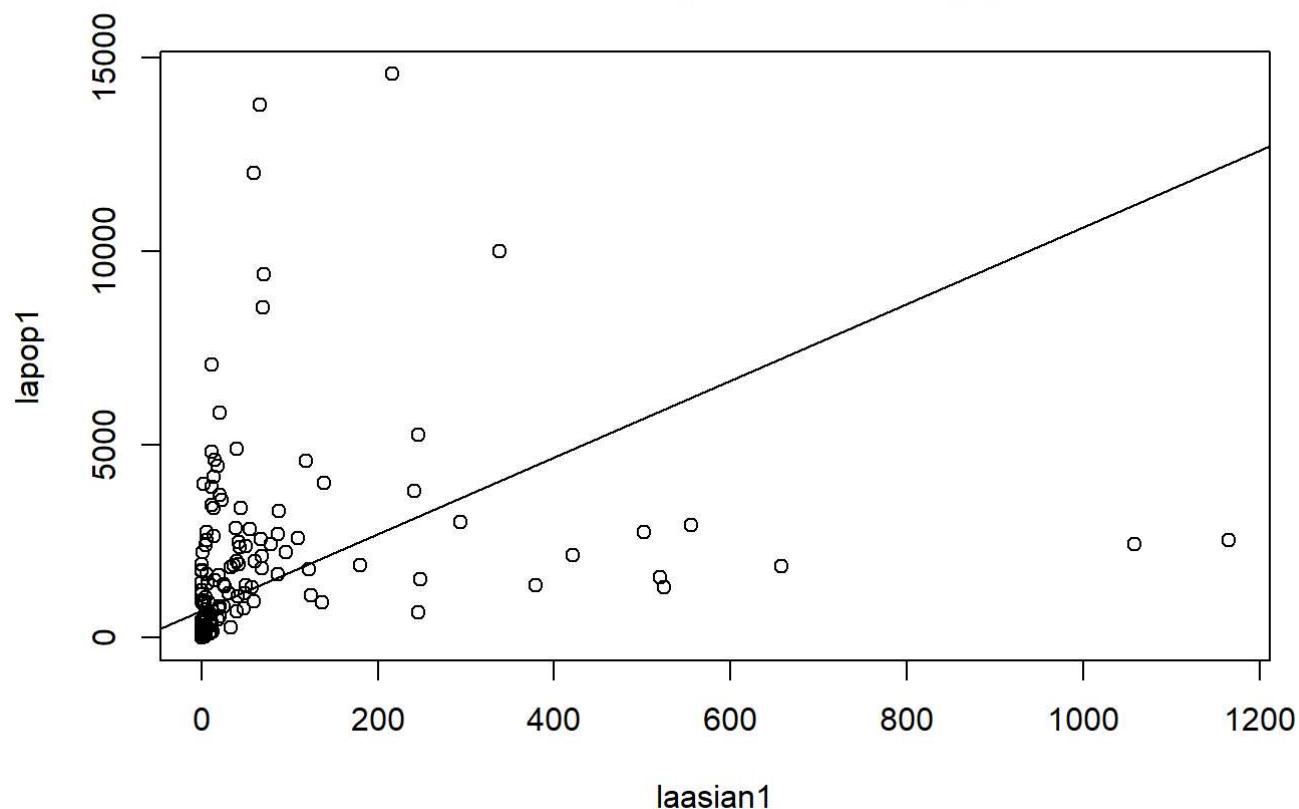
Asian population in Food Deserts compared to total population in Food Deserts



```
plot(GA$laasian1, GA$lapop1, main = "Asian population in Food Deserts compared to total population in Food Deserts",
     xlab= "laasian1", ylab= "lapop1")

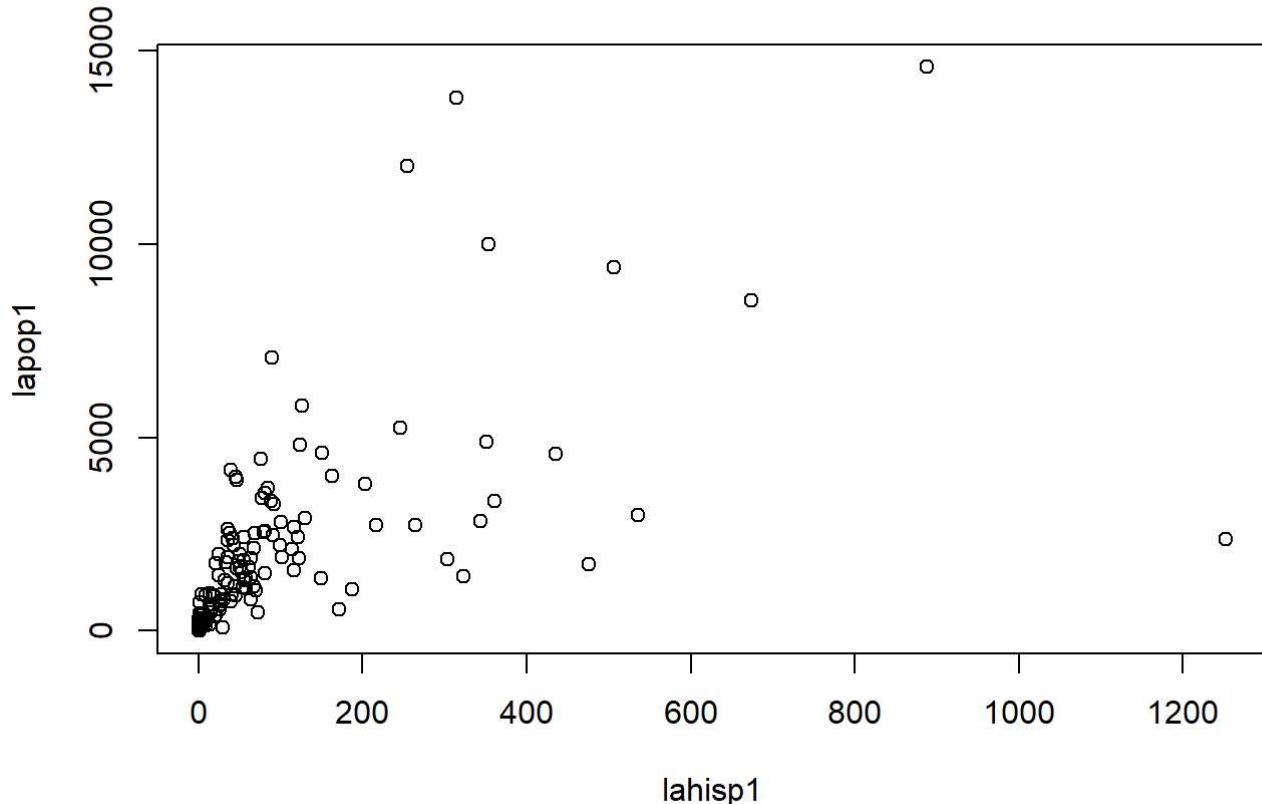
abline(lm(lapop1 ~ lahispl, data = GA))
```

Asian population in Food Deserts compared to total population in Food Deserts



```
plot(GA$lahisp1, GA$lapop1, main = "Regression for Hispanic population in Food Deserts compared to total population in Food Deserts",
     xlab= "lahisp1", ylab= "lapop1")
```

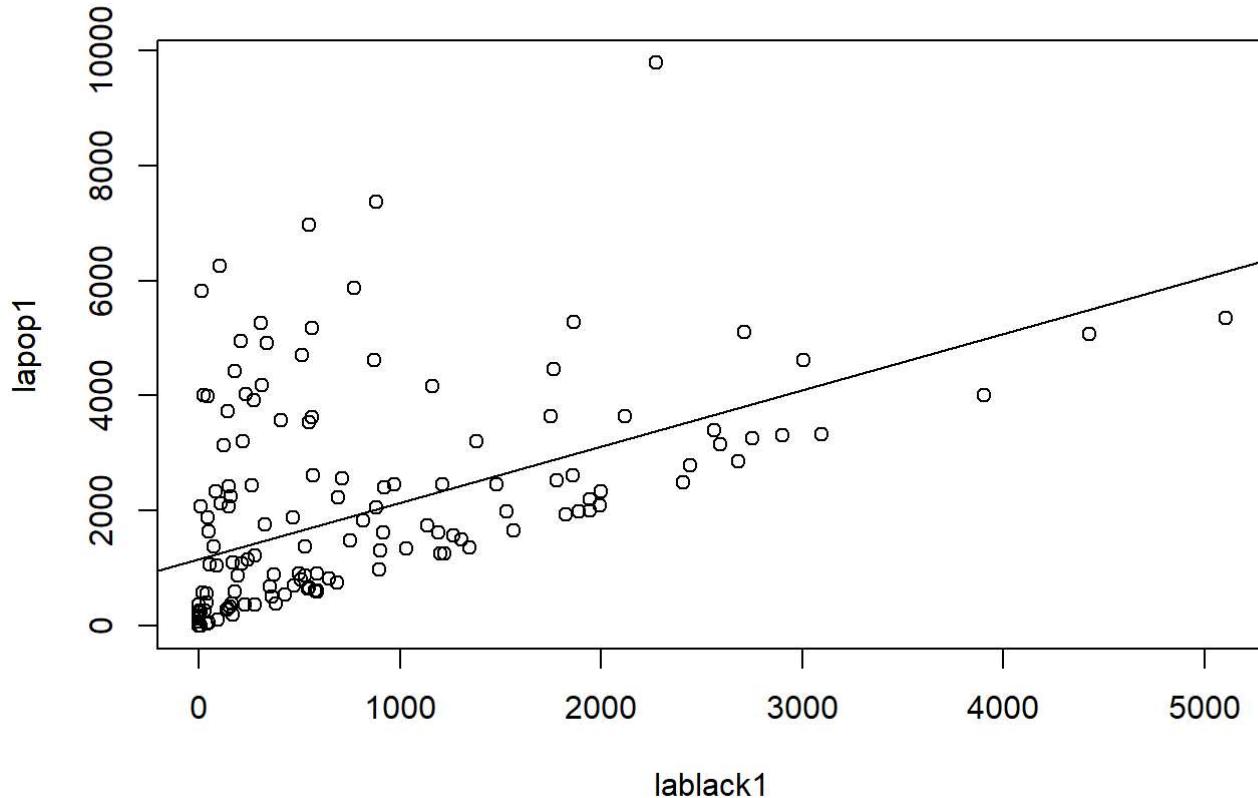
Regression for Hispanic population in Food Deserts compared to total population in Food Deserts



```
plot(AL$lablack1, AL$lapop1, main = "Regression for AA population in Food Deserts compared to total population in Food Deserts",
     xlab= "lablack1", ylab= "lapop1") +
abline(lm(lapop1 ~ lablack1, data = AL, col = "red"))
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...):
##   extra argument 'col' will be disregarded
```

tion for AA population in Food Deserts compared to total population in Fo



```
## integer(0)
```

Linear Regression is a linear model that assumes a linear relationship between input variables(independent variables 'x') and output variable(dependent variable-'y') such that 'y' can be calculated from a linear combination of input variables(x).

```
model <- lm(lapop1 ~ lawhite1, data = AL)
model2 <- lm(lapop1 ~ lablack1, data = AL)
model3 <- lm(lapop1 ~ laasian1, data = AL)
model4 <- lm(lapop1 ~ lahisp1, data = AL)
model5 <- lm(lapop1 ~ lawhite1, data = GA)
model6 <- lm(lapop1 ~ lablack1, data = GA)
model7 <- lm(lapop1 ~ laasian1, data = GA)
model8 <- lm(lapop1 ~ lahisp1, data = GA)
```

Linear regression are often used for predictions. For the essence of time, we will just look at the first equation, and all the other equation follow the same format with each respective race. This equation is predicting the total population greater than 1 mile from a supermarket in Jefferson County, AL with a predictor variable of total African American population, you can also switch the equation inversely to achieve predictions for each individual race based on total population. ##Machine Learning Machine learning is a prediction model used in data science that allows one to predict information based on an algorithm created by the user. In our research we could use machine learning to predict who is more likely to live in urban food deserts based on the variables provided in the data set.