# Analysis of Food Access in the United States

Jessica De Anda

February 2020

## Abstract

# Contents



# List of Figures

# 1 Introduction

OUTLINE:

- Present the issue: access to supermarkets in the US is not uniform across the country. Poor access to supermarkets in this report is defined as living far from a supermarket and lacking vehicle access.
  - Why we should care: poor access to food = lower quality of life
  - Biological impact of food deserts: health; research demonstrates that obesity is prominent in communities living in food deserts. The lack of access to supermarkets with fresh produce perpetuates community dependence on fast food, which is typically unhealthy (high fat, low nutrient).
  - REFERENCES: Gordon-Larsen, 2014 ; National Research Council, 2007 ; White, 2007
- Factors that influence access to food: socioeconomic status
  - Income/economic standing: low income communities are less likely to have access to fresh produce
  - Race: racial profile often influences economic standing. Lower income communities tend to be comprised of racial minorities (largely due to social inequity), therefore there may be a correlation between access to food and racial identity (REFERENCE: Tobin and Weaver, 2017)
- Overview of this report
  - Purpose: to assess trends betwwen socioeconomic status and access to food in the US in order to identify the groups that may need additional services from the government to improve their quality of life
  - Methods: obtained data on 2010 food access in the US from USDA. Calculated the percentage of each racial group (white, black, asian, latino, islander, native american, and other) living at 0.5, 1, 10, and 20 miles from a supermarket.
  - Hypothesis: based on past research, it is likely that all racial minorities will have less access to supermarkets than the white population in the US

# 2 Methods

## 2.1 Importing the data file

Public data on food access in the United States in 2010 was downloaded from the United States Department of Agriculture (USDA) as a csv file. A function was defined within a python script to open the csv file, import the data using the pandas library, and convert the data to a numpy file for future numerical analysis.

```
import pandas as pd
import numpy as np

def np_from_csv(csv_file):
    temp_data = pd.read_csv(csv_file, header = None)
    data = temp_data.to_numpy()
    return data

data = np_from_csv('food_access_data.csv')
```

## 2.2 Analysis of access to supermarkets per racial category

A function was defined to calculate the percentage of a race living at a particular distance from the supermarket. Given an input of race and distance, the defined function first used the numpy.sum operation to calculate the total number of individuals of the race living at the specified distance from the supermarket. This value was then divided by total

number of individuals of the race, which was also generated using the numpy.sum operation. The function returned the calculation as a percentage by multiplying the value by 100.

```
def percent_race_at_distance(input_data, race_distance, race_total, dec = 2):
    # inside the parentheses, the format is: dataset_imported[rows, columns]
    race_distance_sum = np.sum(input_data[:, race_distance])
    race_total_sum = np.sum(input_data[:, race_total])
    percentage = race_distance_sum/race_total_sum * 100
    return round(percentage, dec)
```

A second function took an input of race name and the previously calculated percentage of that race living at a given distance from the supermarket to plot the percentages against the corresponding distances. The plot function of the matplotlib.pyploy library was used to graph this relationship and additional functionalities of this library were used to format the plot.

```
import matplotlib.pyplot as plt

def plot_race_percentage_vs_distance(race, perc):
    [half,one,ten,twenty] = perc
    plt.plot([0.5, 1, 10, 20],[half, one, ten, twenty])
    # the following code formats the plot
    plt.axis([0, 20, 0, 100])
    plt.xlabel('Distance from Supermarket (Miles)')
    plt.ylabel('Percentage')
    plt.title('Percentage of %s Population Living at a Distance from a Supermarket' % race)
    return
```

All of the data in the file was organized into arrays. The data would later be extracted sequentially and inputted into a for loop that combined the two defined functions.

```
total = [white_total, black_total, asian_total, islander_total, native_total, other_total,
    latino_total]
# total is an array of the data columns corresponding to the total population of each race

distance = [[white_half, white_1, white_10, white_20],
            [black_half, black_1, black_10, black_20],
            [asian_half, asian_1, asian_10, asian_20],
            [islander_half, islander_1, islander_10, islander_20],
            [native_half, native_1, native_10, native_20],
            [other_half, other_1, other_10, other_20],
            [latino_half, latino_1, latino_10, latino_20]]
# distance is a 7X4 array given that it holds data for 7 racial categories and 4 distances
# each row corresponds to a race and each column corresponds to a distance

name = ['White', 'Black', 'Asian', 'Islander', 'Native', 'Other', 'Latino']
# name is a list of the racial categories
```

The for loop containing both functions was designed to automatically input and analyze all of the data without having to manually enter each race and distance.

```
perc = np.zeros(7,4)
# create an empty 7X4 array corresponding to the number of races by the number of distances
# the percentages of each race living at a given distance will be organized in this array

for race in range(7):
# the following for loop will run 7 times (1X per race)
    for dist in range(4):
    # nested within the for loop is a second for loop that will run 4 times (1X per distance)
    # this set of code will run a total of 28 times (1X per race and distance)
        perc[race][dist] = percent_race_at_distance(data, distance[race][dist], total[race], 2)
        # each output of the function "percent_race_at_distance" will be added to array "perc"
```

```
    plot_race_percentage_vs_distance(name[race], perc[race])
    # the percentages in "perc" will be inputted in "plot_race_percentage_vs_distance"
    # it will also pull each race from the list "name"
plt.legend(name)
plt.show()
```

## 2.3 Analysis of food access relative to low-income status

The dataset was imported in RStudio and the libraries dplyr and ggplot2 were installed for data analysis and visualization, respectively. In order to evaluate the relationship between income status and distance from the supermarket, the dataset was filtered to include only the census tracts flagged for low income status.

```
library(dplyr)
library(ggplot2)

# filter dataset to include low income census tracts only
low_income <- filter(food_access_data, low_income_flag == 1)
# select columns of the filtered dataset for further analysis
LowInc <- select(low_income, low_income_flag, total_half, total_1, total_10, total_20)
```

The sum() function was used to calculate the total low-income population at each distance based on the values reported in the filtered dataset. This process was repeated for the original dataset to get total population counts. The total low-income population count was divided by that of the total population at each distance and multiplied by 100 in order to calculate the percentage of the population classified as low-income per discrete distance.

```
# calculate the total number of individuals living at each distance
total_half <- sum(food_access_data$total_half)
total_1 <- sum(food_access_data$total_1)
total_10 <- sum(food_access_data$total_10)
total_20 <- sum(food_access_data$total_20)

# calculate total low-income population at each distance
low_half <- sum(LowInc$total_half)
low_1 <- sum(LowInc$total_1)
low_10 <- sum(LowInc$total_10)
low_20 <- sum(LowInc$total_20)

# calculate the percentage of the total population that was classified as low-income
perc_low_half <- low_half/total_half * 100
perc_low_1 <- low_1/total_1 * 100
perc_low_10 <- low_10/total_10 * 100
perc_low_20 <- low_20/total_20 * 100
```

The percentage of the population classified as low-income was plotted against each distance using the ggplot function.

```
# organize calculated percentages relative to distance in a dataframe
df <- data.frame(distance = c("0.5", "1", "10", "20"), perc_low_income = c(perc_low_half, perc_
    low_1, perc_low_10, perc_low_20))

# plot the dataframe
# specify that the x-axis will correspond to the distances
# the y-axis will represent the percentages
# and the distance bars will be colored
ggplot(df, aes(distance, perc_low_income, fill = distance)) +
        # stat = "identity" makes the x-axis discrete and allows geom_bar to plot the bar graph
        geom_bar(stat = "identity") +
        scale_fill_brewer(palette = "Blues") +
        ggtitle("Percent Low Income VS Distance from the Supermarket") +
        xlab("Distance (miles)") +
        ylab("Low Income Population (%)")
```

## 2.4 Analysis of vehicle access at each distance

The analysis of food access among low-income groups served as a model for analysis of vehicle access relative to distance. First, the dataset was filtered to include only the census tracts flagged for low vehicle access.

```
# filter dataset to include low vehicle access census tracts only
low_vehicle_access <- filter(food_access_data, low_vehicle_access_flag == 1)
# select columns of the filtered dataset for further analysis
LowVehicle <- select(low_vehicle_access, low_income_flag, total_half, total_1, total_10, total_
    20)
```

The sum() function was used to calculate the total low-income population at each distance based on the values reported in the filtered dataset. This process was repeated for the original dataset to get total population counts. The total low-income population count was divided by that of the total population at each distance and multiplied by 100 in order to calculate the percentage of the population classified as low-income per discrete distance.

```
# the total number of individuals living at each distance was previously calculated (Section 2.2)

# calculate total population with low vehicle access at each distance
vehicle_half <- sum(LowVehicle$total_half)
vehicle_1 <- sum(LowVehicle$total_1)
vehicle_10 <- sum(LowVehicle$total_10)
vehicle_20 <- sum(LowVehicle$total_20)

# calculate the percentage of the total population that was flagged for low access to vehicles
perc_vehicle_half <- vehicle_half/total_half * 100
perc_vehicle_1 <- vehicle_1/total_1 * 100
perc_vehicle_10 <- vehicle_10/total_10 * 100
perc_vehicle_20 <- vehicle_20/total_20 * 100
```

The percentage of the population with low access to vehicles was plotted against each distance using the ggplot function.

```
# organize calculated percentages relative to distance in a dataframe
df2 <- data.frame(distance = c("0.5", "1", "10", "20"), perc_vehicle = c(perc_vehicle_half, perc_
    vehicle_1, perc_vehicle_10, perc_vehicle_20))

# plot the dataframe
ggplot(df, aes(distance, perc_vehicle, fill = distance)) +
        # stat = "identity" makes the x-axis discrete and allows geom_bar to plot the bar graph
        geom_bar(stat = "identity") +
        scale_fill_brewer(palette = "Reds") +
        ggtitle("Percent Low Vehicle Access VS Distance from the Supermarket") +
        xlab("Distance (miles)") +
        ylab("Population with Low Vehicle Access (%)")
```

## 2.5 Use of Regex

A dictionary reflecting the number of entries corresponding to each county was created In order to gain a sense of where the data is coming from. The dataset was imported as a txt file. The ”county” column was extracted and appended to an empty list. The re.findall value was then used to find all strings in the new list regardless of potential formatting differences and the defaultdict subclass was importd in order to count the amount of occurrences of each county. These values were stored with their corresponding county in a new dictionary.

```
import csv
import re

counties = []
# opens an empty list
```

```
with open ("complete_data.txt", "r") as dset:
# opens the text file and calls it dset
    data2 = csv.DictReader(dset, delimiter = "\t")
    # \t indicates that the delimiter is a tab
    for row in data2:
        counties.append(row['county'])
        # extracts the county column and appends it to the empty list

# use regular expressions to match all strings in the list
# the strings correspond to county names
county_regex = re.compile(r'\w+')
re.findall(auth_regex, str(test_strings))

# count the amount of occurrences of each county and organize them in a dictionary
from collections import defaultdict
county_dict = defaultdict(int)
for county in counties:
    county_dict[county] += 1
```

# 3 Results

- Access based on race
  - Reference figure 1
  - total population for every race decreases the further away you go, but there are differences between groups
- Access based on low-income status
  - Reference figure 2
  - over half of the population living 20 miles from the supermarket is classified as low income
  - for all other distances, low-income population comprises less than half of the total population
- Vehicle access at varying distances
  - Reference figure 3
  - range for all distances: 20-30%
  - Over 25% of the population living 20 miles away from the supermarket lacks access to vehicles

# 4 Discussion

Future directions

- Use GIS to construct a map to better visualize the data. Report randomly selected case studies in order to get a better idea of how living at each of these distances from the supermarket influences different people
- also look at price –¿ doesn't matter that a supermarket is close by if the food is too expensive to buy!
- Research ways to address food insecurity

# 5 References

*** This section will be edited

# 6 Figures

Figure 1: Food access across racial categories

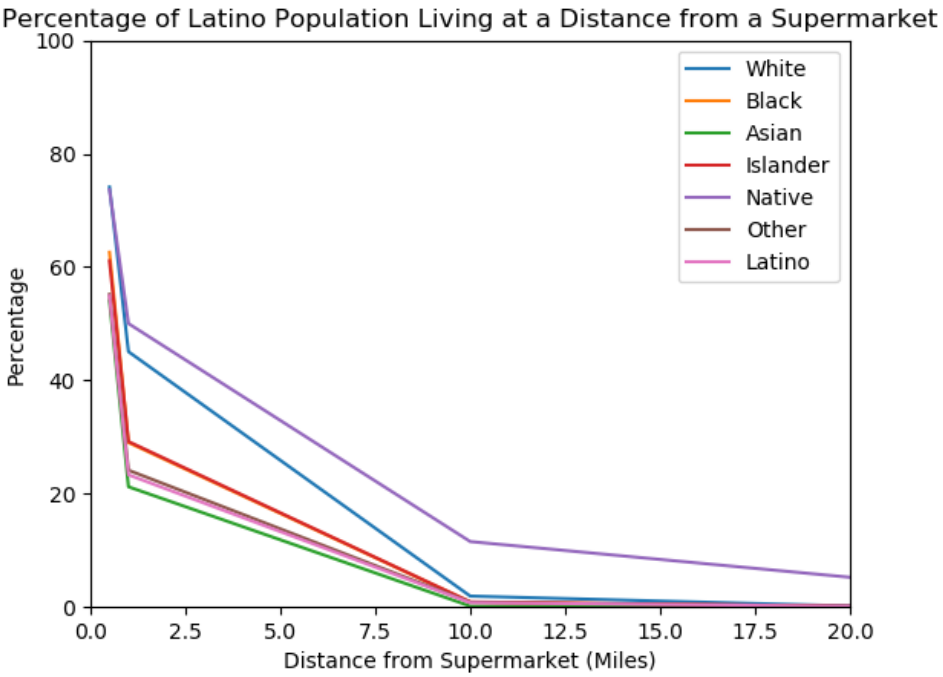

Figure 2: Food access among low-income groups

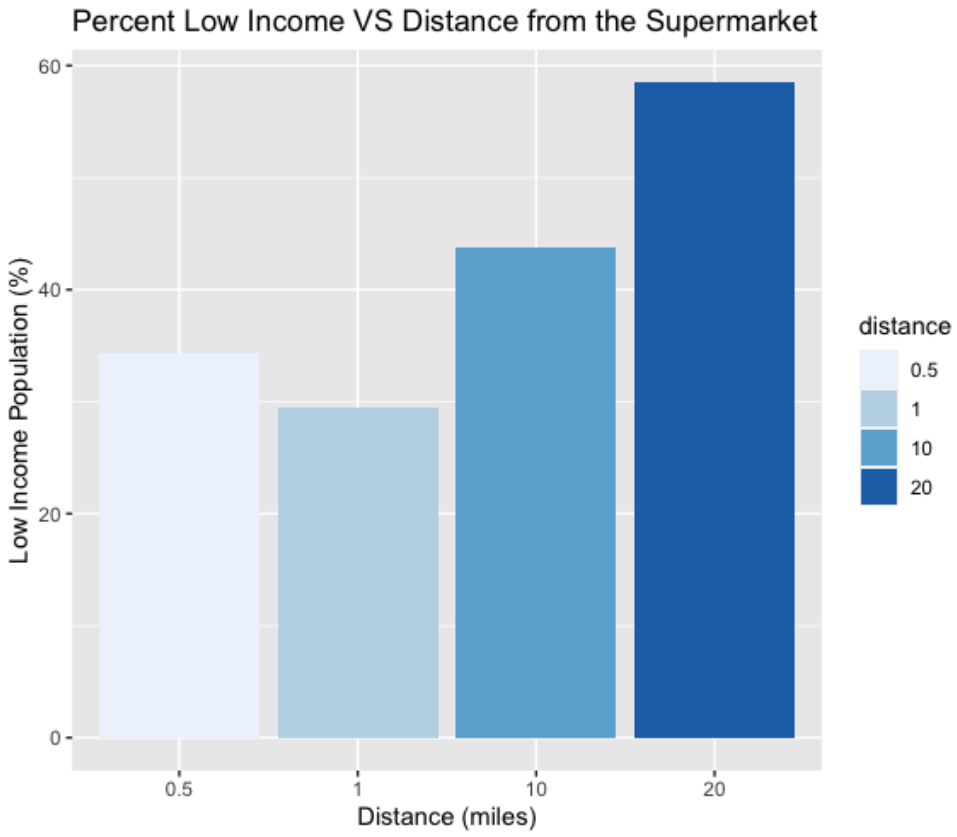

Figure 3: Vehicle access at varying distances from the supermarket