Analysis & Reporting of Obesity in Georgia, U.S and Neighboring States

Oluwasayo Farotimi

2024-01-20

Content

Section 1: Overview

This report focuses on the vital health problem of Obesity in the United States which became prevalent in western countries in the 1980s. By definition, Obesity is a case where the Body Mass Index is equal or above 30 due to abnormal or excessive fat accumulation. The progression of Obesity among adults of age 20 - 74 has risen over the years from 15% (1976 - 1980) to 23.3% (1988 - 1994) to 30.9% (in 1999 - 2000).

Here are some causes of this obesity pandemic:

- Dietary Fat
- Sugar and Sugar-Sweetened Beverages
- Hike of farm bills and food prices

The rate of increase in obesity population has continued in an upward movement from 2000 - 2016. My project follows an examination into this from the year 2015 - 2021.

An investigation into the obesity population in Georgia has been conducted which has birthed this report regarding the trend in the specified year(s). A cross examination of neighboring states has also been carried out to include Obesity percentages in Alabama, Tennessee, North Carolina, South Carolina and Florida in comparison to Georgia, to understand how these border states are performing.

Section 2: Data (Survey)

The Data used in this analysis is a survey from Behavioral Risk Factor Surveillance System (BRFSS). It was obtained from the Center for Disease Control and Prevention (CDC) which can be found via: https://data.cdc.gov/Healthy-Aging/Alzheimer-s-Disease-and-Healthy-Aging-Data/hfr9-rurv (https://data.cdc.gov/Healthy-Aging/Alzheimer-s-Disease-and-Healthy-Aging-Data/hfr9-rurv)

The title of the data is “Alzheimer’s disease and healthy aging in the United States”, it is a survey of 39 health concerns ranging from Alzheimer’s to Depression to the primary case of concern in this project which is Obesity. The Author is keen on interpreting the survey, cleaning and transforming data, Visualizing characteristics, creating an interactive web app (Rshiny) and providing a report of findings.

```
#Needed Libraries
library(dplyr)
library(tidyverse)
library(knitr)
library(plotly)
library(shiny)
library(reshape2)
library(MASS)
library(shiny)
library(plotly)
```

Reading the data

```
df =read.csv("/Users/mac/Downloads/Alzheimer_s_Disease_and_Healthy_Aging_Data.csv")

#viewing the top 5 rows of the dataframe
#output for the head of data will not be shown in pdf version of the project.
head(df)
```

```
#checking the structure of the data
str(df)
```

```
## 'data.frame':    250937 obs. of  39 variables:
##  $ RowId                      : chr  "BRFSS~2021~2021~9004~Q43~TOC11~AGE~RACE" "BRF
SS~2017~2017~9001~Q43~TOC11~AGE~OVERALL" "BRFSS~2019~2019~9002~Q02~TNC02~AGE~OVERALL"
"BRFSS~2020~2020~59~Q43~TOC11~AGE~GENDER" ...
##  $ YearStart                  : int  2021 2017 2019 2020 2020 2015 2020 2021 2021 2
020 ...
##  $ YearEnd                    : int  2021 2017 2019 2020 2020 2015 2020 2021 2021 2
020 ...
##  $ LocationAbbr               : chr  "WEST" "NRE" "MDW" "US" ...
##  $ LocationDesc               : chr  "West" "Northeast" "Midwest" "United States, D
C & Territories" ...
##  $ Datasource                 : chr  "BRFSS" "BRFSS" "BRFSS" "BRFSS" ...
##  $ Class                     : chr  "Overall Health" "Overall Health" "Nutrition/P
hysical Activity/Obesity" "Overall Health" ...
##  $ Topic                     : chr  "Arthritis among older adults" "Arthritis amon
g older adults" "Eating 3 or more vegetables daily" "Arthritis among older adults"
...
##  $ Question                  : chr  "Percentage of older adults ever told they hav
e arthritis" "Percentage of older adults ever told they have arthritis" "Percentage o
f older adults who are eating 3 or more vegetables daily" "Percentage of older adults
ever told they have arthritis" ...
##  $ Response                  : logi  NA NA NA NA NA NA ...
##  $ Data_Value_Unit           : chr  "%" "%" "%" "%" ...
##  $ DataValueTypeID           : chr  "PRCTG" "PRCTG" "PRCTG" "PRCTG" ...
##  $ Data_Value_Type           : chr  "Percentage" "Percentage" "Percentage" "Percen
tage" ...
##  $ Data_Value                : num  31.6 50.3 14.3 55.5 15.2 59.8 6.2 61 3.6 69.1
...
##  $ Data_Value_Alt            : num  31.6 50.3 14.3 55.5 15.2 59.8 6.2 61 3.6 69.1
...
##  $ Data_Value_Footnote_Symbol: chr  "" "" "" "" ...
##  $ Data_Value_Footnote       : chr  "" "" "" "" ...
##  $ Low_Confidence_Limit      : chr  "28.8" "49.1" "13.8" "54.5" ...
##  $ High_Confidence_Limit     : chr  "34.4" "51.6" "14.8" "56.4" ...
##  $ Sample_Size               : logi  NA NA NA NA NA NA ...
##  $ StratificationCategory1   : chr  "Age Group" "Age Group" "Age Group" "Age Grou
p" ...
##  $ Stratification1           : chr  "Overall" "65 years or older" "Overall" "65 ye
ars or older" ...
##  $ StratificationCategory2   : chr  "Race/Ethnicity" "" "" "Gender" ...
##  $ Stratification2           : chr  "Hispanic" "" "" "Female" ...
##  $ StratificationCategory3   : logi  NA NA NA NA NA NA ...
##  $ Stratification3           : logi  NA NA NA NA NA NA ...
##  $ Geolocation               : chr  "" "" "" "" ...
##  $ ClassID                   : chr  "C01" "C01" "C02" "C01" ...
##  $ TopicID                   : chr  "TOC11" "TOC11" "TNC02" "TOC11" ...
##  $ QuestionID                : chr  "Q43" "Q43" "Q02" "Q43" ...
##  $ ResponseID                : logi  NA NA NA NA NA NA ...
##  $ LocationID                : int  9004 9001 9002 59 33 9002 59 9001 17 50 ...
##  $ StratificationCategoryID1 : chr  "AGE" "AGE" "AGE" "AGE" ...
##  $ StratificationID1         : chr  "AGE_OVERALL" "65PLUS" "AGE_OVERALL" "65PLUS"
...
##  $ StratificationCategoryID2 : chr  "RACE" "OVERALL" "OVERALL" "GENDER" ...
##  $ StratificationID2         : chr  "HIS" "OVERALL" "OVERALL" "FEMALE" ...
##  $ StratificationCategoryID3 : logi  NA NA NA NA NA NA ...
##  $ StratificationID3         : logi  NA NA NA NA NA NA ...
##  $ Report                    : logi  NA NA NA NA NA NA ...
```

Information about Variables

Readers should note that in this case, an observation/ row of data is not a response given by one (1) person, rather, it is the aggregated values of responses of people of a particular gender, race or within an age group in a geographical location. Here are the descriptions for the ambiguous columns in the dataset:

- "BRFSS" - a unique identifier for the survey observation
- "Class" - health classification for the "survey observation"
- "Topic" - health focus for "survey observation"
- "YearStart" - start Year of the "survey observation"
- "YearEnd" - end Year of the "survey observation"
- "Question" - question asked in the "survey observation"
- "Data_Value_Type" - metric in which the data we have for the "survey observation" was recorded
- "Data_Value" - data Value
- "Data_Value_Alt" - alternative value which should serve as a replacement for Data Value
- "StratificationCategory1" - first layer of Category which was applied during the survey such as Age
- "Stratification1" - component of the first layer (for instance: in Age, you can find "50 - 64 years" or "65 years or older")
- "StratificationCategory2" - second layer of Category which was applied during the survey such as Race or Gender
- "Stratification2" - component of the second layer (for instance: in Gender, you can find "Male" or "Female")
- "Geolocation" - Geographical location in which the survey was carried out.

Section 3 - Tidying and Cleaning

- 3a. The survey file has 250937 observations of 39 variables
- 3b. Examine empty columns, null Values and unique values
- 3c. Are there identical columns (columns that have the same entries)?
- 3d. Are all datatypes appropriate?

3a. The survey file has 250937 observations of 39 variables

This is a very large dataframe and survey, we have quarter of a million rows. The dataframe is not in the typical format, so much data about various health concerns are recorded simultaneously.

3b. Examine empty columns, Null Values and Unique Values

From viewing the structure of the database in the previous section, It should be observed that the last 3 dataframes appear to be empty, some other columns also seem empty. It is important to check if there are other empty columns as well.

For this,we will need a function:

```
empty_columns <- function(df) {
  empty_cols <- colSums(is.na(df)) == nrow(df)
  return(which(empty_cols))
}

empty_columns(df)
```

```
##                  Response               Sample_Size   StratificationCategory3
##                        10                        20                        25
##           Stratification3                ResponseID StratificationCategoryID3
##                        26                        31                        37
##         StratificationID3                    Report
##                        38                        39
```

```
#The values beneath these columns refer to the position of the column in the databas
e.
```

There are 8 empty columns which will be dropped, these columns are totally empty and we have no use for them. Missing values will also be checked.

```r
#Dropping Identified columns
df <- df |> dplyr::select(-Response, -Sample_Size, -StratificationCategory3,
                   - Stratification3, -ResponseID, -StratificationCategoryID3,
                   -StratificationID3, - Report)

# we check for missing values
colSums(is.na(df))
```

```
##                      RowId                  YearStart
##                          0                          0
##                    YearEnd               LocationAbbr
##                          0                          0
##               LocationDesc                 Datasource
##                          0                          0
##                      Class                      Topic
##                          0                          0
##                   Question            Data_Value_Unit
##                          0                          0
##            DataValueTypeID            Data_Value_Type
##                          0                          0
##                 Data_Value             Data_Value_Alt
##                      81635                      81635
## Data_Value_Footnote_Symbol        Data_Value_Footnote
##                          0                          0
##       Low_Confidence_Limit      High_Confidence_Limit
##                          0                          0
##    StratificationCategory1            Stratification1
##                          0                          0
##    StratificationCategory2            Stratification2
##                          0                          0
##                Geolocation                    ClassID
##                          0                          0
##                    TopicID                 QuestionID
##                          0                          0
##                 LocationID  StratificationCategoryID1
##                          0                          0
##          StratificationID1  StratificationCategoryID2
##                          0                          0
##          StratificationID2
##                          0
```

Although, R shows us that there are no missing values, we have to investigate some columns, particularly the “Geolocation”, “Datavalue_Footnote” and “Data_Value_Footnote_symbol”. In section 2, we can see them as empty strings, if this is the case throughout the database, then they should also be dropped at this point of our data cleaning process.

```r
#creating a function in R to identify the number of unique values in a dataframe
nunique <- function(x) {
  length(unique(x))
}

paste( "The Number of unique values in the Geolocation Column is:",
       nunique(df$Geolocation))
```

```
## [1] "The Number of unique values in the Geolocation Column is: 55"
```

```r
paste( "The Number of unique values in the Data Value Footnote Column is:",
       nunique(df$Data_Value_Footnote))
```

```
## [1] "The Number of unique values in the Data Value Footnote Column is: 6"
```

```r
paste( "The Number of unique values in the Data Value Footnote Symbol Column is:",
       nunique(df$Data_Value_Footnote_Symbol))
```

```
## [1] "The Number of unique values in the Data Value Footnote Symbol Column is: 6"
```

We have confirmed that we have other values in the dataframe, hence, the columns are deemed useful.

It is important to examine the number of unique values in the dataframe and the content of all variables so we can further analyze the survey. For that, a function will be created:

```
#The output of this is very lengthy, hence, it will not be shown.

checkUniqueValues <- function(data, columns) {
  unique_values_list <- lapply(data[columns], unique)
  unique_values <- data.frame(column_name = names(unique_values_list), unique_values_
list)
  return(unique_values)
}

columns <- c("YearStart", "YearEnd", "LocationAbbr", "LocationDesc", "Datasource" ,
             "Class", "Topic", "Question", "Data_Value_Unit", "DataValueTypeID",
             "Data_Value_Type" , "Data_Value_Footnote_Symbol", "Data_Value_Footnote",
             "StratificationCategory1", "Stratification1", "StratificationCategory2",
             "Stratification2", "Geolocation", "ClassID", "TopicID", "QuestionID",
             "LocationID", "StratificationCategoryID1", "StratificationID1",
             "StratificationCategoryID2", "StratificationID2")


unique_values_list <- lapply(columns, function(col_name) {
  col <- df[[col_name]]
  length_unique <- length(unique(col))
  cat("Number of unique values in", col_name, ":", length_unique, "\n")
  unique_values <- unique(col)
  cat("Unique values:", "\n")
  print(unique_values)
  cat("\n")
  return(unique_values)
})
```

3c. Identical Columns

Upon reviewing the output above, it is noted that some columns seem identical, These columns are * YearStart and YearEnd * Data_Value and Data_Value_Alt

```
#Checking if the above columns are truly identical
identical(df$YearStart, df$YearEnd)
```

```
## [1] FALSE
```

```
identical(df$Data_Value, df$Data_Value_Alt)
```

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

```
#Dropping Data_Value_Alt since it is identical and provides no new information.
df <- df |> dplyr::select(-Data_Value_Alt)
```

Also, it appears that not all surveys were carried out in one(1) year time frame, if the YearStart and YearEnd entries were identical then we would have concluded that all survey observations were carried in one(1) year period. To understand our survey more, a function will be created to view these rows:

```
#number of observations out of 250937 survey obs. that do not have the same start and
year end.
sum(df$YearStart != df$YearEnd)
```

```
## [1] 11482
```

```
#There are only 11482 observations survey observations that have this characteristic


not_same_year <- df[df$YearStart != df$YearEnd, ]
not_same_year <- not_same_year |> dplyr::select("YearStart", "YearEnd")
kable(head(not_same_year))
```

	YearStart	YearEnd
104	2016	2021
334	2016	2021
344	2016	2021
380	2016	2021
384	2016	2021
408	2016	2021

```
unique(not_same_year$YearStart)
```

```
## [1] 2016
```

```
unique(not_same_year$YearEnd)
```

```
## [1] 2021
```

The period above can be seen to only go from 2016 - 2021, which corresponds to the farthest and most recent year in the master database. We can form a new conclusion that the survey observations in the database either have the same start and end survey year or that the observations starts in 2016 and ends in 2021.

3d. Are all datatypes appropriate?

In this case,every variable has the right datatype,hence, no change will be done here.

Section 4 - Data Transformation

In this section,data operations such as subsetting, filtering and manipulation of dataframes will be done in order to zoom into Georgia and neighboring states.

4a Obesity in Georgia State

```
#subsetting observations that record data about Obesity
df_obesity = df[df$Topic == 'Obesity', ]
#9 columns are irrelevant at this point, this is because they contain the same value
unneeded <- df_obesity |> dplyr::select(Class, Question, Topic, TopicID, QuestionID,
                                 Data_Value_Type, Data_Value_Unit, DataValueTypeID,
                                 StratificationCategory1, Datasource)
head(unneeded)
```

```
#drop unneeded columns for clarity in analysis and decision making
df_obesity <- df_obesity |> dplyr::select(-Class, -Question, - Topic,-TopicID,
                                   - QuestionID, -DataValueTypeID, -Data_Value_Type,
                                   -StratificationCategory1, -Data_Value_Unit,
                                   -Datasource)
```

To zoom in on Georgia, the dataframe will be filtered with 2021 will being the primary focus.

```
df_obesity_ga2021 = df_obesity[(df_obesity$YearStart == 2021) &
                                 (df_obesity$YearEnd == 2021) &
                                 (df_obesity$LocationDesc == "Georgia"),  ]
```

specific columns will be selected for the type of visualizations we want to create in the next section , since the “YearStart” and “YearEnd” values are the same,one of these variables will be selected and renamed to “Year”, some other variables will be renamed as well, the index of the new dataframe will also be reset.

```
df_focus <- df_obesity_ga2021 |> dplyr::select(YearStart, Data_Value, Stratification
1,
                                          StratificationCategoryID2, Stratification2)


#rename column
df_focus <- df_focus |>
  rename(Year = YearStart, Category1 = Stratification1,
         CategoryID2 = StratificationCategoryID2, Category2 = Stratification2,
         Data_Percent = Data_Value)



#reset index
rownames(df_focus) <- 0:(nrow(df_focus) - 1)
df_focus <- data.frame(df_focus)
kable(head(df_focus))
```

	Year	Data_Percent	Category1	CategoryID2	Category2
0	2021	40.8	50-64 years	OVERALL	
1	2021	NA	Overall	RACE	Asian/Pacific Islander
2	2021	28.9	65 years or older	OVERALL	
3	2021	29.2	65 years or older	GENDER	Female
4	2021	31.8	Overall	RACE	White, non-Hispanic
5	2021	46.2	Overall	RACE	Black, non-Hispanic

```
str(df_focus)
```

```
## 'data.frame':    24 obs. of  5 variables:
##  $ Year        : int  2021 2021 2021 2021 2021 2021 2021 2021 2021 2021 ...
##  $ Data_Percent: num  40.8 NA 28.9 29.2 31.8 46.2 37.8 26.2 NA 34.4 ...
##  $ Category1   : chr  "50-64 years" "Overall" "65 years or older" "65 years or old
er" ...
##  $ CategoryID2 : chr  "OVERALL" "RACE" "OVERALL" "GENDER" ...
##  $ Category2   : chr  "" "Asian/Pacific Islander" "" "Female" ...
```

```
# Our First focus is on the race of people with obesity from ages 50-64 Ages
df_obesega1 <- df_focus[(df_focus$Category1 == '50-64 years') &
                          (df_focus$CategoryID2 == 'RACE'),]
kable(df_obesega1)
```

	Year	Data_Percent	Category1	CategoryID2	Category2

10	2021	NA	50-64 years	RACE	Native Am/Alaskan Native
18	2021	NA	50-64 years	RACE	Asian/Pacific Islander
19	2021	51.6	50-64 years	RACE	Black, non-Hispanic
21	2021	35.3	50-64 years	RACE	Hispanic
22	2021	37.4	50-64 years	RACE	White, non-Hispanic

In age group 50-64, There is no data for people who are Native American/ Alaskan Native. This could be due to the the location where the survey was conducted. However, we will prepare this dataframe for visualization by dropping these columns. This operation will be performed on subsequent dataframes.

```
df_obesega1 <- na.omit(df_obesega1)
kable(df_obesega1)
```

	Year	Data_Percent	Category1	CategoryID2	Category2
19	2021	51.6	50-64 years	RACE	Black, non-Hispanic
21	2021	35.3	50-64 years	RACE	Hispanic
22	2021	37.4	50-64 years	RACE	White, non-Hispanic

```
# Focusing on the Gender of people with obesity with ages 50-64 Ages
df_obesega2 <- df_focus[(df_focus$Category1 == '50-64 years') &
                          (df_focus$CategoryID2 == 'GENDER'),]
kable(df_obesega2)
```

	Year	Data_Percent	Category1	CategoryID2	Category2
12	2021	39.1	50-64 years	GENDER	Male
16	2021	42.6	50-64 years	GENDER	Female

```
#Focusing on the Race of people with obesity with ages 65 or older
df_obesega3<- df_focus[(df_focus$Category1 == '65 years or older') &
                         (df_focus$CategoryID2 == 'RACE'),] |>
  na.omit(df_obesega1)
kable(df_obesega3)
```

	Year	Data_Percent	Category1	CategoryID2	Category2
6	2021	37.8	65 years or older	RACE	Black, non-Hispanic
7	2021	26.2	65 years or older	RACE	White, non-Hispanic

```
#Focusing on the Gender of people with obesity with ages 65 or older
df_obesega4<- df_focus[(df_focus$Category1 == '65 years or older') &
                         (df_focus$CategoryID2 == 'GENDER'),]
kable(df_obesega4)
```

	Year	Data_Percent	Category1	CategoryID2	Category2
3	2021	29.2	65 years or older	GENDER	Female
11	2021	28.5	65 years or older	GENDER	Male

```
#Examinining the overall percentage of Obese Persons in Georgia in the Year 2021
kable(df_focus[(df_focus$Category1 == 'Overall') & (df_focus$CategoryID2 == 'OVERAL
L'),])
```

	Year	Data_Percent	Category1	CategoryID2	Category2
23	2021	35.4	Overall	OVERALL	

We will proceed to perform more transformations. Our focus will be widened to capture more data from 2016 - 2021,particularly the overall percentages.

Scientific Inquiries

- What is the Overall yearly percentage of Obese persons in Georgia from 2016 - 2021?
- Can we get a dataframe with corresponding percentages from 2016 - 2021 for neighboring states of Georgia?

Answers will be provided to these inquiries in the next subsections

4b. Obesity Rates in Georgia (%) from 2015 - 2021

A function has been created for this:

```
get_filtered_data <- function(year_start_list, year_end_list, location_list) {
  combined_df <- data.frame()

  for (i in seq_along(year_start_list)) {
    df_new <- df_obesity[
      (df_obesity$YearStart == year_start_list[i]) &
      (df_obesity$YearEnd == year_end_list[i]) &
      (df_obesity$LocationDesc == location_list[i]), ]

    df_focus1 <- df_new |>
      dplyr::select(Data_Value,YearStart,LocationDesc, StratificationCategoryID1,
                    Stratification1, StratificationCategoryID2, Stratification2) |>
      filter(Stratification1 == 'Overall' & StratificationCategoryID2 == 'OVERALL') |
>
       rename(Year = YearStart)

    combined_df <- rbind(combined_df, data.frame(df_focus1))
  }

  return(combined_df)
}

# Example usage with lists
year_start_list <- c(2015, 2016, 2017, 2018, 2019, 2020, 2021)
year_end_list <- c(2015, 2016, 2017, 2018, 2019, 2020, 2021)
location_list <- rep("Georgia", 7)

df_ga_years <- get_filtered_data(year_start_list, year_end_list, location_list)

head(df_ga_years)
```

4c. Obesity Rates(%) in Georgia and Neighboring States from 2015 - 2021

The image below is a cutout map of the neighboring states of Georgia, we will focus on these additional 5 states as well.

```
knitr::include_graphics("/Users/mac/Downloads/Neigboring States of Georgia.png", rel_
path = FALSE)
```

Tennessee
North Carolina
South Carolina
Georgia
Alabama
Florida

The previous function has been modified to get us the overall percentage(s) of obesity from neigboring States, Here a dataframe in form of a list of lists will be created:

```
get_filtered_data <- function(year_start_list, year_end_list, location_list) {
  state_data_list <- list()

  for (state in unique(location_list)) {
    df_new <- df_obesity[
      (df_obesity$YearStart %in% year_start_list) &
      (df_obesity$YearEnd %in% year_end_list) &
      (df_obesity$LocationDesc == state), ]

    df_focus2 <- df_new |>
      dplyr::select(Data_Value,YearStart, LocationDesc, StratificationCategoryID1, St
ratification1, StratificationCategoryID2, Stratification2) |>
      filter(Stratification1 == 'Overall' & StratificationCategoryID2 == 'OVERALL') |
>
      rename(Year = YearStart)

    state_data_list[[paste0("df_combined_", gsub(" ", "", state))]] <- data.frame(df_
focus2)
  }

  return(state_data_list)
}

States <- c("Georgia", "Alabama", "Tennessee", "North Carolina", "South Carolina", "F
lorida")
year_start_list <- c(2015, 2016, 2017, 2018, 2019, 2020, 2021)
year_end_list <- c(2015, 2016, 2017, 2018, 2019, 2020, 2021)
location_list <- rep(States, each = 7)

df_neighborstates <- get_filtered_data(year_start_list, year_end_list, location_list)
print(df_neighborstates)
```

```
## $df_combined_Georgia
##   Data_Value Year LocationDesc StratificationCategoryID1 Stratification1
## 1       34.2 2015      Georgia                       AGE         Overall
## 2       36.3 2019      Georgia                       AGE         Overall
## 3       34.8 2016      Georgia                       AGE         Overall
## 4       34.8 2018      Georgia                       AGE         Overall
## 5       36.2 2020      Georgia                       AGE         Overall
## 6       33.9 2017      Georgia                       AGE         Overall
## 7       35.4 2021      Georgia                       AGE         Overall
##   StratificationCategoryID2 Stratification2
## 1                   OVERALL
## 2                   OVERALL
## 3                   OVERALL
## 4                   OVERALL
## 5                   OVERALL
## 6                   OVERALL
## 7                   OVERALL
```

```
##
## $df_combined_Alabama
##   Data_Value Year LocationDesc StratificationCategoryID1 Stratification1
## 1       38.8 2021      Alabama                       AGE         Overall
## 2       40.0 2020      Alabama                       AGE         Overall
## 3       36.2 2015      Alabama                       AGE         Overall
## 4       36.7 2019      Alabama                       AGE         Overall
## 5       37.3 2018      Alabama                       AGE         Overall
## 6       36.0 2016      Alabama                       AGE         Overall
## 7       36.7 2017      Alabama                       AGE         Overall
##   StratificationCategoryID2 Stratification2
## 1                   OVERALL
## 2                   OVERALL
## 3                   OVERALL
## 4                   OVERALL
## 5                   OVERALL
## 6                   OVERALL
## 7                   OVERALL
##
## $df_combined_Tennessee
##   Data_Value Year LocationDesc StratificationCategoryID1 Stratification1
## 1       35.1 2021    Tennessee                       AGE         Overall
## 2       37.7 2020    Tennessee                       AGE         Overall
## 3       35.3 2018    Tennessee                       AGE         Overall
## 4       37.4 2019    Tennessee                       AGE         Overall
## 5       33.6 2015    Tennessee                       AGE         Overall
## 6       34.1 2017    Tennessee                       AGE         Overall
## 7       34.5 2016    Tennessee                       AGE         Overall
##   StratificationCategoryID2 Stratification2
## 1                   OVERALL
## 2                   OVERALL
## 3                   OVERALL
## 4                   OVERALL
## 5                   OVERALL
## 6                   OVERALL
## 7                   OVERALL
##
## $df_combined_NorthCarolina
##   Data_Value Year   LocationDesc StratificationCategoryID1 Stratification1
## 1       32.0 2015 North Carolina                       AGE         Overall
## 2       32.9 2016 North Carolina                       AGE         Overall
## 3       35.6 2020 North Carolina                       AGE         Overall
## 4       34.2 2017 North Carolina                       AGE         Overall
## 5       38.2 2021 North Carolina                       AGE         Overall
## 6       35.6 2019 North Carolina                       AGE         Overall
## 7       34.4 2018 North Carolina                       AGE         Overall
##   StratificationCategoryID2 Stratification2
## 1                   OVERALL
## 2                   OVERALL
## 3                   OVERALL
## 4                   OVERALL
## 5                   OVERALL
## 6                   OVERALL
## 7                   OVERALL
## 
## $df_combined_SouthCarolina
##   Data_Value Year   LocationDesc StratificationCategoryID1 Stratification1
## 1       37.3 2021 South Carolina                       AGE         Overall
## 2       36.7 2020 South Carolina                       AGE         Overall
## 3       35.9 2019 South Carolina                       AGE         Overall
## 4       35.2 2018 South Carolina                       AGE         Overall
## 5       33.0 2016 South Carolina                       AGE         Overall
## 6       32.8 2015 South Carolina                       AGE         Overall
## 7       34.2 2017 South Carolina                       AGE         Overall
##   StratificationCategoryID2 Stratification2
## 1                   OVERALL                
## 2                   OVERALL                
## 3                   OVERALL                
## 4                   OVERALL                
## 5                   OVERALL                
## 6                   OVERALL                
## 7                   OVERALL                
## 
## $df_combined_Florida
##   Data_Value Year LocationDesc StratificationCategoryID1 Stratification1
## 1       28.8 2016      Florida                       AGE         Overall
## 2       30.0 2020      Florida                       AGE         Overall
## 3       29.3 2019      Florida                       AGE         Overall
## 4       31.5 2018      Florida                       AGE         Overall
## 5       29.0 2015      Florida                       AGE         Overall
## 6       29.3 2017      Florida                       AGE         Overall
##   StratificationCategoryID2 Stratification2
## 1                   OVERALL                
## 2                   OVERALL                
## 3                   OVERALL                
## 4                   OVERALL                
## 5                   OVERALL                
## 6                   OVERALL                
```

```
arrange_by_year <- function(df, column_name) {
  df[[column_name]] <- df[[column_name]] |>
    arrange(Year)
  return(df)
}

columns_to_arrange <- c(
  "df_combined_Georgia",
  "df_combined_Alabama",
  "df_combined_Tennessee",
  "df_combined_NorthCarolina",
  "df_combined_SouthCarolina",
  "df_combined_Florida"
)

for (column in columns_to_arrange) {
  df_neighborstates <- arrange_by_year(df_neighborstates, column)
}

print(df_neighborstates)
```

4d Unrecorded Observation (Florida, 2021)

The State of Florida does not have a value for the year 2021, This can be deemed as a missing value. To solve this problem, a row for 2021 which uses the value for 2019 & 2020 will be created.

```
data_2021 = (df_neighborstates$df_combined_Florida$Data_Value[5] + df_neighborstates$
df_combined_Florida$Data_Value[6])/2

fl_column2021 = c(data_2021, 2021, "Florida", "AGE", "Overall", "OVERALL", "")
df_neighborstates$df_combined_Florida <- rbind(df_neighborstates$df_combined_Florida,
fl_column2021)
```

The df_neighborstates is in in a list of list format which is not suitable for visualizations, hence, a new dataframe will be created for graphical needs.

```
#We will use one of the years in the state dataframes, since the years are all the sa
me
data_nstates <- data.frame(df_neighborstates$df_combined_Georgia$Year,
                   df_neighborstates$df_combined_Georgia$Data_Value,
                   df_neighborstates$df_combined_Alabama$Data_Value,
                   df_neighborstates$df_combined_Tennessee$Data_Value,
                   df_neighborstates$df_combined_NorthCarolina$Data_Value,
                   df_neighborstates$df_combined_SouthCarolina$Data_Value,
                   df_neighborstates$df_combined_Florida$Data_Value)

data_nstates <- data_nstates |>
  rename(Year = df_neighborstates.df_combined_Georgia.Year,
         Georgia = df_neighborstates.df_combined_Georgia.Data_Value,
         Alabama = df_neighborstates.df_combined_Alabama.Data_Value,
         Tennessee = df_neighborstates.df_combined_Tennessee.Data_Value,
         NorthCarolina = df_neighborstates.df_combined_NorthCarolina.Data_Value,
         SouthCarolina = df_neighborstates.df_combined_SouthCarolina.Data_Value,
         Florida = df_neighborstates.df_combined_Florida.Data_Value)

#This is to change the data type of Florida from character to Int
data_nstates$Florida <- as.integer(data_nstates$Florida)

kable(data_nstates)
```

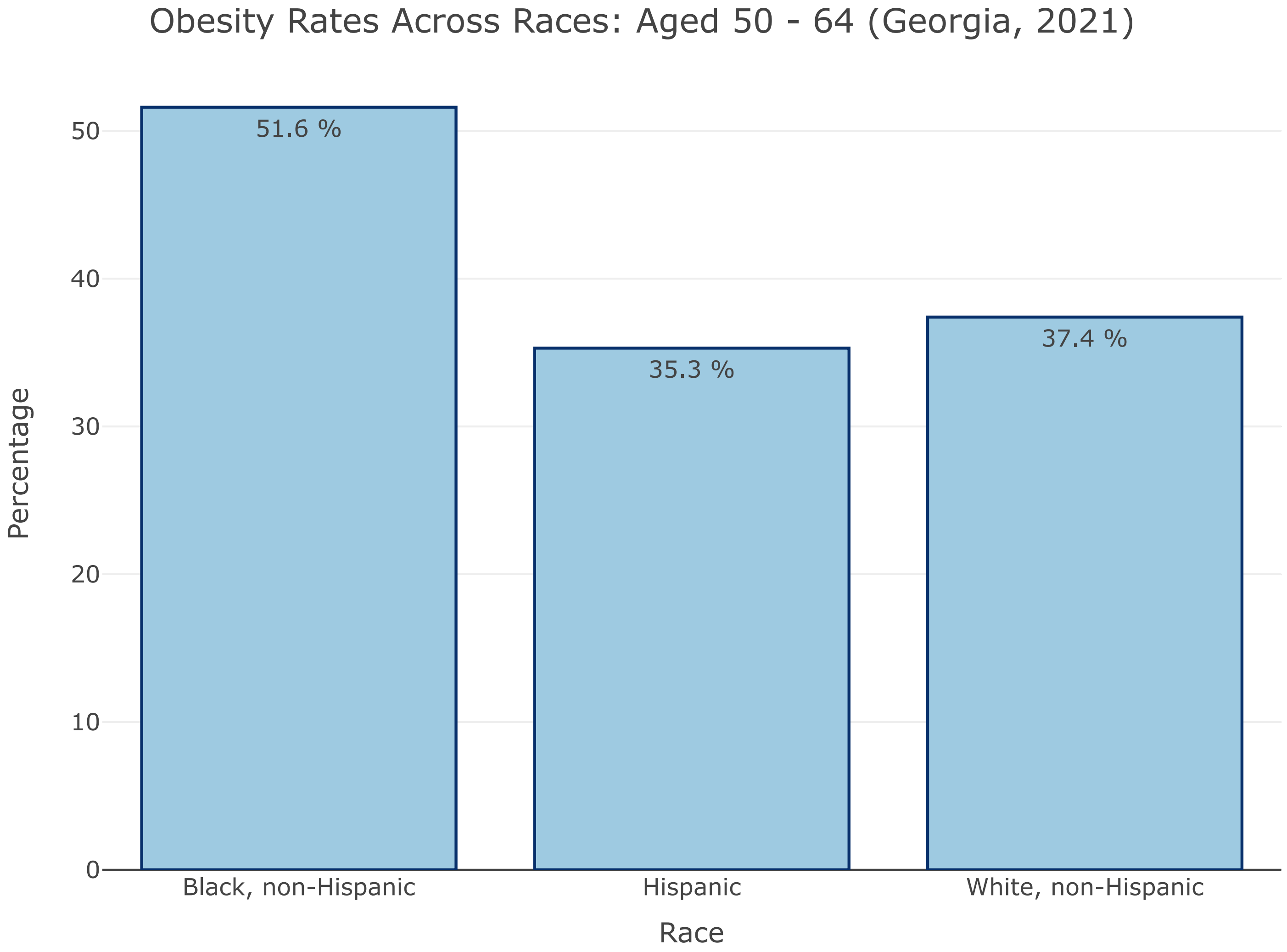
Year	Georgia	Alabama	Tennessee	NorthCarolina	SouthCarolina	Florida
2015	34.2	36.2	33.6	32.0	32.8	29
2016	34.8	36.0	34.5	32.9	33.0	28
2017	33.9	36.7	34.1	34.2	34.2	29
2018	34.8	37.3	35.3	34.4	35.2	31
2019	36.3	36.7	37.4	35.6	35.9	29
2020	36.2	40.0	37.7	35.6	36.7	30
2021	35.4	38.8	35.1	38.2	37.3	29

The visualizations of the transformed data can be seen in Section 5.

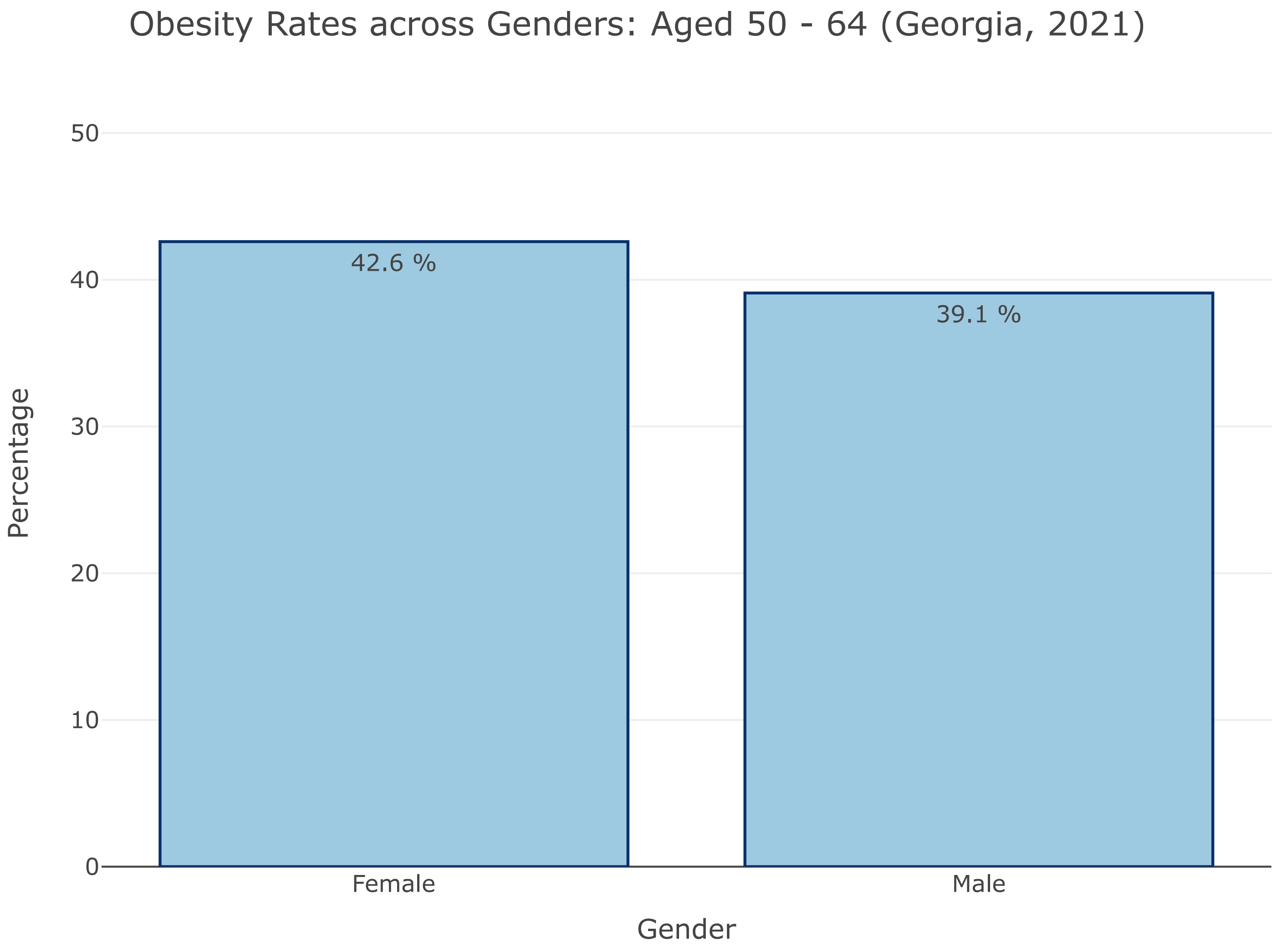
Section 5 - Data Visualization

5a. Barcharts

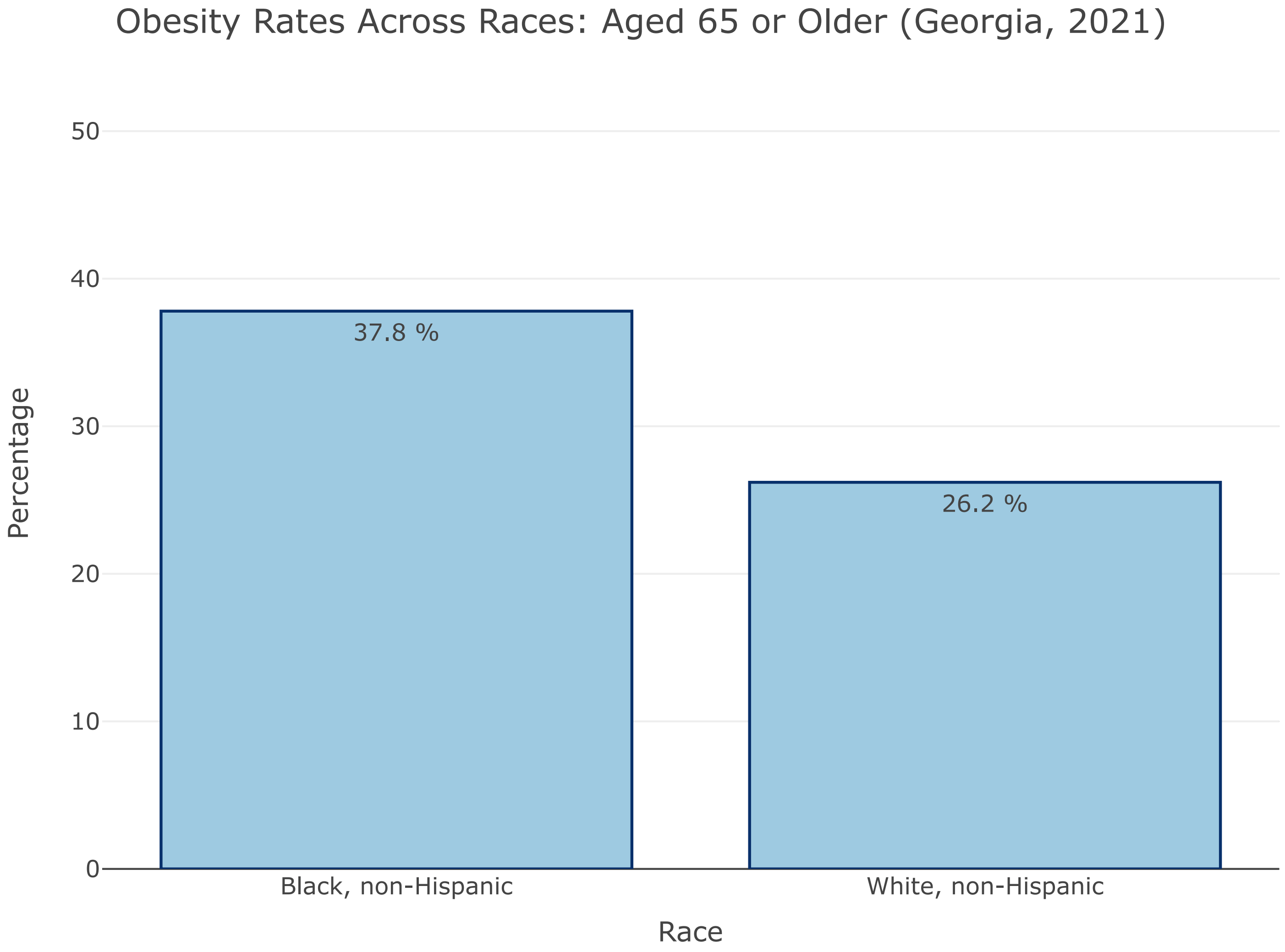
```
# Visualization of race of people with obesity from ages 50-64 Ages
fig <- plot_ly(df_obesega1, x = ~Category2, y = ~Data_Percent, type = 'bar',
             text = paste(df_obesega1$Data_Percent, "%"), textposition = 'auto',
             marker = list(color = 'rgb(158,202,225)',
                           line = list(color = 'rgb(8,48,107)', width = 1.5)))
fig <- fig %>% layout(title = "Obesity Rates Across Races: Aged 50 - 64 (Georgia, 202
1)",
         xaxis = list(title = "Race"),
         yaxis = list(title = "Percentage", range = c(0, 55)))

fig
```



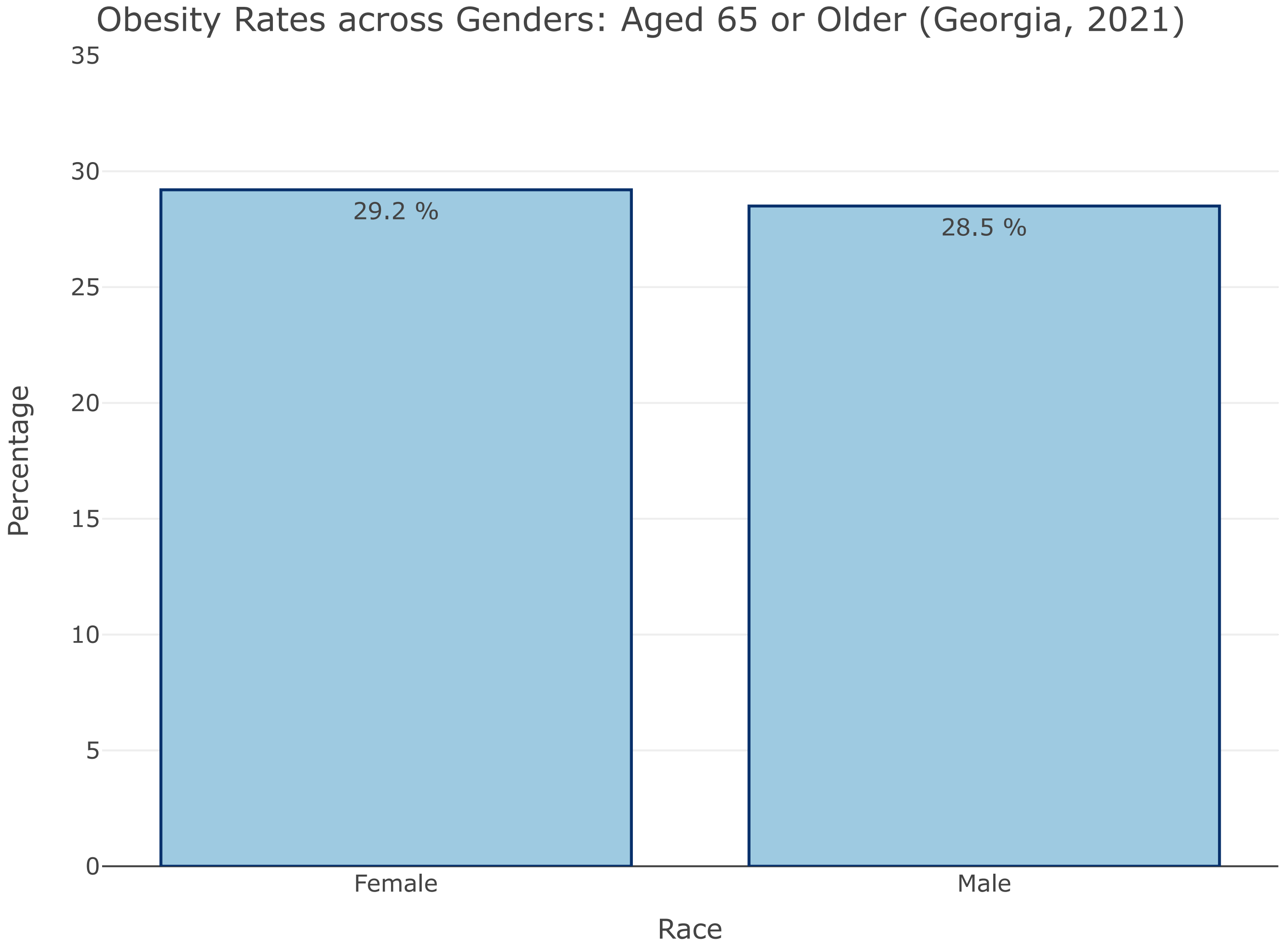
```
# Visualization of gender of people with obesity from ages 50-64 Ages
fig <- plot_ly(df_obesega2, x = ~Category2, y = ~Data_Percent, type = 'bar',
             text = paste(df_obesega2$Data_Percent, "%"), textposition = 'auto',
             marker = list(color = 'rgb(158,202,225)',
                           line = list(color = 'rgb(8,48,107)', width = 1.5)))
fig <- fig %>% layout(title = "Obesity Rates across Genders: Aged 50 - 64 (Georgia, 2
021)",
         xaxis = list(title = "Gender"),
         yaxis = list(title = "Percentage", range = c(0, 55)))

fig
```



```r
# Visualization of race of people with obesity from age 65 or older
fig <- plot_ly(df_obesega3, x = ~Category2, y = ~Data_Percent, type = 'bar',
             text = paste(df_obesega3$Data_Percent, "%"), textposition = 'auto',
             marker = list(color = 'rgb(158,202,225)',
                           line = list(color = 'rgb(8,48,107)', width = 1.5)))
fig <- fig %>% layout(title = "Obesity Rates Across Races: Aged 65 or Older (Georgia,
2021)",
         xaxis = list(title = "Race"),
         yaxis = list(title = "Percentage", range = c(0, 55)))

fig
```

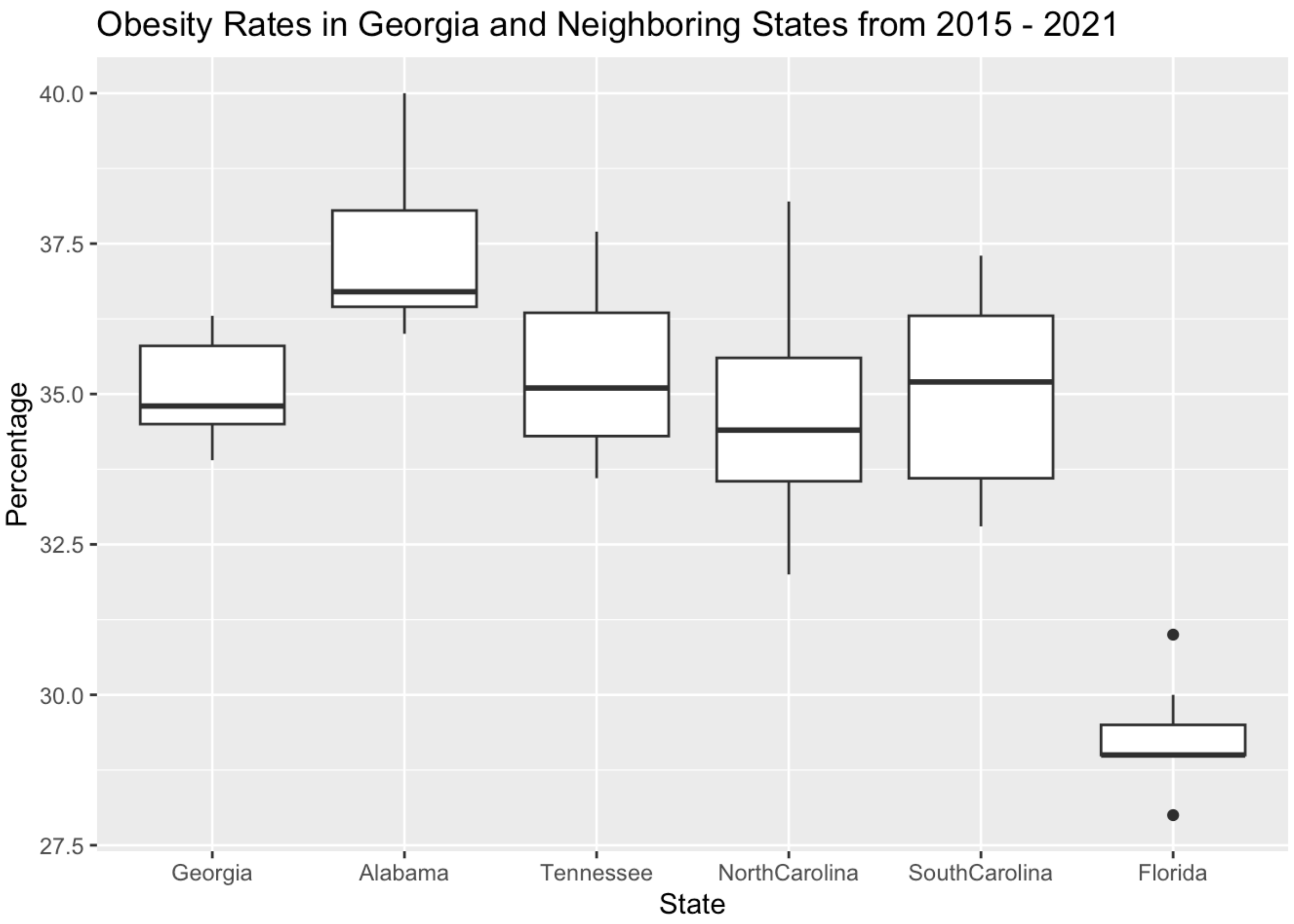


```
# Visualization of gender of people with obesity from age 65 or Older
fig <- plot_ly(df_obesega4, x = ~Category2, y = ~Data_Percent, type = 'bar',
             text = paste(df_obesega4$Data_Percent, "%"), textposition = 'auto',
             marker = list(color = 'rgb(158,202,225)',
                           line = list(color = 'rgb(8,48,107)', width = 1.5)))
fig <- fig %>% layout(title = "Obesity Rates across Genders: Aged 65 or Older (Georgi
a, 2021)",
         xaxis = list(title = "Race"),
         yaxis = list(title = "Percentage", range = c(0, 35)))

fig
```



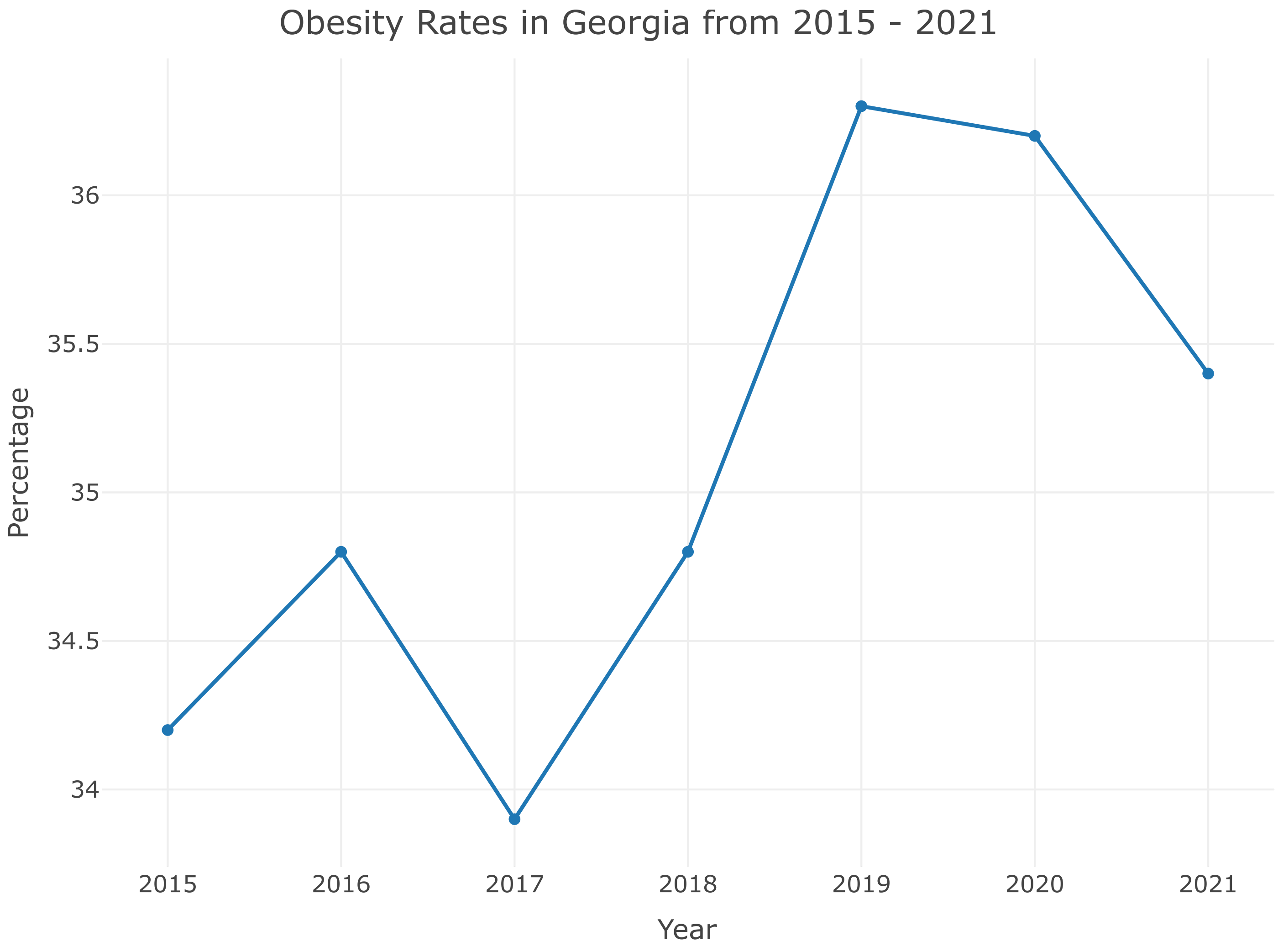
5b. BoxPlot

```
#Transformation for Boxplot: Converting the data to long format
data_long <- data_nstates |>
  dplyr::select(-Year) |>
  melt()

ggplot(data_long, aes(x = variable, y = value)) +
  geom_boxplot() +
  labs(title = "Obesity Rates in Georgia and Neighboring States from 2015 - 2021",
       x = "State",
       y = "Percentage")
```

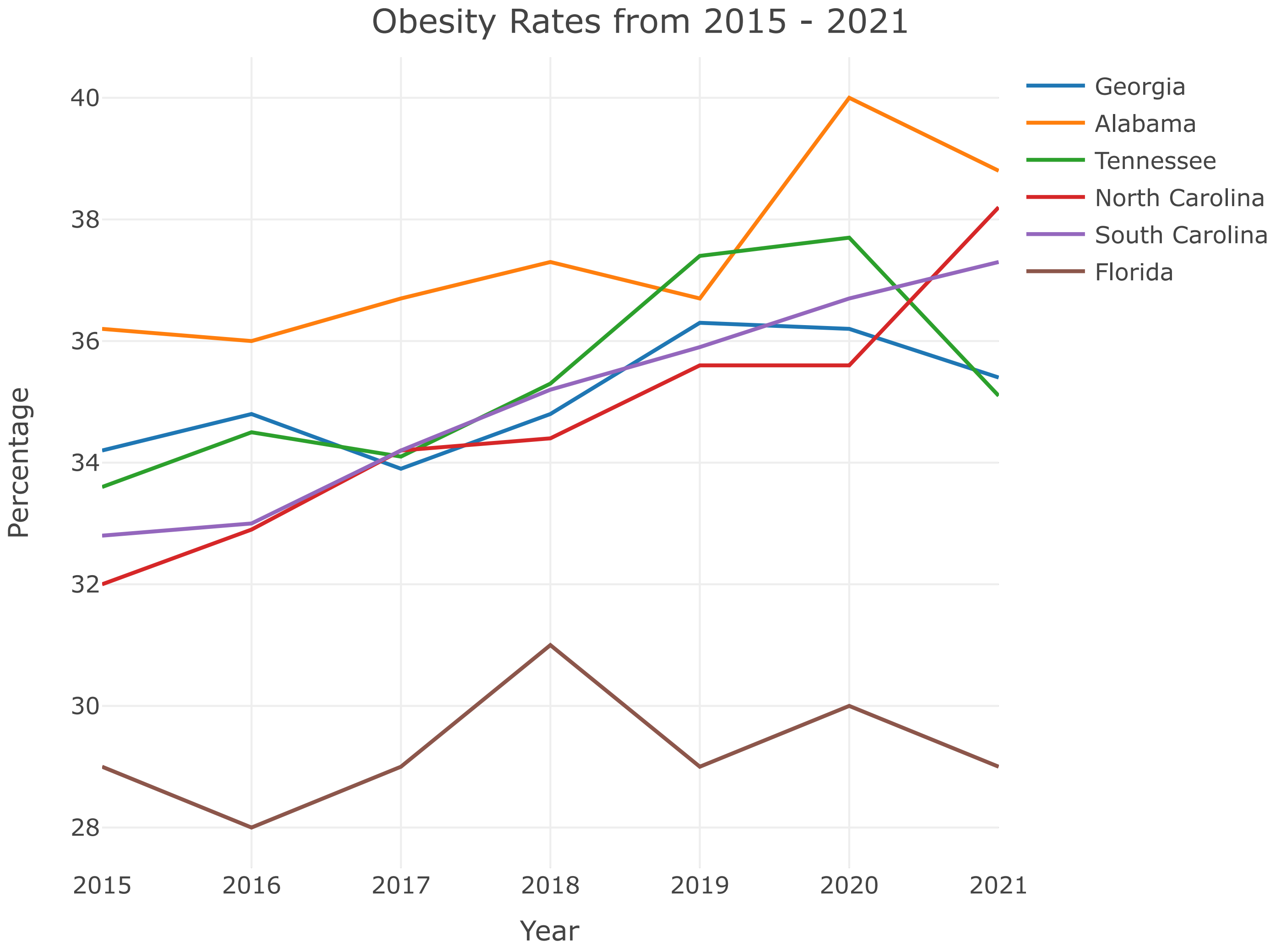


5c. Linegraphs

```
fig <- plot_ly(df_ga_years, x = ~df_ga_years$Year, y = ~df_ga_years$Data_Value,
               name = 'Georgia', type = 'scatter', mode = 'lines+markers',
               text = paste(df_ga_years$Data_Value, "%"), textposition = 'auto')

fig <- fig %>% layout(
    title = "Obesity Rates in Georgia from 2015 - 2021",
    xaxis = list(title = "Year"),
    yaxis = list(title = "Percentage")
     )

fig
```



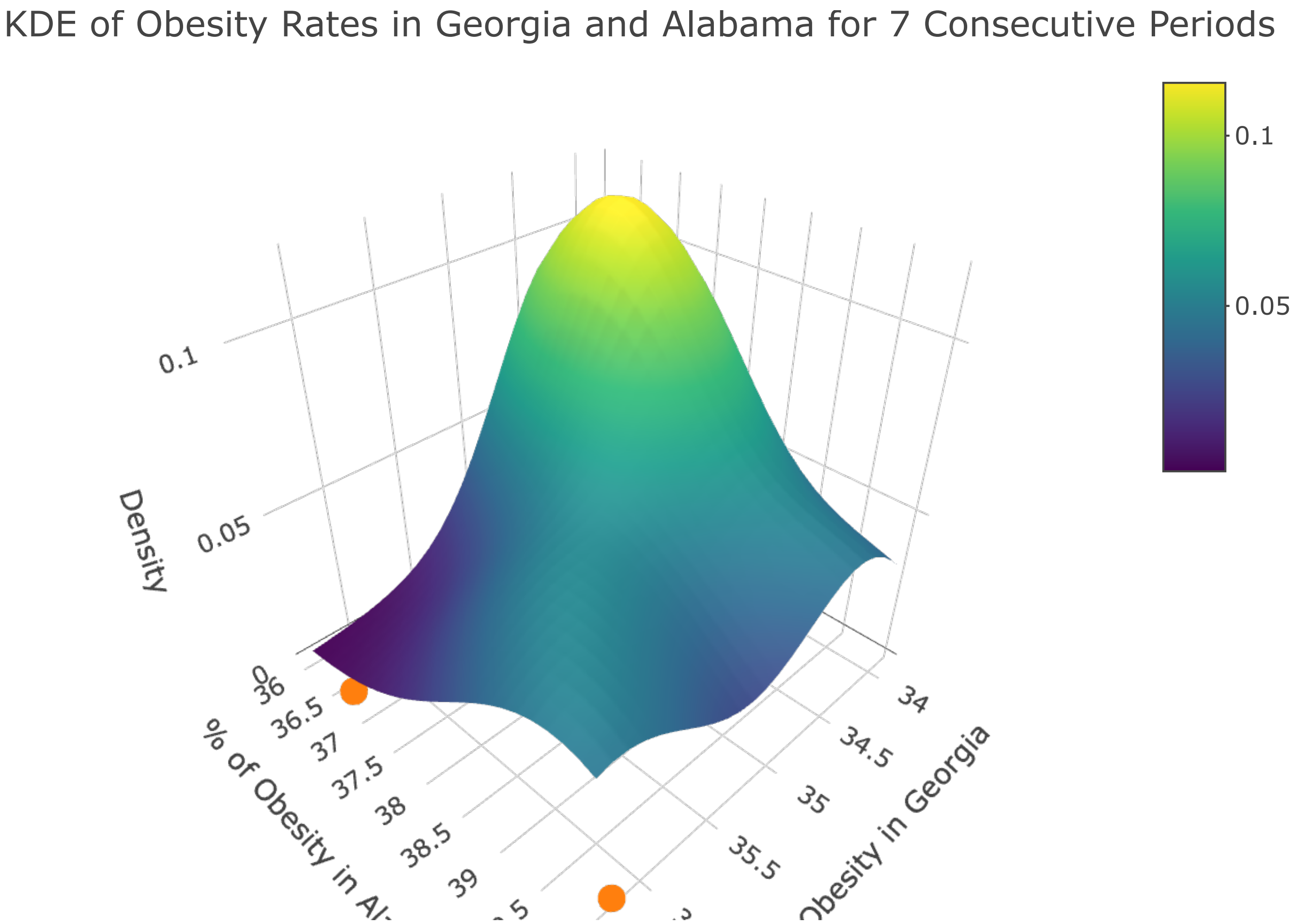
```
library(plotly)

fig <- plot_ly(data_nstates, x = ~data_nstates$Year, y = ~data_nstates$Georgia,
               name = 'Georgia', type = 'scatter', mode = 'lines')
fig <- fig %>% add_trace(y = ~data_nstates$Alabama,
                         name = 'Alabama', mode = 'lines')
fig <- fig %>% add_trace(y = ~data_nstates$Tennessee,
                         name = 'Tennessee', mode = 'lines')
fig <- fig %>% add_trace(y = ~data_nstates$NorthCarolina,
                         name = 'North Carolina', mode = 'lines')
fig <- fig %>% add_trace(y = ~data_nstates$SouthCarolina,
                         name = 'South Carolina', mode = 'lines')
fig <- fig %>% add_trace(y = ~data_nstates$Florida,
                         name = 'Florida', mode = 'lines')
fig <- fig %>% layout(
    title = "Obesity Rates from 2015 - 2021",
    xaxis = list(title = "Year"),
    yaxis = list(title = "Percentage")
    )

fig
```



5d. Kernel Density Estimation (KDE)

For the KDE, we will use Georgia and Alabama State values over 7 Periods (2015 - 2021)

```
surf <- kde2d(data_nstates$Georgia, data_nstates$Alabama)

plot_ly(x = surf$x, y = surf$y, z = surf$z,
        type = "surface") |>
  add_trace(x = data_nstates$Georgia,
            y = data_nstates$Alabama, z = 0,
            type = "scatter3d", mode = "markers") |>
  layout(title = "KDE of Obesity Rates in Georgia and Alabama for 7 Consecutive Perio
ds",
         scene = list(
           xaxis = list(title = "% of Obesity in Georgia "),
           yaxis = list(title = "% of Obesity in Alabama "),
           zaxis = list(title = "Density")
         ))
```



Section 6: Rshiny App

A Rshiny app has been created to further communicate the information from the dataset to the end users. This web app enables you to alternate between Georgia and all other neighboring states, upon which you will get a line graph for the overall obesity from 2015 - 2021.

You are advised to view the app using the Rmd file

```r
#Transformation for Rshiny app
data1 <- data_nstates
rownames(data1) <- data1$Year
data1 <- data1 |> dplyr::select(-Year)

#Rshiny App

ui <- fluidPage(
  titlePanel("Data Visualization for Obesity"),
  radioButtons("state", label = "Choose Georgia or a Neighboring State",
               choices = unique(colnames(data1)),
               selected = colnames(data1)[1]),
  mainPanel(
    plotlyOutput("line_plot")
  )
)

server <- function(input, output, session) {
  onSessionEnded(stopApp)
  output$line_plot <- renderPlotly({
    fig <- plot_ly()

    state <- input$state

    fig <- fig %>% add_trace(
      x = ~rownames(data1),
      y = ~data1[, state],
      name = state,
      type = 'scatter',
      mode = 'lines'
    )

    fig <- fig %>% layout(
    title = "Obesity Rates from 2015 - 2021",
    xaxis = list(title = "Year"),
    yaxis = list(title = "Percentage")
    )
    fig
  })
}

shinyApp(ui, server)
```

Data Visualization for Obesity

Choose Georgia or a Neighboring State

● Georgia

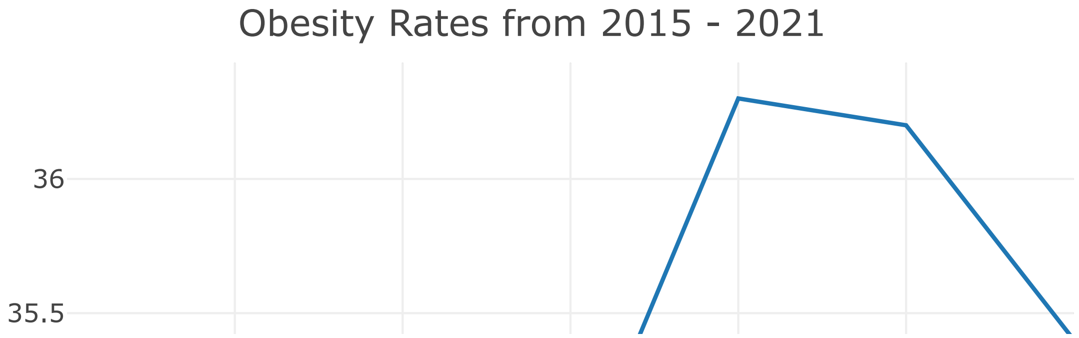
○ Alabama

○ Tennessee

○ NorthCarolina

○ SouthCarolina

○ Florida



Obesity Rates from 2015 - 2021

Section 7: Reporting & Conclusion

In Georgia, For Year 2021:

Age group (50-64)

- The rate of Obesity(%) for Black,Non Hispanic race is considerably higher(51.6%) than the rate in the Hispanic (35.3%) and White, Non-Hispanic races (37.4%)
- The rate of Obesity(%) for Female Gender is slightly higher(42.6%) than the rate of the Male Gender (39.1%)

Age group (65 or Older)

- The rate of Obesity(%) for Black,Non Hispanic race is higher(37.8%) than the rate in White, non-Hispanic race (36.2%)
- The rate of Obesity(%) for Male and Female Genders are similar (28.5% and 29.2% respectively).

Overall

- From 2017 to 2019, Georgia experienced an approximate 3% increase in Obesity.
- From 2019 to 2021, Georgia experienced an approximate 1% decrease in Obesity.

Comparing Means of Georgia and Neighboring States

- Alabama has the highest rate of Obesity while Florida has the lowest rate of Obesity from 2015 - 2021
- Georgia, Tennessee, North Carolina, and South Carolina all have similar rates from 2015 - 2021
- Florida has been able to keep the rate of Obesity comparatively low compared to every other highlighted state through out the period of 2015 - 2021

Section 8: References

1. Temple, N.J. The Origins of the Obesity Epidemic in the USA–Lessons for Today. Nutrients 2022, 14, 4253. https://doi.org/10.3390/nu14204253 (https://doi.org/10.3390/nu14204253)

2. Bleich, S.; Cutler, D.; Murray, C.; Adams, A. Why is the developed world obese? Annu. Rev. Public Health 2008, 29, 273–295

3. Ng, M.; Fleming, T.; Robinson, M.; Thomson, B.; Graetz, N.; Margono, C.; Mullany, E.C.; Biryukov, S.; Abbafati, C.; Abera, S.F.;et al. Global, regional, and national prevalence of overweight and obesity in children and adults during 1980–2013: A systematic analysis for the Global Burden of Disease Study 2013. Lancet 2014, 384, 766–781

4. Rodgers, A.; Woodward, A.; Swinburn, B.; Dietz, W.H. Prevalence trends tell us what did not precipitate the US obesity epidemic. Lancet Public Health 2018, 3, e162–e163.