STA 141A Project: Food Insecurity

Brooke Kerstein, Gabriel Jones

2023-09-13

Introduction

Food insecurity, a pervasive threat to a wide variety of demographics in America, is defined by a lack of accessibility to consistently supplied, adequately nourishing food necessary to live an active, healthy lifestyle. Though much effort has been made to address this basic need issue, Food insecurity has remained a persistent threat over the decades. Feeding America, a non-profit organization dedicated to addressing this issue, reports that at least 34 million adults and 9 million children experience food insecurity every year, and these numbers have only exponentiated as COVID-19 grew to a peak in 2020 and early 2021.

In this report, we wish to address the characteristics of food insecurity and its many factors in California, plotting data retrieved from dedicated census cites such as The United States Census Bureau, CA.gov, and Feeding America's own data caches to investigate the trends of food insecurity over time as well as factors of interest that might prove vital to understanding what exactly contributes to food insecurity. With this knowledge, we plan to construct optimal predictive models using regression techniques and machine learning to ultimately test our own findings regarding the predictors of food insecurity.

*Remark: This PDF is meant to be viewed as a download. Each "...can be viewed **here**..." sentence is a hyperlink to a specific section of the R Appendix. Downloading this report as a PDF will assure this capability is retained.*

Our Factors

Age: Retrieved from the United States Census Bureau, we were interested in investigating age and food insecurity, specifically how different stages of an individual's life might impact their access to food. In this context, we investigated those populations of California that were under 18 or over 65, to assess the tail-ends of the age demographic for any outstanding trends that appear in children or seniors.

Disability: Just as with the Age factor, we retrieved this data from the United States Census Bureau. Our interest in disability status as a potential predictor for food insecurity stems from interpretive intent. In and of itself, we are looking at a time period that has seen much change in terms of disability advocacy and policy action. COVID-19 was a volatile catalyst in disrupting many of the natural data trends that we investigate in this report and disability status, seen throughout California's counties, is one disruption of major note for food insecurity.

Unemployment and Median Income: Unlike our other two factors, unemployment and median income data were taken from CA.gov. This financial data is the pivotal factor regarding food insecurity. We originally had an idea of how financial elements contributed to food insecurity, however, being able to truly visualize this factor, as well as compare it against other factors assisted us with constructing a more insightful predictive model later on.

Data Wrangling

Our first task came in the form of data collection and wrangling. We began a dialogue with the representatives at Feed America and were able to retrieve data ranging from 2010 to 2020 regarding food insecurity rates in every county in America. Because of our desire to localize our scope, we resolved to include and analyze only information pertaining to the counties in California.

Feed America Data 2010-2020

Our Feed America food insecurity data was given to us as a folder containing 11 .xlsx files. Each file contained food insecurity data for every county in the US. Upon exploring the format of each file, we noticed that there were inconsistencies with the columns between data sets, thus we would need to clean each data set individually before merging them all together. As a quick note, our initial data wrangling process was done to include all columns from the data set as we did not know which variables we wanted to use. This version of cleaning can be seen **here** in the R Appendix.

Once we had an idea of how we wanted to structure our analysis, we were able to quickly streamline the process of data wrangling. We cleaned up the 2019-2021 data so that it was in a desirable format, this would act as our baseline format for the other data sets to match. While the data for each year had slight differences from each other, the process of cleaning them followed the same general process. First, we began by selecting only California data and removing any undesired columns. We then changed column names and added a "year" column to store the year that this data was obtained for. This new process can be seen **here** in the R Appendix.

Below is an excerpt of one of the raw files Feed America gave to us:

```
## # A tibble: 6 x 6
##    FIPS State `County, State`          Year `Overall Food Insecurity Rate`
##   <dbl> <chr> <chr>                    <dbl> <chr>
## 1  1001 AL    Autauga County, Alabama  2021 13.3%
## 2  1003 AL    Baldwin County, Alabama  2021 11.8%
## 3  1005 AL    Barbour County, Alabama  2021 17.8%
## 4  1007 AL    Bibb County, Alabama     2021 14.9%
## 5  1009 AL    Blount County, Alabama   2021 13.7%
## 6  1011 AL    Bullock County, Alabama  2021 15.6%
## # i 1 more variable: `# of Food Insecure Persons Overall` <dbl>
```

And here is an excerpt of our final Feed America 2010-2020 data frame, which includes only clean information regarding food insecurity within California's counties from 2010-2020:

```
##   year         county overall_food_insecurity_rate
## 1 2010 Alameda County                          16.4
## 2 2011 Alameda County                          16.2
## 3 2012 Alameda County                          15.6
## 4 2013 Alameda County                          15.3
## 5 2014 Alameda County                          14.9
## 6 2015 Alameda County                          14.3
```

Age Data 2019

To retain simplicity, as well as avoid any unnecessary exercises in data wrangling, we decided to first retrieve age data for one year, 2019, and analyze that data, as well as 2019 data for our other factors. We chose 2019 as our year of reference as it represents the most recent data that was not impacted by the effects of

COVID-19. We utilized data provided in our disability data set for 2019, retrieved from The United Census Bureau, to build a clean data frame of the age data we required. This file contained much information that was not necessary for the analysis that we wanted to perform. In conjunction with this influx of information, the column naming conventions served to complicate our understanding of the data.

We first limited our data set to only counties in California and then filtered based on "Estimate", as these columns provided whole number data we could evaluate. We further trimmed our data and found, as discussed in the introduction, that the most interesting columns contained information at the two age extremes, children (<18) and seniors (>65). With this in mind, we further tailored our data to display this information in an easily understandable way. The entire age data wrangling technique for 2019 can be seen **here** in the R Appendix.

Below is an excerpt of the un-wrangled age data:

```
## # A tibble: 6 x 3
##   GEO_ID         NAME                    S1810_C01_001E
##   <chr>          <chr>                   <chr>
## 1 Geography      Geographic Area Name    Estimate!!Total!!Total civilian nonins~
## 2 0400000US06    California              38997581
## 3 0500000US01003 Baldwin County, Alabama 220911
## 4 0500000US01015 Calhoun County, Alabama 111075
## 5 0500000US01043 Cullman County, Alabama 82841
## 6 0500000US01049 DeKalb County, Alabama  70392
```

And here is an excerpt of our final age 2019 data frame, which includes only clean information regarding ages <18 and >65:

```
## # A tibble: 6 x 4
##   county              total_population population_under_18 population_over_65
##   <chr>                          <dbl>               <dbl>              <dbl>
## 1 Alameda County               1661492               0.205              0.141
## 2 Butte County                  217007               0.204              0.183
## 3 Contra Costa County          1148991               0.226              0.160
## 4 El Dorado County              191502               0.200              0.219
## 5 Fresno County                 987054               0.285              0.125
## 6 Humboldt County               134571               0.190              0.185
```

Disability Data 2010-2020

We used the same techniques from the Age Data 2019 section to wrangle with the Disability data. First, we wrangled with 2019 data, however, as we developed our understanding of the relationship between Disability and food insecurity, we expanded our wrangling to 2010 through 2018, as well as 2020. Our interest, to preserve simplicity, scoped across all recognized disability categories; hearing difficulty, vision difficulty, ambulatory difficulty, cognitive difficulty, and self-care difficulty. The entire disability data wrangling technique for 2010 through 2012 and 2013 through 2020 can be seen **here** in the R Appendix.

Here is an excerpt of our final disability 2010-2020 data frame, which includes only clean information regarding disability within California's counties from 2010-2020:

```
##   year         county percent_disabled
## 1 2010 Alameda County              8.7
## 2 2011 Alameda County              9.1
## 3 2012 Alameda County              9.1
## 4 2013 Alameda County              9.6
## 5 2014 Alameda County              9.6
## 6 2015 Alameda County              9.9
```

Unemployment Data and Median Income Data 2010-2020

The financial data we used was sourced from data.ca.gov and consisted of two different data sets containing information on unemployment and median income in California. Both of these data sets were relatively clean, thus the wrangling process was fairly simple. For the unemployment data, we filtered the data to select only California counties and the years 2010-2021. We then calculated the means for each county during each year because we were given multiple data points for each county-year combination. The entire unemployment wrangling technique for 2010 through 2020 can be seen **here** in the R Appendix.

Below is an excerpt of the un-wrangled unemployment data:

```
## # A tibble: 6 x 3
##   area_type         area_name                          date
##   <chr>             <chr>                              <chr>
## 1 State             California                         01/01/1976
## 2 State             California                         01/01/1976
## 3 County            Los Angeles County                 01/01/1976
## 4 County            Los Angeles County                 01/01/1976
## 5 Metropolitan Area Los Angeles-Long Beach-Glendale MD 01/01/1976
## 6 Metropolitan Area Los Angeles-Long Beach-Glendale MD 01/01/1976
```

Here is an excerpt of our final unemployment 2010-2020 data frame, which includes only clean information regarding unemployment within California's counties from 2010-2020::

```
## # A tibble: 6 x 3
##    year county           unemployment_rate_avg
##   <dbl> <chr>                            <dbl>
## 1  2010 Alameda County                   0.111
## 2  2010 Alpine County                    0.136
## 3  2010 Amador County                    0.144
## 4  2010 Butte County                     0.142
## 5  2010 Calaveras County                 0.148
## 6  2010 Colusa County                    0.213
```

The median income data was cleaned in a similar fashion. Cleaning processes unique to this data set included removing unnecessary counties from the "county" column, and merging the income data with our main data frame from 2010-2020 rather than 2010-2021. This was because our income data did not contain any information from 2021. The entire median income wrangling technique for 2010 through 2020 can be seen **here** in the R Appendix.

Below is an excerpt of the un-wrangled median income data:

```
## # A tibble: 6 x 3
##   taxable_year county          population
##          <dbl> <chr>                <dbl>
## 1         2016 Mono                 13801
## 2         2006 Nonresident              0
## 3         2010 Sacramento         1420447
## 4         2006 San Diego          3077877
## 5         2019 El Dorado           192012
## 6         1999 San Luis Obispo     240500
```

Here is an excerpt of our final median income 2010-2020 data frame, which includes only clean information regarding median income within California's counties from 2010-2020::

```
## # A tibble: 6 x 3
##    year county           median_income
##   <dbl> <chr>                    <dbl>
## 1  2010 Alameda County           41936
## 2  2010 Alpine County            37999
## 3  2010 Amador County            35819
## 4  2010 Butte County             28709
## 5  2010 Calaveras County         34171
## 6  2010 Colusa County            28051
```

Final Data Frame

Our final Data frame for 2019 contained food insecurity rate per county, unemployment rate per county, median income per county, percent individuals who identified as disabled per county, and proportion of the population under 18 or over 65. A separate data frame was created for all factor information excluding age data. This data frame spans from 2010 to 2020 and was used to simplify visualization and modeling throughout the rest of the report. The methodology used to compile all of this information is available in the R Appendix. The methodology used to compile all of this information can be seen **here** in the R Appendix.

Our final factor data frame for 2019 is observed in part below.

```
## # A tibble: 6 x 9
##   county           total_population population_under_18 population_over_65  year
##   <chr>                       <dbl>               <dbl>              <dbl> <dbl>
## 1 Alameda County            1661492               0.205              0.141  2019
## 2 Butte County               217007               0.204              0.183  2019
## 3 Contra Costa Co~          1148991               0.226              0.160  2019
## 4 El Dorado County           191502               0.200              0.219  2019
## 5 Fresno County              987054               0.285              0.125  2019
## 6 Humboldt County            134571               0.190              0.185  2019
## # i 4 more variables: overall_food_insecurity_rate <dbl>,
## #   percent_disabled <dbl>, unemployment_rate_avg <dbl>, median_income <dbl>
```

Our final factor data frame excluding age for 2010 through 2020 is observed in part below.

```
##   year         county overall_food_insecurity_rate percent_disabled
## 1 2010 Alameda County                         16.4              8.7
## 2 2011 Alameda County                         16.2              9.1
## 3 2012 Alameda County                         15.6              9.1
## 4 2013 Alameda County                         15.3              9.6
## 5 2014 Alameda County                         14.9              9.6
## 6 2015 Alameda County                         14.3              9.9
```

Analysis

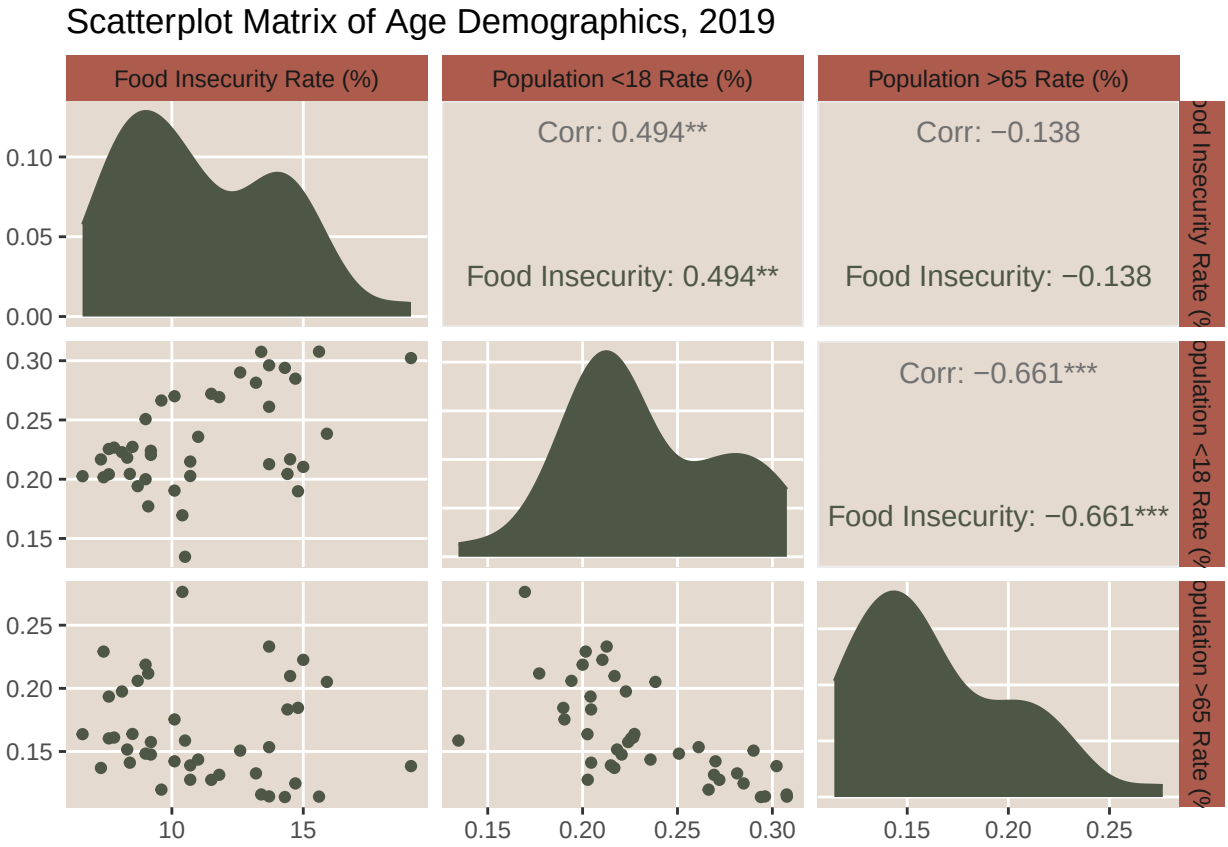
Data Visualization

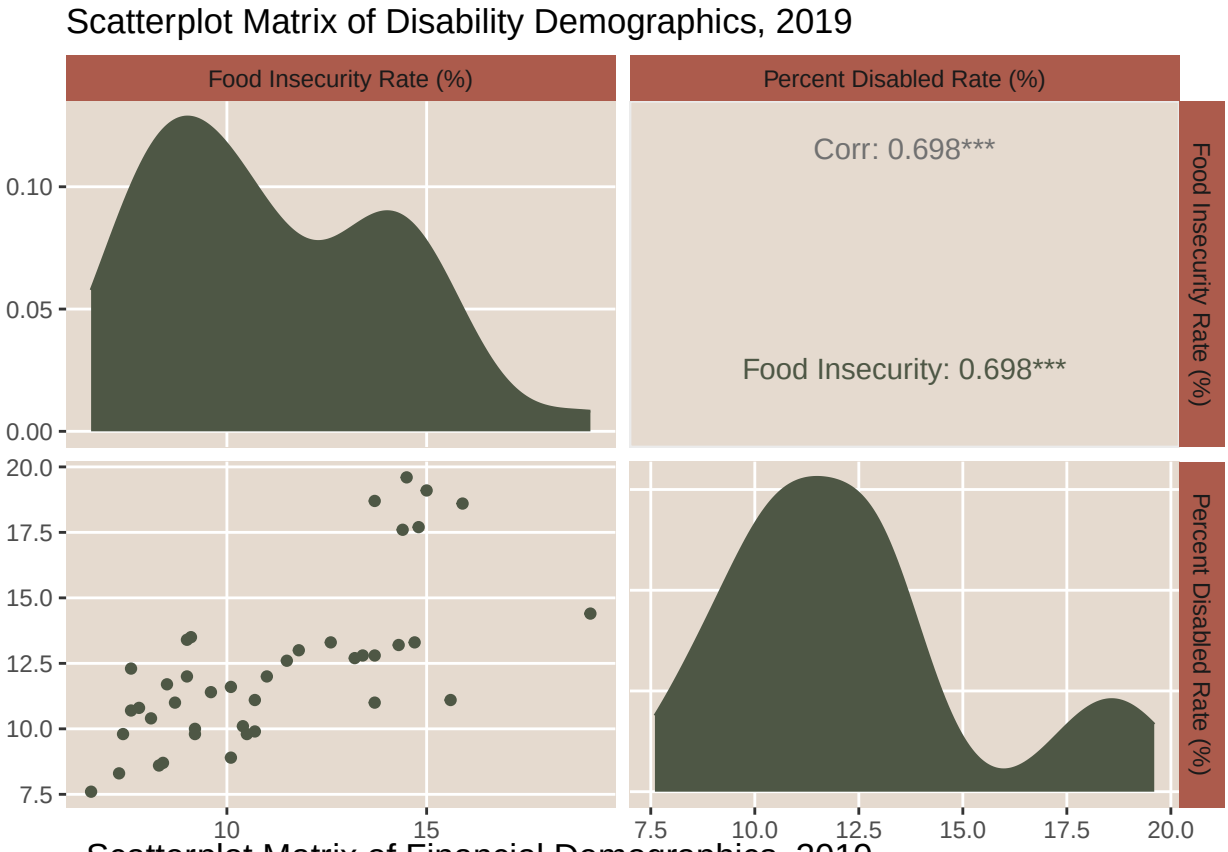
Correlation Matrices

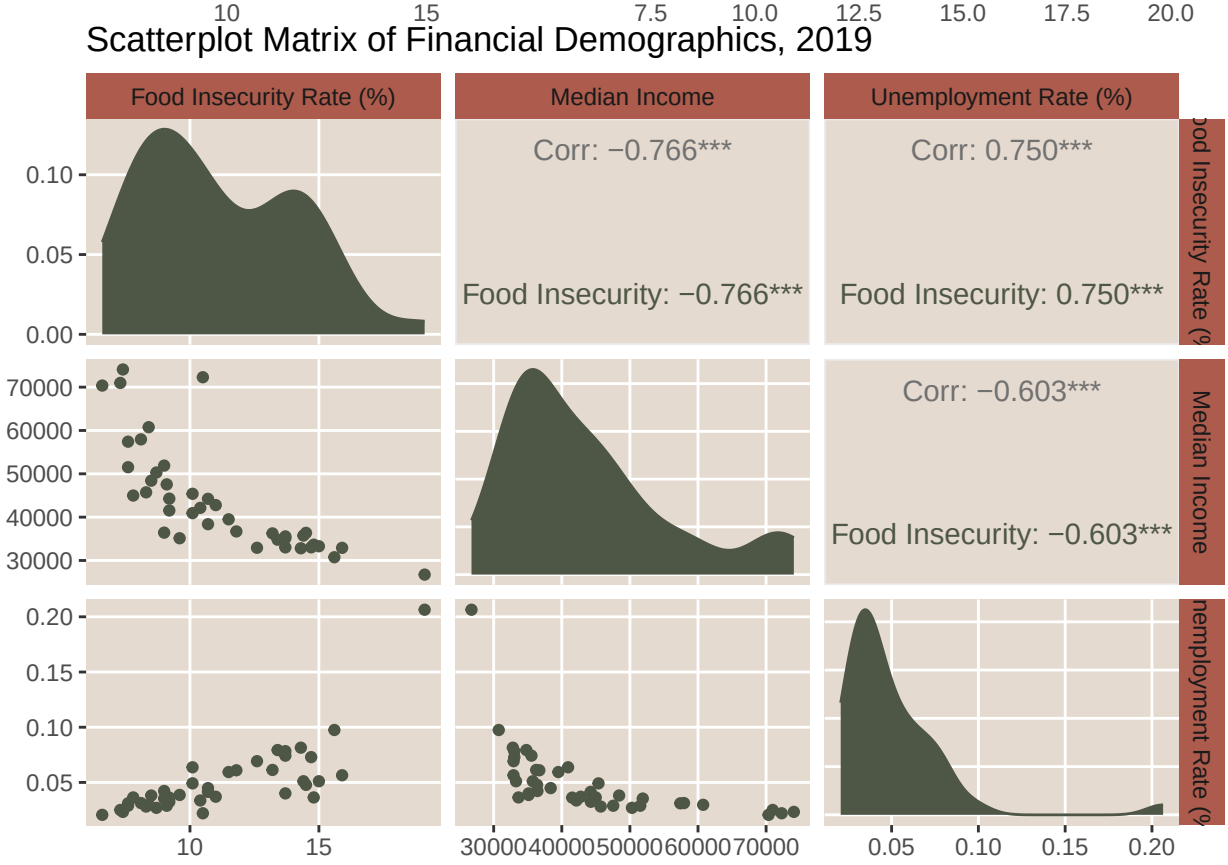
With our clean data frame, we were able to begin analysis. Before visualization, we decided to assess the correlation between food insecurity and the factors described in the introduction to rule out any factors that might not be entirely applicable. As described previously, we chose the year 2019 for this correlation

assessment. All of the correlation matrices for our factors are depicted below. The methodology used to produce these matrices can be seen **here** in the R Appendix.

Here are all of our correlation matrix visualizations:





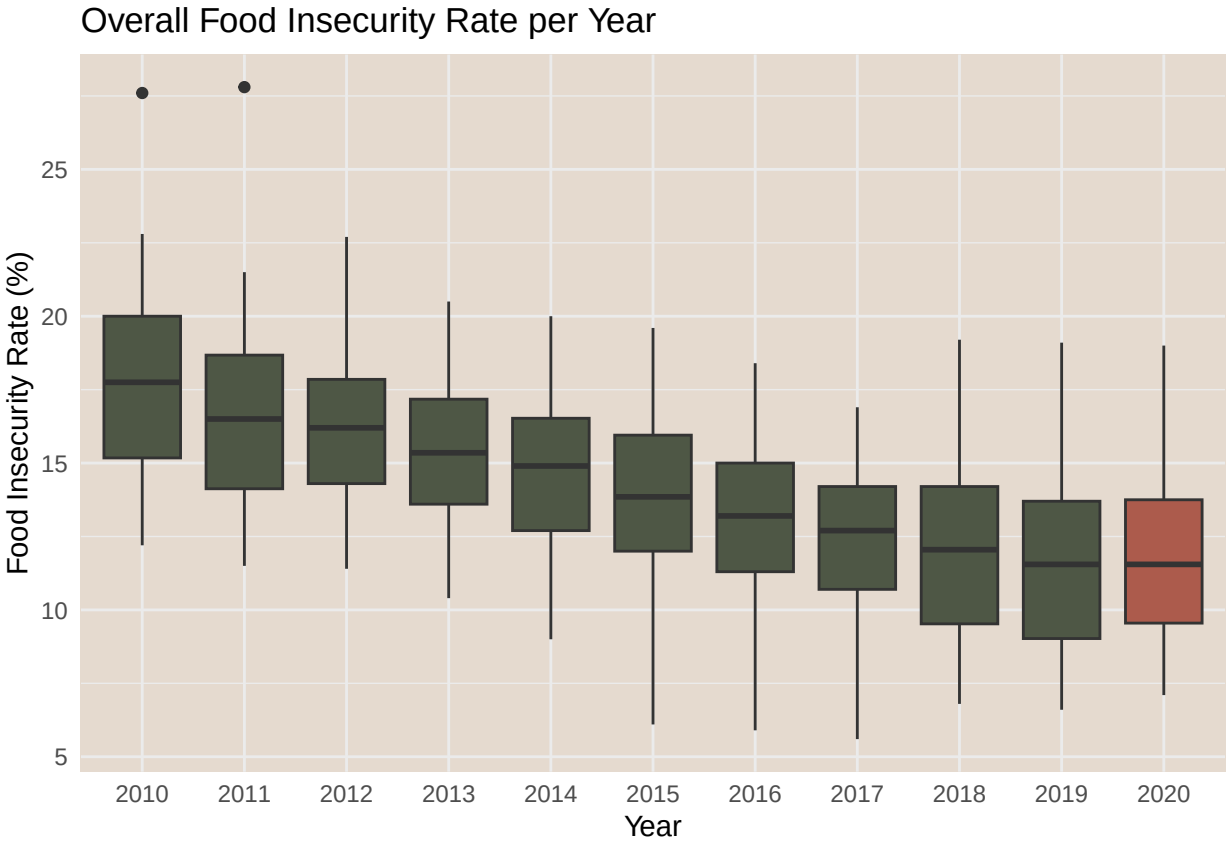


It is clear that the disability, unemployment, and median income factors all possessed the highest correlations

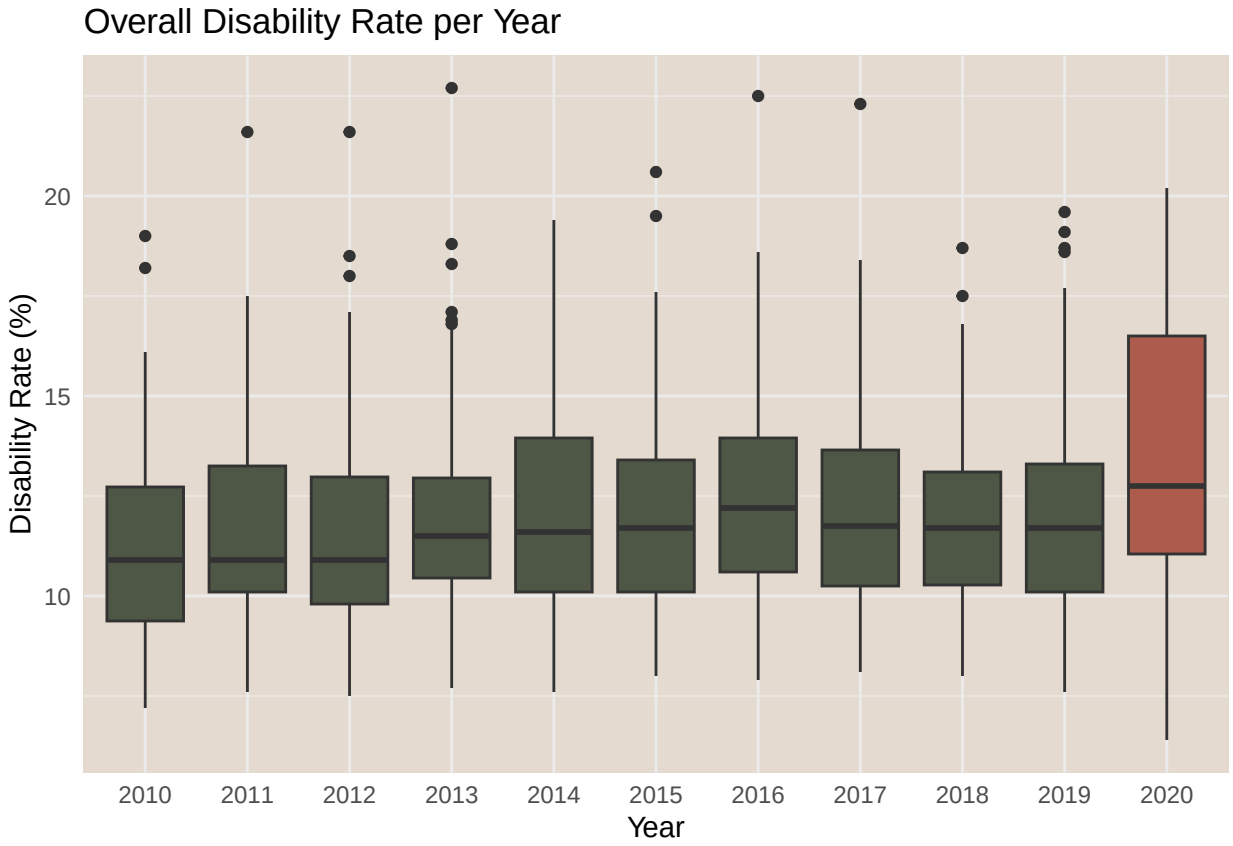
with the food insecurity data; 0.689, 0.750, and -0.766 respectively. These relationships allow us to determine, as well, that as income increases, food insecurity decreases, and as disability rate and unemployment rate increase, so does food insecurity. More visualization, however, is warranted to further our understanding of these factors.

Box Plots

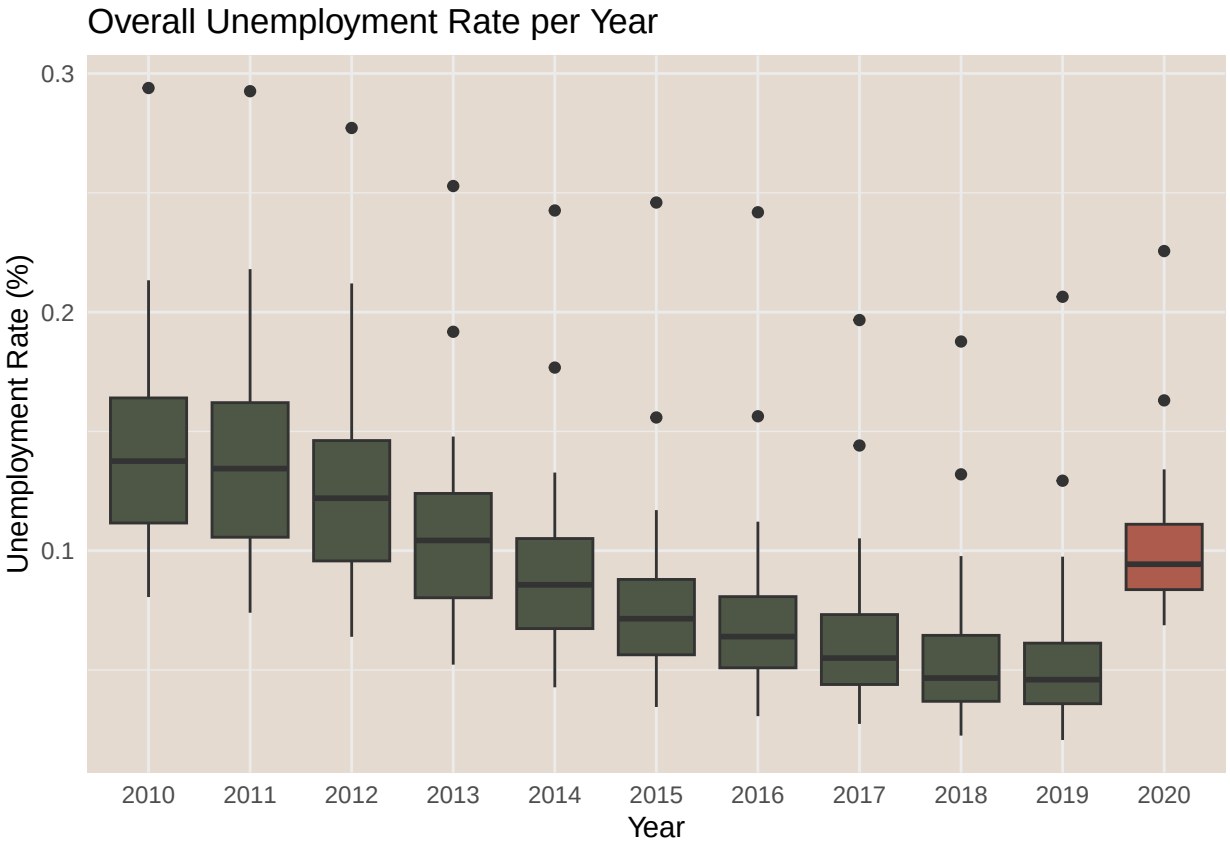
We further used yearly boxplots to show the inter-quantile range of each variable during each year. Through these plots, we were able to gain some extra insights as to what we would expect from the relationships between our determining factors and food insecurity rate. This ultimately allows us to understand how to use statistical models to forecast food insecurity rate. The methodology used to produce these boxplot graphs can be seen **here** in the R Appendix.



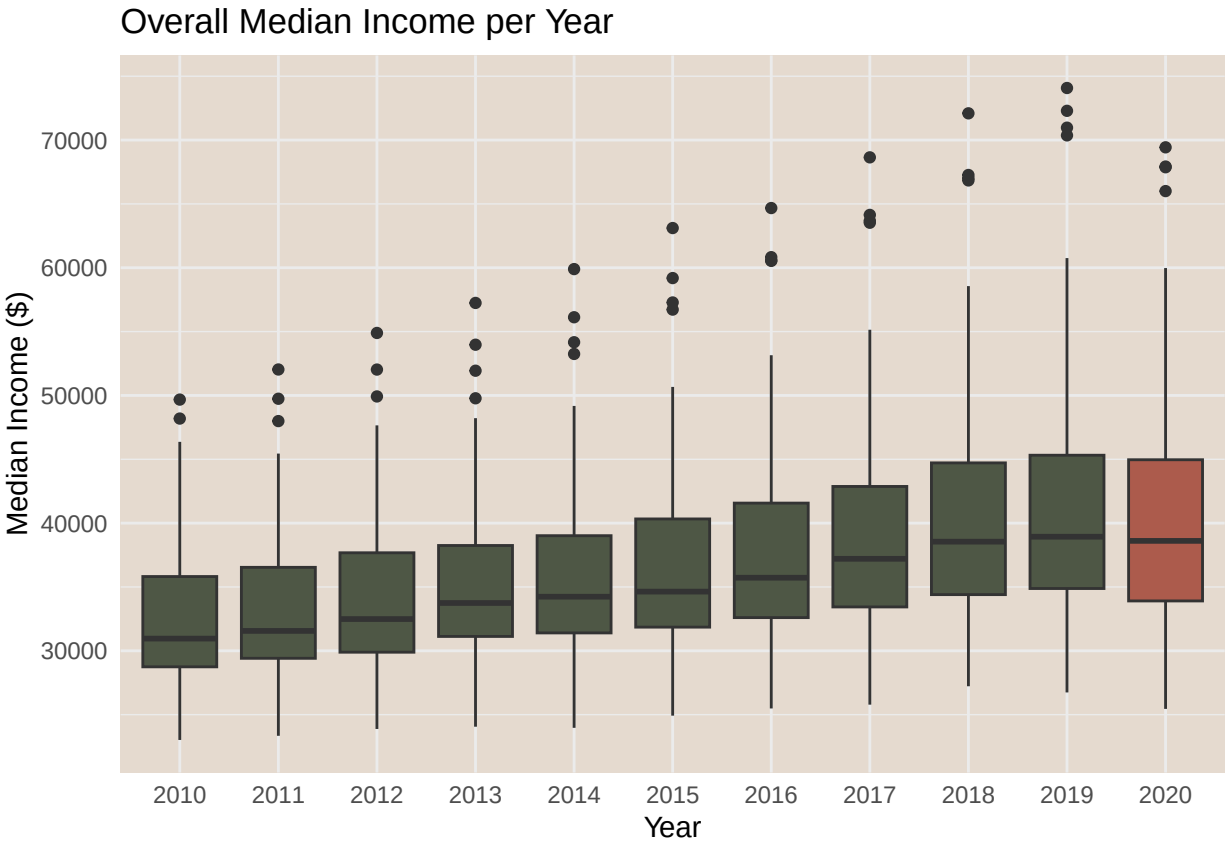
Starting with the yearly food insecurity rate, we notice that there is a general downturn in food insecurity each year from 2010-2019. We also notice that this downturn is disrupted in 2020 where the food insecurity rate stays relatively the same as the previous year. We interpret this as having to do with the events of COVID-19 and the sudden lack of resources available to those in need of assistive programs such as Feeding America.



This boxplot, which depicts disability rate across counties over time, displays an interesting characteristic, remaining fairly level until 2019 and then shooting up in both median and interquartile range in 2020. This leads to speculation on the effects of the events of 2020 with regard to disability awareness. COVID-19 greatly limited the capability of assistive services such as caretaking and transportation programs. It is possible that those individuals who did not identify as disabled reassessed their status due to new accessibility issues. The residual effects of COVID-19, as well, left many with sudden sensory and ambulatory issues, though these issues, for some, might have lessened over time, they remained impediments to lifestyle throughout 2020. This information is liable to produce skewed accuracy results when predictive modeling is performed. Surprisingly low accuracy is expected for prediction in relation to 2020 values.



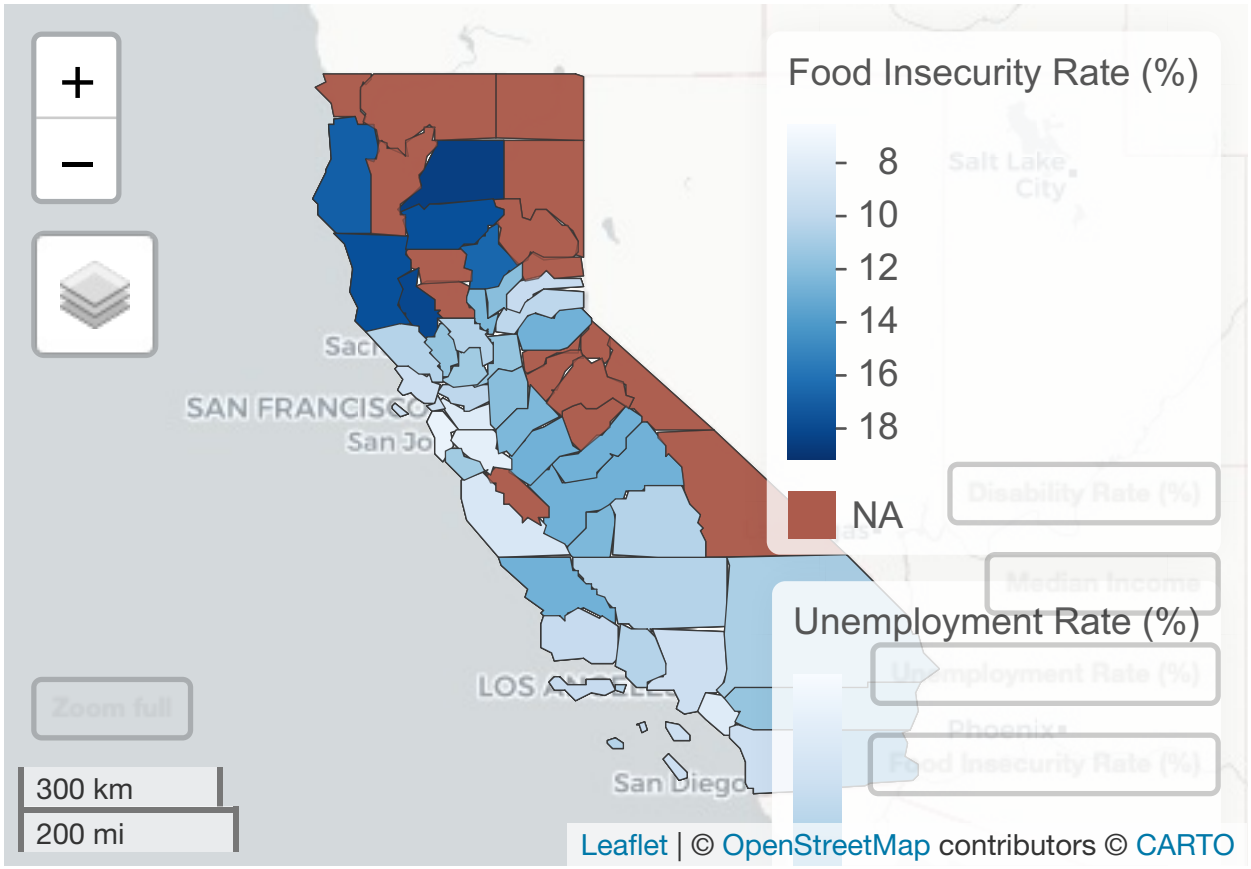
The box plot for unemployment rate in California shows a trend similar to the previous disability rate graph. Here we observe a large downturn in the unemployment rate from 2010-2019, however, we see a large increase in unemployment rate in 2020. This is likely caused by the lockdown issued in 2020 where many people were no longer able to attend their jobs. This inconsistency may make predicting food insecurity in 2020 less accurate.



When performing the same visualizations for median income, we observe a similar phenomenon as the food insecurity rate. Here, we notice an upward trend in median income from 2010-2019 followed by a disruption in 2020 where the IQR remains similar to the previous year. While this provides evidence that median income would be useful in forecasting food insecurity, our predictions based on this may be disrupted by the outliers shown in the graph. These outliers are likely the median income for counties with higher-income job opportunities.

California County Map

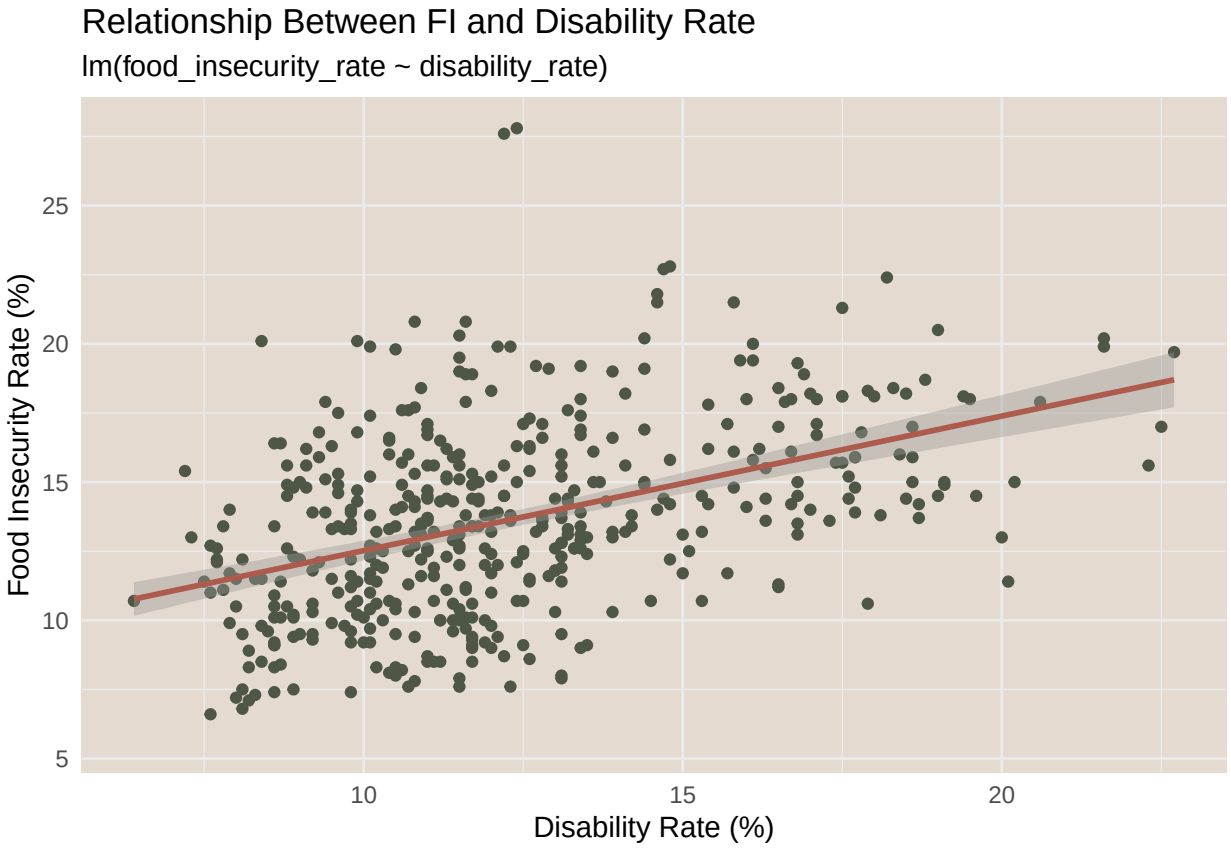
As a final step to visualizing the relationships between food insecurity and all of its factors, we created a map visualization with shapefile. This map depicts a blue-toned gradient to exemplify food insecurity, unemployment, and disability rates as well as median income in dollars projected onto county limits. The red counties contain incomplete or no data from our datasets. As we iterate through the interactive layers, we can see that the map showed a potentially negative relationship between food insecurity and income, and positive relationships between food insecurity and unemployment status and disability. Because we are presenting this report in a pdf format, we were unable to embed the HTML required to make this map interactive, however, we provide this code for interested parties in email format, separate from this report. The methodology used to produce this map can be seen **here** in the R Appendix.



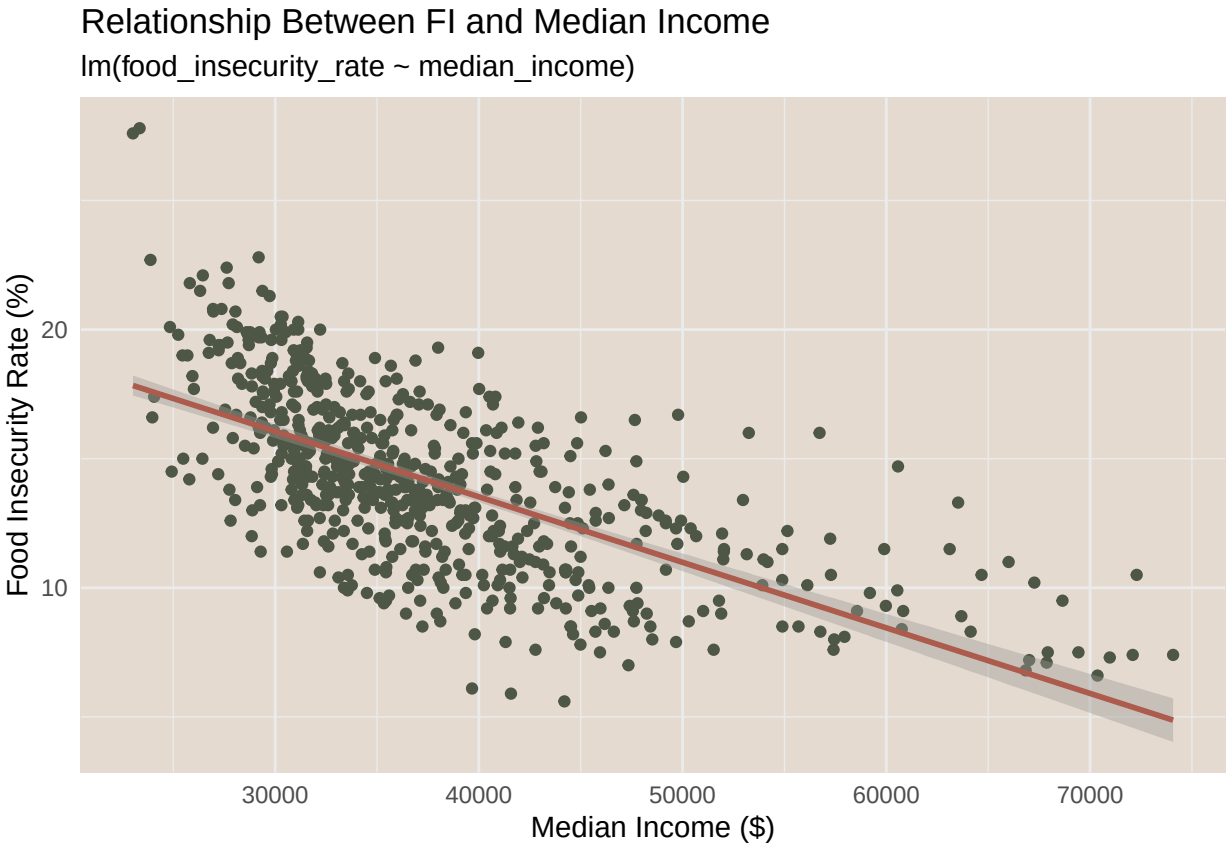
Modeling

Linear Models

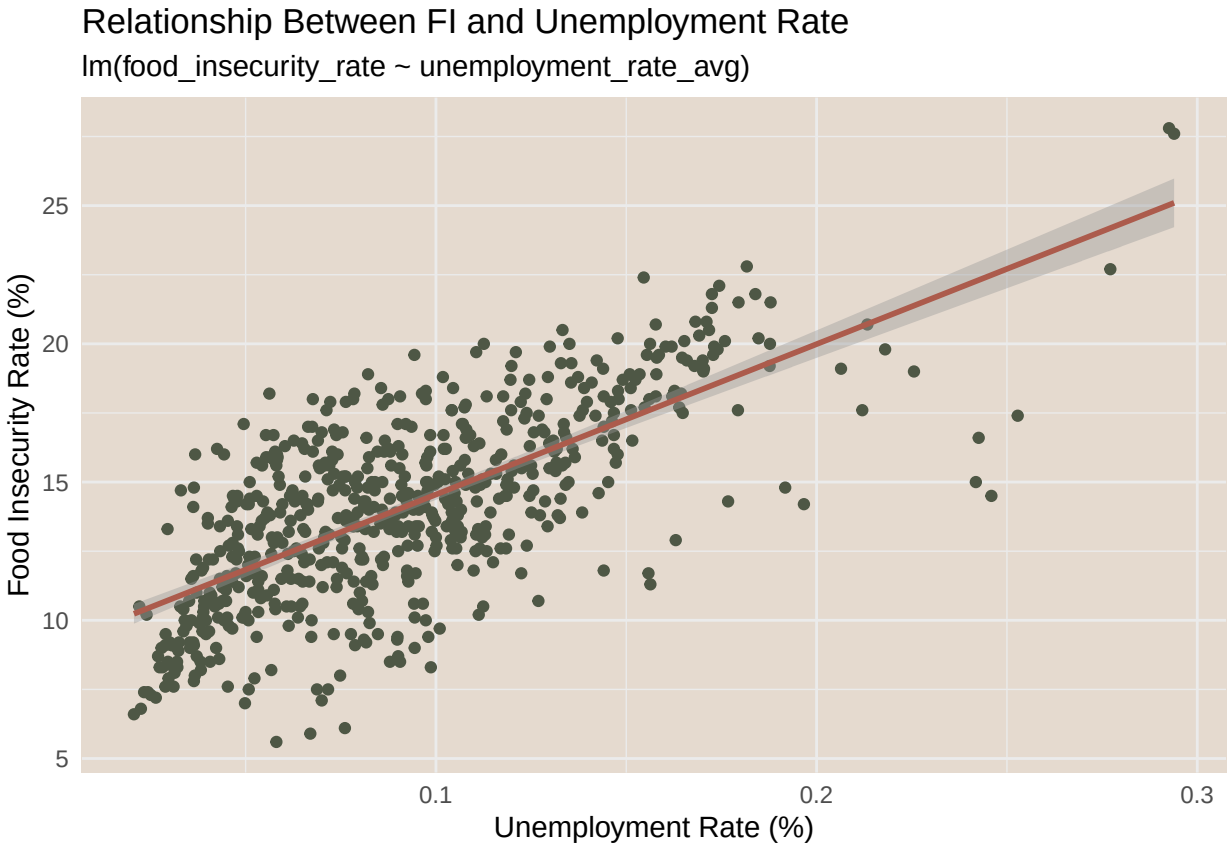
We began our modeling process by fitting linear regression models to all three of our predictors. The general purpose behind these visualizations was for us to get a better idea of the relationship between our predictors and food insecurity rate as we will not be able to visualize them all together in a multivariate linear model. The methodology used to produce these models can be seen **here** in the R Appendix.



When fitting disability rate to food insecurity, we observe that the two variables have a slight positive relationship. This relationship may have inaccuracies in predicting food insecurity for different counties due to the variability across each county. Fitting a model to each county individually may increase the accuracy of a multivariate model including disability rate as a predictor.



When fitting median income to food insecurity, we observe that the two variables have a negative relationship. This relationship is better defined in comparison to the disability rate model, however, the large number of data points in the \$30000-\$40000 range may make it difficult to accurately predict the food insecurity rate for counties with a median income within that range.



When fitting unemployment rate to food insecurity, we observe that the two variables have a positive relationship. This model has the same concern as the median income model as counties lying in the 0-10% unemployment rate may be inaccurately estimated due to their variability. Due to the concerns of all three of these linear models, we decided to fit our predictive models for both all counties and for each county individually to compare the accuracy of each.

AIC

We wanted, to solidify our plans for our predictive modeling, to perform AIC on each iteration of prediction modeling. We use a forward-step directional approach, as we wanted to see the progression of improvement in each iteration of factor adding. This provides visual evidence for the argument of full model usage for predictive modeling. It is observable here that, in terms of linear regression, the addition of all three factors into the predictive model, unemployment rate, disability, and median income, results in the smallest AIC, and consequently, the best fit linear model. In the following sections, we use this information to construct our models. The methodology used to produce this AIC result can be seen **here** in the R Appendix.

```
##
## Call:
## lm(formula = overall_food_insecurity_rate ~ unemployment_rate_avg +
##     percent_disabled + median_income, data = na.omit(majorDF))
##
## Coefficients:
##           (Intercept)  unemployment_rate_avg       percent_disabled
##             9.159e+00              4.261e+01              2.716e-01
##         median_income
##            -6.762e-05
```

Predictive Modeling

We decided to use two approaches to predicting food insecurity based on our predictive variables to compare which model performed best. The first model we used was an additive linear model which fit unemployment rate, disability rate, and median income together to food insecurity. We chose this model because we observed linear relationships between all three of our predictors and food insecurity. This model was also relatively simple to implement. The second model we used was a random forest model using the same formula as the additive linear model. This model was chosen as a means of reducing the variance in our data, which should increase the accuracy of our predictions.

When selecting our test data, we decided to test against both 2019 and 2020 data. This was done in consideration of the trend disruptions between values in 2019 and 2020 for all of our predictors. The expectation from this was that the predictive models would have better accuracy in predicting food insecurity for 2019 than for 2020. Thus, for our training data, we used data between 2010-2018 and 2010-2019, with each being tested against 2019 and 2020 respectively.

The methodologies used to produce the general Linear predictive and Random Forest models against 2019 can be seen here in the R Appendix.

The methodologies used to produce the general Linear predictive and Random Forest models against 2020 can be seen here in the R Appendix.

Lastly, to further address the concerns brought up in the discussion of our single-factor linear models, we wrote our predictive models using both data points regardless of county and models fit for each county individually. The expectation from this was that the models fit for each county would be significantly more accurate due to accounting for each county's trends. Our only concern from this was not having sufficient data points to fit these models towards.

The methodologies used to produce the by county Linear predictive models against 2019 and 2020 can be seen here in the R Appendix.

The methodologies used to produce the by county Random Forest models against 2019 and 2020 can be seen here in the R Appendix.

Predictive Modeling Results: Accuracy

After fitting our models and calculating their accuracy, we found that the general predictions for food insecurity in 2019 were more accurate than our predictions for 2020. This again, was likely due to data in 2019 following the same general trend as previous years, while data in 2020 had a disruption to that trend. When comparing within test groups, we find that the overall random forest model and by county linear model performed the best when testing against the 2019 data, with the random forest performing slightly better. When testing against the 2020 data, we found that fitting the random forest model by county performed significantly better than the other models tested against 2020. The methodology used to produce this accuracy results table can be seen **here** in the R Appendix.

Model.Type	Test.Accuracy.2019	Test.Accuracy.2020
Overall Linear Regression	0.8671502	0.6799830
Overall Random Forest	0.9342458	0.6690453
By County Linear Model	0.9253086	0.7075946
By County Random Forest	0.8699000	0.8220620

Conclusion

Our interest in understanding the factors involved in food insecurity in California led us to identify three different factors we reasoned might be useful in predicting food insecurity. We found that of the three factors; Age, Disability, and Financial status, disability and financial status were most highly correlated with food insecurity. Plotting these factors across the years in a boxplot graph, we saw that food insecurity had a general downward trend, disrupted by the boxplot for 2020. Disability had a fairly level trend until

2020, when it spiked significantly. Income had an upward trend and unemployment status had a downward trend. There were similar disruptions to these trends as with food insecurity and disability, in the year 2020. All of these disruptions can be attributed, in part, to the effects of the COVID-19 pandemic. For further visual comparison, we constructed a shapefile-generated map of all the counties in California. We layered each factor excluding age on top of the food insecurity map visualization and could see relationships between all of the factors mapped and food insecurity. The map showed a potentially negative relationship between food insecurity and income and positive relationships between food insecurity and unemployment status and disability. We linearly modeled these relationships, and confirmed what we had observed when plotting our boxplot graphs and the map. In preparation for predictive modeling, we assessed model fitness with AIC, observing that food insecurity regressed with unemployment status, disability, and median income resulted in the most optimal model parameters. We then compared two predictive models, linear regression and random forest set at two conditions; disruption or no disruption exemplified by testing against 2019 (no disruption) or 2020 (disruption). We concluded that the random forest model performed best when predicting future food insecurity rates. Additionally, we found that generating models for each individual county performed better when predicting food insecurity rates in counties that experienced disruption to their trends in unemployment rate, median income, and disability rates.

Resources

Feeding America Data, Map the Meal Gap: https://map.feedingamerica.org/county/2021/overall/california

U.S. Census Bureau, Disability Data: https://data.census.gov/table?q=California+Disability+by+County

U.S. Census Bureau, Mapview: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html1

CA.gov, Labor Market Information Division: https://data.edd.ca.gov/Labor-Force-and-Unemployment-Rates/Local-Area-Unemployment-Statistics-LAUS-/e6gw-gvii

CA.gov, CA Open Data Portal: https://data.ca.gov/dataset/b-6-comparison-by-county2

R Appendix

Libraries

```
# Libraries: Check for installed packages before loading
list.of.packages <- c("dplyr", "tidyr", "ggplot2", "hrbrthemes", "gganimate",
                      "png", "gifski", "ggridges", "tidyverse", "tibble",
                      "mapview", "sp", "janitor", "GGally", "RColorBrewer",
                      "MASS", "knitr","matlib", "lubridate", "pdftools",
                      "stringr", "ggmap", "ggsci", "patchwork", "ddpcr",
                      "caret", "car", "paletteer")

new.packages <-
  list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
lapply(list.of.packages, require, character.only=TRUE)
```

Importing All Data

```
#Import All Data Sets
rawdata2010 <- readr::read_csv("FeedAmerica/FeedAmerica2010.csv")
rawdata2011 <- readr::read_csv("FeedAmerica/FeedAmerica2011.csv")
rawdata2012 <- readr::read_csv("FeedAmerica/FeedAmerica2012.csv")
rawdata2013 <- readr::read_csv("FeedAmerica/FeedAmerica2013.csv")
rawdata2014 <- readr::read_csv("FeedAmerica/FeedAmerica2014.csv")
rawdata2015 <- readr::read_csv("FeedAmerica/FeedAmerica2015.csv")
rawdata2016 <- readr::read_csv("FeedAmerica/FeedAmerica2016.csv")
rawdata2017 <- readr::read_csv("FeedAmerica/FeedAmerica2017.csv")
rawdata2018 <- readr::read_csv("FeedAmerica/FeedAmerica2018.csv")
rawdata2019_2021 <- readr::read_csv("FeedAmerica/FeedAmerica2019_2021.csv")

AgeData2019 <- readr::read_csv("DisabilityData/DisabilityData2019.csv")

unemploymentdataraw <- readr::read_csv("Local_Area_Unemployment_Statistics__LAUS_.csv")
unemploymentdataraw <- janitor::clean_names(unemploymentdataraw)

incomedataraw <- readr::read_csv("incomedata.csv")
incomedataraw <- janitor::clean_names(incomedataraw)

DisData2021 <- readr::read_csv("DisabilityData/DisabilityData2021.csv")
DisData2020 <- readr::read_csv("DisabilityData/DisabilityData2020.csv")
DisData2019 <- readr::read_csv("DisabilityData/DisabilityData2019.csv")
DisData2018 <- readr::read_csv("DisabilityData/DisabilityData2018.csv")
DisData2017 <- readr::read_csv("DisabilityData/DisabilityData2017.csv")
DisData2016 <- readr::read_csv("DisabilityData/DisabilityData2016.csv")
DisData2015 <- readr::read_csv("DisabilityData/DisabilityData2015.csv")
DisData2014 <- readr::read_csv("DisabilityData/DisabilityData2014.csv")
DisData2013 <- readr::read_csv("DisabilityData/DisabilityData2013.csv")
DisData2012 <- readr::read_csv("DisabilityData/DisabilityData2012.csv")
DisData2011 <- readr::read_csv("DisabilityData/DisabilityData2011.csv")
DisData2010 <- readr::read_csv("DisabilityData/DisabilityData2010.csv")
```

Original Feed America Cleaning Method

```
# Clean up the baseline data frame that we will be adding yearly data to
cleandata2019 <- rawdata2019_2021 %>%
  select(
    -"Food Insecurity Rate among Black Persons (all ethnicities)",
    -"Food Insecurity Rate among Hispanic Persons (any race)",
    -"Food Insecurity Rate among White, non-Hispanic Persons",
    -"Weighted weekly $ needed by FI"
  ) %>%
  filter(State == "CA")
cleandata2019 <- janitor::clean_names(cleandata2019)

# Clean up other yearly data and put it in the same format as our baseline data frame
cleandata2010 <- rawdata2010 %>%
  # Select Desired Data
  filter(State == "CA") %>%
  select(-"...19") %>%
```

```
  mutate("Year" = rep(2010, length(.$State))) %>%
  # Clean Names
  rename(., "Overall Food Insecurity Rate" = "2010 Food Insecurity Rate") %>%
  rename(., "# of Food Insecure Persons Overall" = "# Food Insecure Persons in 2010") %>%
  rename(., "Child Food Insecurity Rate" = "2010 Child food insecurity rate") %>%
  rename(., "# of Food Insecure Children" = "# of Food Insecure Children in 2010") %>%
  rename(., "Cost Per Meal" = "2010 Cost Per Meal") %>%
  rename(., "Weighted Annual Food Budget Shortfall" =
           "2010 Weighted Annual Food Budget Shortfall") %>%
  rename(., `% food insecure children in HH w/ HH incomes above 185 FPL` =
           `% food insecure children w/ incomes above 185 FPL`) %>%
  mutate("% FI > Low Threshold" = rep("NA", length(.$Year))) %>%
  as.data.frame(.)

# Make columns the same type
cleandata2010$`# of Food Insecure Children` <-
  as.character(cleandata2010$`# of Food Insecure Children`)
cleandata2010 <- janitor::clean_names(cleandata2010) %>%
  select(
    fips,
    state,
    county_state,
    year,
    overall_food_insecurity_rate, number_of_food_insecure_persons_overall,
    low_threshold_in_state, low_threshold_type, high_threshold_in_state,
    high_threshold_type, percent_fi_low_threshold, percent_fi_btwn_thresholds,
    percent_fi_high_threshold, percent_fi_low_threshold_2,
    child_food_insecurity_rate, number_of_food_insecure_children,
    percent_food_insecure_children_in_hh_w_hh_incomes_below_185_fpl,
    percent_food_insecure_children_in_hh_w_hh_incomes_above_185_fpl,
    cost_per_meal, weighted_annual_food_budget_shortfall
  )
# Merge into one data frame
cleandata <- rbind(cleandata2019, cleandata2010)
```

Final Feed America Cleaning Method 2010-2020

```
# Clean Up Feed America 2019-2020
FeedData2019_2021 <- rawdata2019_2021%>%
  filter(State=="CA")

FeedData2019_2021 <- FeedData2019_2021[,c(3:5)]

FeedData2019_2021 <- janitor::clean_names(FeedData2019_2021)

FeedData2019_2021[,3] <-
  sapply(FeedData2019_2021[,3],function(x) as.numeric(gsub("%","",x)))

FeedData2019_2021 <-
  FeedData2019_2021[c("year","county_state","overall_food_insecurity_rate")]

# Clean Up Feed America 2010-2018
```

```r
cleanFeed <- function(data){

  dataCA <- data%>%
    filter(State=="CA")

  dataCA <- dataCA[,c(3,4)]

  dataCA[,2] <- sapply(dataCA[,2],function(x) as.numeric(gsub("%","",x)))

  colnames(dataCA) <- c("county_state","overall_food_insecurity_rate")

  return (dataCA)
}

FeedData2018 <-
  cbind("year" = rep(2018, nrow(cleanFeed(rawdata2018))), cleanFeed(rawdata2018))
FeedData2017 <-
  cbind("year" = rep(2017, nrow(cleanFeed(rawdata2017))), cleanFeed(rawdata2017))
FeedData2016 <-
  cbind("year" = rep(2016, nrow(cleanFeed(rawdata2016))), cleanFeed(rawdata2016))
FeedData2015 <-
  cbind("year" = rep(2015, nrow(cleanFeed(rawdata2015))), cleanFeed(rawdata2015))
FeedData2014 <-
  cbind("year" = rep(2014, nrow(cleanFeed(rawdata2014))), cleanFeed(rawdata2014))
FeedData2013 <-
  cbind("year" = rep(2013, nrow(cleanFeed(rawdata2013))), cleanFeed(rawdata2013))
FeedData2012 <-
  cbind("year" = rep(2012, nrow(cleanFeed(rawdata2012))), cleanFeed(rawdata2012))
FeedData2011 <-
  cbind("year" = rep(2011, nrow(cleanFeed(rawdata2011))), cleanFeed(rawdata2011))
FeedData2010 <-
  cbind("year" = rep(2010, nrow(cleanFeed(rawdata2010))), cleanFeed(rawdata2010))

# Clean Feed America Data Frame 2010-2020
FeedData <-
  rbind(FeedData2010,FeedData2011,FeedData2012,
        FeedData2013,FeedData2014,FeedData2015,
        FeedData2016,FeedData2017,FeedData2018,
        FeedData2019_2021)

countyNameDisable <- FeedData$county_state

FeedData <- FeedData%>%
  mutate(county_state = gsub(", California", "", county_state))%>%
  filter(year!=2021)%>%
  rename(county = county_state)%>%
  arrange(county)

countyName <- FeedData$county
```

Age Data Cleaning Method 2019

```
### Clean up Age Data for 2019
AgeData2019CA <- AgeData2019 %>%
  filter(NAME %in% c(countyNameDisable, 'Geographic Area Name')) %>%
  row_to_names(row_number = 1)

colnames(AgeData2019CA)[3] <- "Total Population"

Age2019 <- AgeData2019CA %>%
  .[, !grepl("Margin of Error", colnames(.))] %>%
  .[, !grepl("Annotation", colnames(.))] %>%
  .[, !grepl("Percent", colnames(.))] %>%
  .[, grepl(
    "Geographic Area Name|Total Population|Population under 18 years|
    Population 65 years and over",
    colnames(.)
  )] %>%
  .[, !grepl("years!!|over!!", colnames(.))]

Age2019$'Geographic Area Name' <-gsub(", California", "", Age2019$'Geographic Area Name')

colnames(Age2019)[1] <- "county"

colnames(Age2019) <- gsub(".*\\Estimate!!", "", colnames(Age2019))
colnames(Age2019) <- gsub("Total civilian noninstitutionalized population!!",
                                        "",colnames(Age2019))
colnames(Age2019) <-gsub("DISABILITY TYPE BY DETAILED AGE!!", "", colnames(Age2019))
colnames(Age2019) <- gsub("!!", ": ", colnames(Age2019))
colnames(Age2019) <- gsub(": :", ":", colnames(Age2019))

Age2019 <- Age2019 %>% mutate_at(-1, as.numeric)

TailAge2019 <- Age2019[,c(1:4)]
for (i in c(3:4)) {
  TailAge2019[, i] <- Age2019[, i] / Age2019[, 2]
}

colnames(TailAge2019) <-
  c("county","total_population","population_under_18","population_over_65")
```

Disability Data Cleaning Method 2010-2020

```
# Clean Up Disability 2013-2020
cleanDis1 <- function(data){

  dataCA <- data %>%
    filter(NAME %in% c(countyNameDisable, 'Geographic Area Name')) %>%
    row_to_names(row_number = 1)

  colnames(dataCA)[3] <- "Total Population"
```

```
  dataCA1 <- dataCA %>%
    .[,!grepl("Margin of Error", colnames(.))] %>%
    .[,!grepl("Annotation", colnames(.))] %>%
    .[, grepl("Geographic Area Name|Percent with a disability",colnames(.))] %>%
    .[,!grepl("population!",colnames(.))]

  dataCA1$'Geographic Area Name' <-
    gsub(", California", "", dataCA1$'Geographic Area Name')
  colnames(dataCA1)[1] <- "county"
  colnames(dataCA1)[2] <- "Percent with a disability"

  dataCA1 <- dataCA1 %>% mutate_at(-1, as.numeric)

  return (dataCA1)

}

Dis2020 <-
  cbind("year" = rep(2020, nrow(cleanDis1(DisData2020))), cleanDis1(DisData2020))
Dis2019 <-
  cbind("year" = rep(2019, nrow(cleanDis1(DisData2019))), cleanDis1(DisData2019))
Dis2018 <-
  cbind("year" = rep(2018, nrow(cleanDis1(DisData2018)[, c(1, 2)])),
        cleanDis1(DisData2018)[, c(1, 2)])
Dis2017 <-
  cbind("year" = rep(2017, nrow(cleanDis1(DisData2017)[, c(1, 2)])),
        cleanDis1(DisData2017)[, c(1, 2)])
Dis2016 <-
  cbind("year" = rep(2016, nrow(cleanDis1(DisData2016)[, c(1, 2)])),
        cleanDis1(DisData2016)[, c(1, 2)])
Dis2015 <-
  cbind("year" = rep(2015, nrow(cleanDis1(DisData2015)[, c(1, 2)])),
        cleanDis1(DisData2015)[, c(1, 2)])
Dis2014 <-
  cbind("year" = rep(2014, nrow(cleanDis1(DisData2014)[, c(1, 2)])),
        cleanDis1(DisData2014)[, c(1, 2)])
Dis2013 <-
  cbind("year" = rep(2013, nrow(cleanDis1(DisData2013)[, c(1, 2)])),
        cleanDis1(DisData2013)[, c(1, 2)])

# Clean Up Disability 2010-2012
cleanDis2 <- function(data){

  dataCA <- data %>%
    filter(NAME %in% c(countyNameDisable, 'Geographic Area Name')) %>%
    row_to_names(row_number = 1)

  colnames(dataCA)[3] <- "Total Population"

  dataCA1 <- dataCA %>%
    .[,!grepl("Margin of Error", colnames(.))] %>%
    .[,!grepl("Annotation", colnames(.))] %>%
    .[, grepl("Geographic Area Name|Percent with a disability",colnames(.))] %>%
```

```
    .[,!grepl("population!",colnames(.))]

  dataCA1$'Geographic Area Name' <-
    gsub(", California", "", dataCA1$'Geographic Area Name')
  colnames(dataCA1)[1] <- "county"
  colnames(dataCA1)[2] <- "Percent with a disability"

  dataCA1 <- (dataCA1[,c(1,2)] %>% mutate_at(-1, as.numeric))

  return (dataCA1)
}

Dis2012 <- cbind("year"=rep(2012,nrow(cleanDis2(DisData2012))),cleanDis2(DisData2012))
Dis2011 <- cbind("year"=rep(2011,nrow(cleanDis2(DisData2011))),cleanDis2(DisData2011))
Dis2010 <- cbind("year"=rep(2010,nrow(cleanDis2(DisData2010))),cleanDis2(DisData2010))

# Total Disability 2010-2021
totalDisability <-
  rbind(Dis2010,Dis2011,Dis2012,Dis2013,
        Dis2014,Dis2015,Dis2016,Dis2017,
        Dis2018,Dis2019,Dis2020)
totalDisability <- totalDisability %>% arrange(county)
colnames(totalDisability) <- c("year","county","percent_disabled")
```

Unemployment Data Cleaning Method 2010-2020

```
### Clean Up Unemployment 2010-2020
UnemploymentData <- unemploymentdataraw%>%
  filter(area_type=="County", status_preliminary_final=="Final")%>%
  filter(year>=2010 & year<2021)%>%
  filter(!area_name %in% c("Non Residential County",
                           "Resident Out of State County",
                           "Unallocated County",
                           "Resident Out of State County",
                           "Nonresident20 County",
                           "Resident Out-of-State19 County",
                           "Resident Out-of-State County"))%>%
  group_by(year, area_name)%>%
  summarise("unemployment_rate_avg"=mean(unemployment_rate))%>%
  distinct(.)%>%
  ungroup() %>%
  rename("county"="area_name")
```

Income Data Cleaning Method 2010-2020

```
IncomeData <- incomedataraw%>%
  filter(taxable_year >= 2010 & taxable_year <= 2021)%>%
  filter(
    !county %in% c(
```

```
      "Nonresident",
      "Resident Out of State County",
      "Unallocated",
      "Resident Out of State County",
      "Nonresident20 County",
      "Resident Out-of-State19 County",
      "Resident Out-of-State County"
    )
  ) %>%
  rename("year"="taxable_year")%>%
  mutate("county"=paste(.$county, "County"))%>%
  arrange(year,county)

IncomeData <- IncomeData[,c(1,2,6)]
```

Final Data Frame with all Factors

```
# Final Data Frame
majorDF <- FeedData %>% full_join(totalDisability)
majorDF <- majorDF %>% full_join(UnemploymentData)
majorDF <- majorDF %>% full_join(IncomeData)

majorDF <- majorDF %>%
  filter(!county %in%
           c(
             "Resident Out of State County",
             "Nonresident20 County",
             "Resident Out-of-State19 County",
             "Resident Out-of-State County"
           ))

majorDF2019 <- majorDF %>% filter(year==2019)
majorDF2019 <- TailAge2019 %>% inner_join(majorDF2019)
```

All Correlation Matrix Methodologies

```
# Age Demographics 2019 GGpair
ggpairs(majorDF2019[,c(6,3,4)],
        columnLabels=c("Food Insecurity Rate (%)",
                       "Population <18 Rate (%)",
                       "Population >65 Rate (%)"),
        aes(fill="Food Insecurity",col="Food Insecurity"),
        title = "Scatterplot Matrix of Age Demographics, 2019") +
  scale_fill_manual(values=c("#4E5745"))+
  scale_color_manual(values=c("#4E5745"))+
  theme(strip.background = element_rect(fill = "#AC5B4C"),
        panel.background=element_rect(fill="#E5DACF",color="#E5DACF",
                                      size=0.5,linetype="solid"),
        panel.grid.minor = element_blank())
```

```
# Disability Demographics 2019 GGpair
ggpairs(majorDF2019[,c(6,7)],
        columnLabels=c("Food Insecurity Rate (%)","Percent Disabled Rate (%)"),
        aes(fill="Food Insecurity",col="Food Insecurity"),
        title = "Scatterplot Matrix of Disability Demographics, 2019") +
  scale_fill_manual(values=c("#4E5745"))+
  scale_color_manual(values=c("#4E5745"))+
  theme(strip.background = element_rect(fill = "#AC5B4C"),
        panel.background=element_rect(fill="#E5DACF",color="#E5DACF",
                                      size=0.5,linetype="solid"),
        panel.grid.minor = element_blank())

# Income and Unemployment Demographics 2019 GGpair
ggpairs(majorDF2019[,c(6,9,8)],
        columnLabels=c("Food Insecurity Rate (%)",
                       "Median Income","Unemployment Rate (%)"),
        aes(fill="Food Insecurity",col="Food Insecurity"),
        title = "Scatterplot Matrix of Financial Demographics, 2019") +
  scale_fill_manual(values=c("#4E5745"))+
  scale_color_manual(values=c("#4E5745"))+
  theme(strip.background = element_rect(fill = "#AC5B4C"),
        panel.background=element_rect(fill="#E5DACF",color="#E5DACF",
                                      size=0.5,linetype="solid"),
        panel.grid.minor = element_blank())
```

All Boxplot Methodologies

```
# Food Insecurity Boxplot
majorDF[,c(1,3)] %>%
  mutate(year=as.factor(year))%>%
  ggplot(aes(x=year, y=overall_food_insecurity_rate, fill=(year==c(2020)))) +
  geom_boxplot(show.legend = F)+
  scale_fill_manual(values=c("#4E5745","#AC5B4C"))+
  theme_minimal()+
  theme(panel.background=element_rect(fill="#E5DACF",color="#E5DACF",
                                      size=0.5,linetype="solid"))+
  labs(title="Overall Food Insecurity Rate per Year",x="Year",
       y="Food Insecurity Rate (%)")

# Disability Boxplot
majorDF[,c(1,4)] %>%
  mutate(year=as.factor(year))%>%
  ggplot(aes(x=year, y=percent_disabled, fill=(year==2020))) +
  geom_boxplot(show.legend = F)+
  scale_fill_manual(values=c("#4E5745","#AC5B4C"))+
  theme_minimal()+
  theme(panel.background=element_rect(fill="#E5DACF",color="#E5DACF",
                                      size=0.5,linetype="solid"))+
  labs(title="Overall Disability Rate per Year",x="Year",
       y="Disability Rate (%)")

# Unemployment Boxplot
```

```
majorDF[,c(1,5)] %>%
  mutate(year=as.factor(year))%>%
  ggplot(aes(x=year, y=unemployment_rate_avg, fill=(year==2020))) +
  geom_boxplot(show.legend = F)+
  scale_fill_manual(values=c("#4E5745","#AC5B4C"))+
  theme_minimal()+
  theme(panel.background=element_rect(fill="#E5DACF",color="#E5DACF",
                                      size=0.5,linetype="solid"))+
  labs(title="Overall Unemployment Rate per Year",x="Year",
       y="Unemployment Rate (%)")

# Income Boxplot
majorDF[,c(1,6)]%>%
  mutate(year=as.factor(year))%>%
  ggplot(aes(x=year, y=median_income, fill=(year==2020))) +
  geom_boxplot(show.legend = F)+
  scale_fill_manual(values=c("#4E5745","#AC5B4C"))+
  theme_minimal()+
  theme(panel.background=element_rect(fill="#E5DACF",color="#E5DACF",
                                      size=0.5,linetype="solid"))+
  labs(title="Overall Median Income per Year",x="Year",
       y="Median Income ($)")
```

ShapeFile Map Methodologies

```
shape <- sf::read_sf(dsn = "CA_Counties_ShapeFile",
                     layer = "CA_Counties_TIGER2016")
counties <- shape['NAME'] %>% arrange(NAME)
colnames(counties)[1] <- "county"

majorDF2019$county <- gsub(" County","",majorDF2019$county)
counties <- counties %>% full_join(majorDF2019)

mapview(
  counties,
  zcol = "overall_food_insecurity_rate",
  layer.name = "Food Insecurity Rate (%)",
  map.types = "CartoDB.Positron",
  na.color = "#AC5B4C",
  col.regions = brewer.pal(100, "Blues"),
  alpha.regions = 1
) +
  mapview(
    counties,
    zcol = "unemployment_rate_avg",
    layer.name = "Unemployment Rate (%)",
    map.types = "CartoDB.Positron",
    na.color = "#AC5B4C",
    col.regions = brewer.pal(100, "Blues"),
    alpha.regions = 1
  ) +
```

```
  mapview(
    counties,
    zcol = "median_income",
    layer.name = "Median Income",
    map.types = "CartoDB.Positron",
    na.color = "#AC5B4C",
    col.regions = brewer.pal(100, "Blues"),
    alpha.regions = 1
  ) +
  mapview(
    counties,
    zcol = "percent_disabled",
    layer.name = "Disability Rate (%)",
    map.types = "CartoDB.Positron",
    na.color = "#AC5B4C",
    col.regions = brewer.pal(100, "Blues"),
    alpha.regions = 1
  )
```

All Linear Model Methodologies

```
# Food Insecurity vs. Disability Linear Model
majorDF%>%
  ggplot(aes(x=percent_disabled, y=overall_food_insecurity_rate))+
  geom_point(color="#4E5745")+
  geom_smooth(method="lm", color="#AC5B4C", show.legend = F)+
  theme_minimal()+
  theme(panel.background =
          element_rect(fill="#E5DACF", color = "#E5DACF",
                       size = 0.5, linetype = "solid"))+
  labs(title="Relationship Between FI and Disability Rate",
       subtitle="lm(food_insecurity_rate ~ disability_rate)",
       x="Disability Rate (%)",
       y="Food Insecurity Rate (%)")

# Food Insecurity vs. Median Income Linear Model
majorDF%>%
  ggplot(aes(x=median_income, y=overall_food_insecurity_rate))+
  geom_point(color="#4E5745")+
  geom_smooth(method="lm", color="#AC5B4C", show.legend = F)+
  theme_minimal()+
  theme(panel.background =
          element_rect(fill="#E5DACF", color = "#E5DACF",
                       size = 0.5, linetype = "solid"))+
  labs(title="Relationship Between FI and Median Income",
       subtitle="lm(food_insecurity_rate ~ median_income)",
       x="Median Income ($)",
       y="Food Insecurity Rate (%)")

# Food Insecurity vs. Unemployment Linear Model
majorDF%>%
  ggplot(aes(x=unemployment_rate_avg, y=overall_food_insecurity_rate))+
```

```
  geom_point(color="#4E5745")+
  geom_smooth(method="lm", color="#AC5B4C", show.legend = F)+
  theme_minimal()+
  theme(panel.background =
          element_rect(fill="#E5DACF", color = "#E5DACF",
                       size = 0.5, linetype = "solid"))+
  labs(title="Relationship Between FI and Unemployment Rate",
       subtitle="lm(food_insecurity_rate ~ unemployment_rate_avg)",
       x="Unemployment Rate (%)",
       y="Food Insecurity Rate (%)")
```

AIC for a Predictive Linear Regression Model

```
none_mod <-
  lm(overall_food_insecurity_rate~1,data=na.omit(majorDF)) ##model with only intercept
full_mod <-
  lm(overall_food_insecurity_rate~unemployment_rate_avg+median_income+percent_disabled,
     data=na.omit(majorDF))

stepAIC(full_mod, scope=list(upper=full_mod, lower = ~1),
        direction="backward", k=2, trace = FALSE)
```

Overall Predictive Models against 2019 Data

```
# [Overall: 2010-2018 against 2019] Linear Model:
#Food Insecurity ~ Unemployment Rate + Disability Rate + Median Income
majorDFtrain <- na.omit(majorDF)%>%
  filter(year<2019)

majorDFtest <- na.omit(majorDF)%>%
  filter(year==2019)

model <-
  lm(
    overall_food_insecurity_rate ~
      unemployment_rate_avg + percent_disabled + median_income,
    data = majorDFtrain
  )

lm_results <- data.frame(
  "county" = majorDFtest$county,
  "predicted_food_insecurity_rate" =
    as.numeric(model$coefficients[1]) + as.numeric(model$coefficients[2]) *
    majorDFtest$unemployment_rate_avg + as.numeric(model$coefficients[3]) *
    majorDFtest$percent_disabled +
    as.numeric(model$coefficients[4]) * majorDFtest$median_income,
  "actual_food_insecurity_rate" = majorDFtest$overall_food_insecurity_rate
)

lm_overall_acc19 <-
```

```
  1 - mean((lm_results$predicted_food_insecurity_rate-
              lm_results$actual_food_insecurity_rate)/
             lm_results$actual_food_insecurity_rate)

# [Overall: 2010-2018 against 2019] Random Forest Model:
#Food Insecurity ~ Unemployment Rate + Disability Rate + Median Income
train_rf <- train(overall_food_insecurity_rate ~ unemployment_rate_avg + median_income +
                    percent_disabled,
                  method="rf", data=majorDFtrain,
                  tuneGrid=data.frame(mtry=1:2),
                  trControl=trainControl(method="cv", number=5))

pred <- as.numeric(predict(train_rf, newdata=majorDFtest))

rf_results <-
  data.frame("county"=majorDFtest$county,
                          "predicted_food_insecurity_rate"= pred,
                          "actual_food_insecurity_rate"=
               majorDFtest$overall_food_insecurity_rate)

rf_overall_acc19 <-
  1 - mean((rf_results$predicted_food_insecurity_rate-
              rf_results$actual_food_insecurity_rate)/
             rf_results$actual_food_insecurity_rate)
```

Overall Predictive Models against 2020 Data

```
# [Overall: 2010-2019 against 2020] Linear Model:
#Food Insecurity ~ Unemployment Rate + Disability Rate + Median Income
majorDFtrain <- na.omit(majorDF)%>%
  filter(year<2020)

majorDFtest <- na.omit(majorDF)%>%
  filter(year==2020)

model <-
  lm(
    overall_food_insecurity_rate ~
      unemployment_rate_avg + percent_disabled + median_income,
    data = majorDFtrain
  )

lm_results <- data.frame(
  "county" = majorDFtest$county,
  "predicted_food_insecurity_rate" = as.numeric(model$coefficients[1]) +
    as.numeric(model$coefficients[2]) *
    majorDFtest$unemployment_rate_avg + as.numeric(model$coefficients[3]) *
    majorDFtest$percent_disabled +
    as.numeric(model$coefficients[4]) * majorDFtest$median_income,
  "actual_food_insecurity_rate" = majorDFtest$overall_food_insecurity_rate
)
```

```
lm_overall_acc20 <-
  1 - mean((
    abs(
      lm_results$predicted_food_insecurity_rate -
        lm_results$actual_food_insecurity_rate
    )
  ) / lm_results$actual_food_insecurity_rate)

# [Overall: 2010-2019 against 2020] Random Forest Model:
#Food Insecurity ~ Unemployment Rate + Disability Rate + Median Income
train_rf <-
  train(
    overall_food_insecurity_rate ~
      unemployment_rate_avg + median_income + percent_disabled,
    method = "rf",
    data = majorDFtrain,
    tuneGrid = data.frame(mtry = 1:2),
    trControl = trainControl(method = "cv", number = 5)
  )

pred <- as.numeric(predict(train_rf, newdata=majorDFtest))

rf_results <- data.frame(
  "county" = majorDFtest$county,
  "predicted_food_insecurity_rate" = pred,
  "actual_food_insecurity_rate" = majorDFtest$overall_food_insecurity_rate
)

rf_overall_acc20 <-
  1 - mean((
    rf_results$predicted_food_insecurity_rate - rf_results$actual_food_insecurity_rate
  ) / rf_results$actual_food_insecurity_rate
  )
```

By County Linear Predictive Models against 2019, 2020 Data

```
# [By County: 2010-2018 against 2019 and 2010-2019 against 2020] Linear Model:
#Food Insecurity ~ Unemployment Rate + Disability Rate + Median Income
lm_trainr <- function(c,y) {
  # Define Training Data (2010-2019) and Test Data (2020)
  majorDFtrain <- na.omit(majorDF) %>%
    filter(year < y)%>%
    filter(county==c)

  majorDFtest <- na.omit(majorDF) %>%
    filter(year == y)%>%
    filter(county==c)

  if (nrow(majorDFtrain) == 0 | nrow(majorDFtest) == 0){

    rf_results <- data.frame("county"=c,
```

```
            "predicted_food_insecurity_rate"=NA,
            "actual_food_insecurity_rate"=NA)

    rf_acc <- NA

    return(list(results=rf_results, accuracy=rf_acc))
  }

  model <-
    lm(
      overall_food_insecurity_rate ~ unemployment_rate_avg +
        percent_disabled + median_income,
      data = majorDFtrain
    )

  lm_results <- data.frame(
    "county" = majorDFtrain$county,
    "predicted_food_insecurity_rate" =
      as.numeric(model$coefficients[1]) + as.numeric(model$coefficients[2]) *
      majorDFtest$unemployment_rate_avg + as.numeric(model$coefficients[3]) *
      majorDFtest$percent_disabled +
      as.numeric(model$coefficients[4]) * majorDFtest$median_income,
    "actual_food_insecurity_rate" = majorDFtest$overall_food_insecurity_rate
  )

  lm_acc <-
    1 - mean(
      abs(
        lm_results$predicted_food_insecurity_rate -
          lm_results$actual_food_insecurity_rate
      ) / lm_results$actual_food_insecurity_rate
    )

  return(list(results = lm_results, accuracy = lm_acc))
}

results20 <-
  data.frame(county=c(), predicted_food_insecurity_rate=c(), actual_food_insecurity=c())
accuracy20 <- c()

for (c in unique(majorDF$county)){
  trainr_data <- lm_trainr(c,2020)
  #results20 <- rbind(results20, trainr_data[[1]])
  accuracy20 <- c(accuracy20, trainr_data[[2]])
}

#results20$accuracy20 <- accuracy20
lm_county_acc20 <- mean(accuracy20, na.rm=T)

results19 <- data.frame(county=c(),
                        predicted_food_insecurity_rate=c(), actual_food_insecurity=c())
accuracy19 <- c()
```

```r
for (c in unique(majorDF$county)){
  trainr_data <- lm_trainr(c,2019)
  results19 <- rbind(results19, trainr_data[[1]])
  accuracy19 <- c(accuracy19, trainr_data[[2]])
}

#results19$accuracy19 <- accuracy19
lm_county_acc19 <- mean(accuracy19, na.rm=T)
```

By County Random Forest Predictive Models against 2019, 2020 Data

```r
# [By County: 2010-2018 against 2019 and 2010-2019 against 2020]
#Random Forest Model: Food Insecurity ~
#Unemployment Rate + Disability Rate + Median Income
rf_trainr <- function(c,y) {
  # Define Training Data (2010-2019) and Test Data (2020)
  majorDFtrain <- na.omit(majorDF) %>%
    filter(year < y)%>%
    filter(county==c)

  majorDFtest <- na.omit(majorDF) %>%
    filter(year == y)%>%
    filter(county==c)

  if (nrow(majorDFtrain) <= 1 | nrow(majorDFtest) == 0){

    rf_results <- data.frame("county"=c,
           "predicted_food_insecurity_rate"=NA,
           "actual_food_insecurity_rate"=NA)

    rf_acc <- NA

    return(list(results=rf_results, accuracy=rf_acc))
  }

  train_rf <-
    train(
      overall_food_insecurity_rate ~
        unemployment_rate_avg + percent_disabled + median_income,
      method = "rf",
      data = majorDFtrain,
      tuneGrid = data.frame(mtry = 1:2),
      trControl = trainControl(method = "cv", number = 5)
    )

  pred <- as.numeric(predict(train_rf, newdata=majorDFtest))

  rf_results <-
    data.frame("county"=majorDFtest$county,
               "predicted_food_insecurity_rate" = pred,
               "actual_food_insecurity_rate" = majorDFtest$overall_food_insecurity_rate)
```

```
  rf_acc <-
    1 - mean((
      rf_results$predicted_food_insecurity_rate - rf_results$actual_food_insecurity_rate
    ) / rf_results$actual_food_insecurity_rate
    )

  return(list(results=rf_results, accuracy=rf_acc))
}

results20 <-
  data.frame(county=c(),
             predicted_food_insecurity_rate=c(), actual_food_insecurity=c())
accuracy20 <- c()

for (c in unique(majorDF$county)){
  trainr_data <- rf_trainr(c,2020)
  #results20 <- rbind(results20, trainr_data[[1]])
  accuracy20 <- c(accuracy20, trainr_data[[2]])
}

#results20$accuracy20 <- accuracy20
rf_county_acc20 <- mean(accuracy20, na.rm=T)

results19 <-
  data.frame(county=c(),
             predicted_food_insecurity_rate=c(), actual_food_insecurity=c())
accuracy19 <- c()

for (c in unique(majorDF$county)){
  trainr_data <- rf_trainr(c,2019)
  results19 <- rbind(results19, trainr_data[[1]])
  accuracy19 <- c(accuracy19, trainr_data[[2]])
}

#results19$accuracy19 <- accuracy19
rf_county_acc19 <- mean(accuracy19, na.rm=T)
```

Predictive Model Accuracy Table

```
kable(data.frame("Model Type"=c("Overall Linear Regression", "Overall Random Forest",
                                "By County Linear Model", "By County Random Forest"),
                 "Test Accuracy 2019"=c(lm_overall_acc19, rf_overall_acc19,
                                        lm_county_acc19, rf_county_acc19),
                "Test Accuracy 2020"=c(lm_overall_acc20, rf_overall_acc20,
                                       lm_county_acc20, rf_county_acc20)))
```