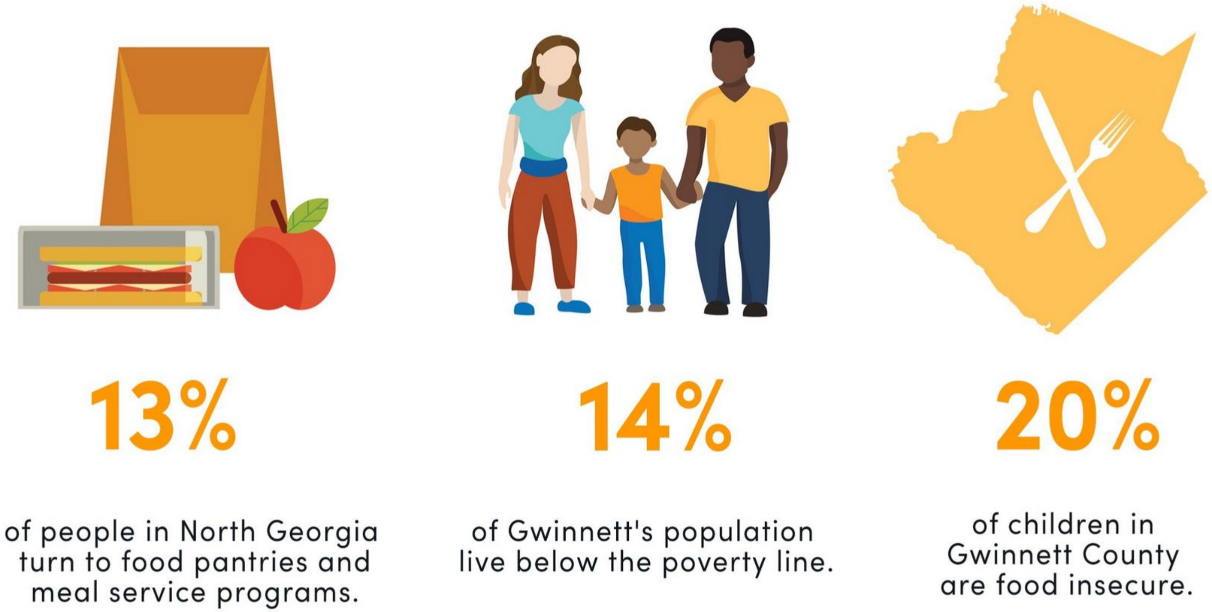
SPRINT 2 RETROSPECTIVE

ELIZABETH DOSS - DATA ANALYZER, CLIENT LIASON

CALEB CEDENO - PROJECT MANAGER, VISUALIZATIONS

DAMARIS MONTECINOS - DATA MODELER, PROJECT DOCUMENTER

SATISFEED



13%

of people in North Georgia turn to food pantries and meal service programs.

14%

of Gwinnett's population live below the poverty line.

20%

of children in Gwinnett County are food insecure.

- Gwinnett County based foodbank
- Weekly food distributions
- Fill in summer gaps for children

TECHNOLOGIES

🚀 **GOOGLE COLAB** - CLOUD-BASED JUPYTER NOTEBOOK ENVIRONMENT FOR CODING AND COLLABORATION

🐍 **PYTHON** - CORE PROGRAMMING LANGUAGE FOR DATA PROCESSING AND ANALYSIS.

🔗 **GITHUB** - VERSION CONTROL AND COLLABORATION FOR CODE MANAGEMENT.

📊 **POWER BI** - DATA VISUALIZATION AND DASHBOARD CREATION TOOL

TEAM PLAN

📅 Sprint 1 - Due: 02/05

- 📌 Collect Data 📥
  📌 Clean Data 🧹

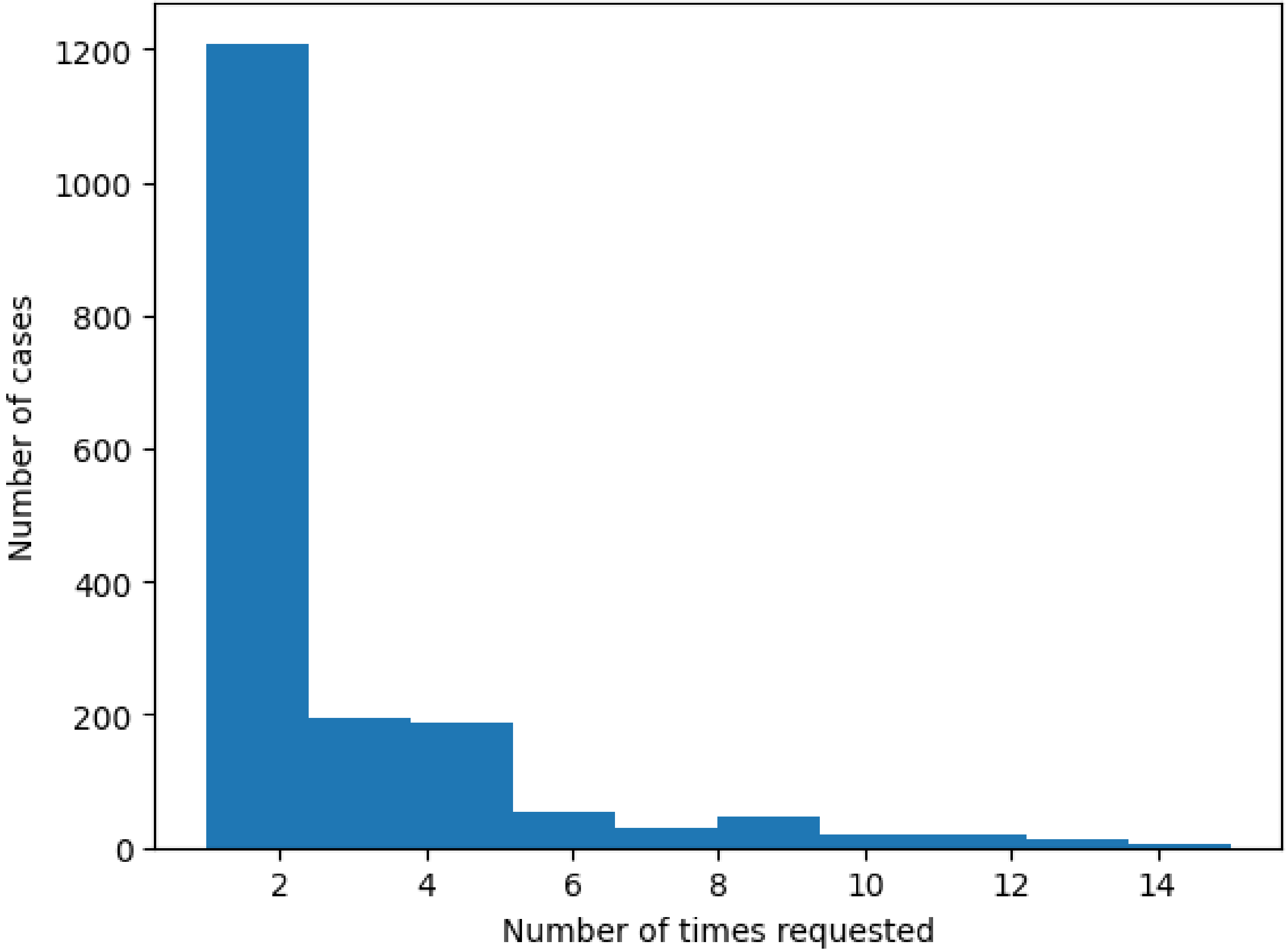
📅 Sprint 2 - Due: 03/24

- 📌 Analyze Data 📊

📅 Sprint 3 - Due: 04/24

- 📌 Visualize Data 📈
  📌 Create Dashboard 🖥️

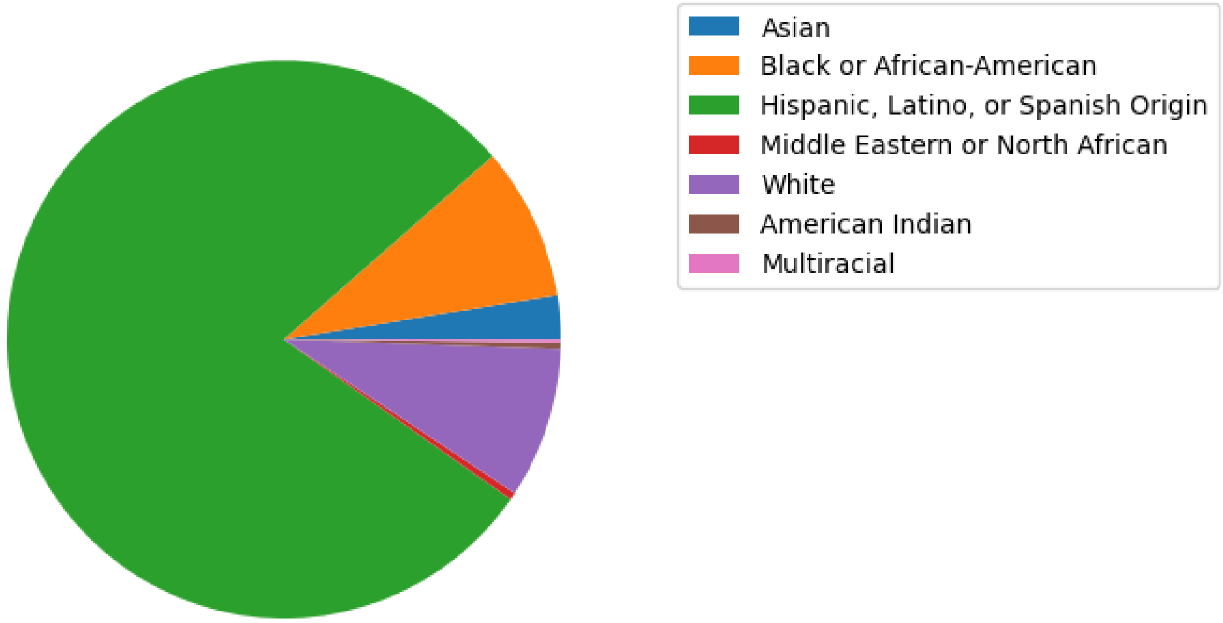
Elizabeth



HOW OFTEN DO PEOPLE REQUEST ASSISTANCE?

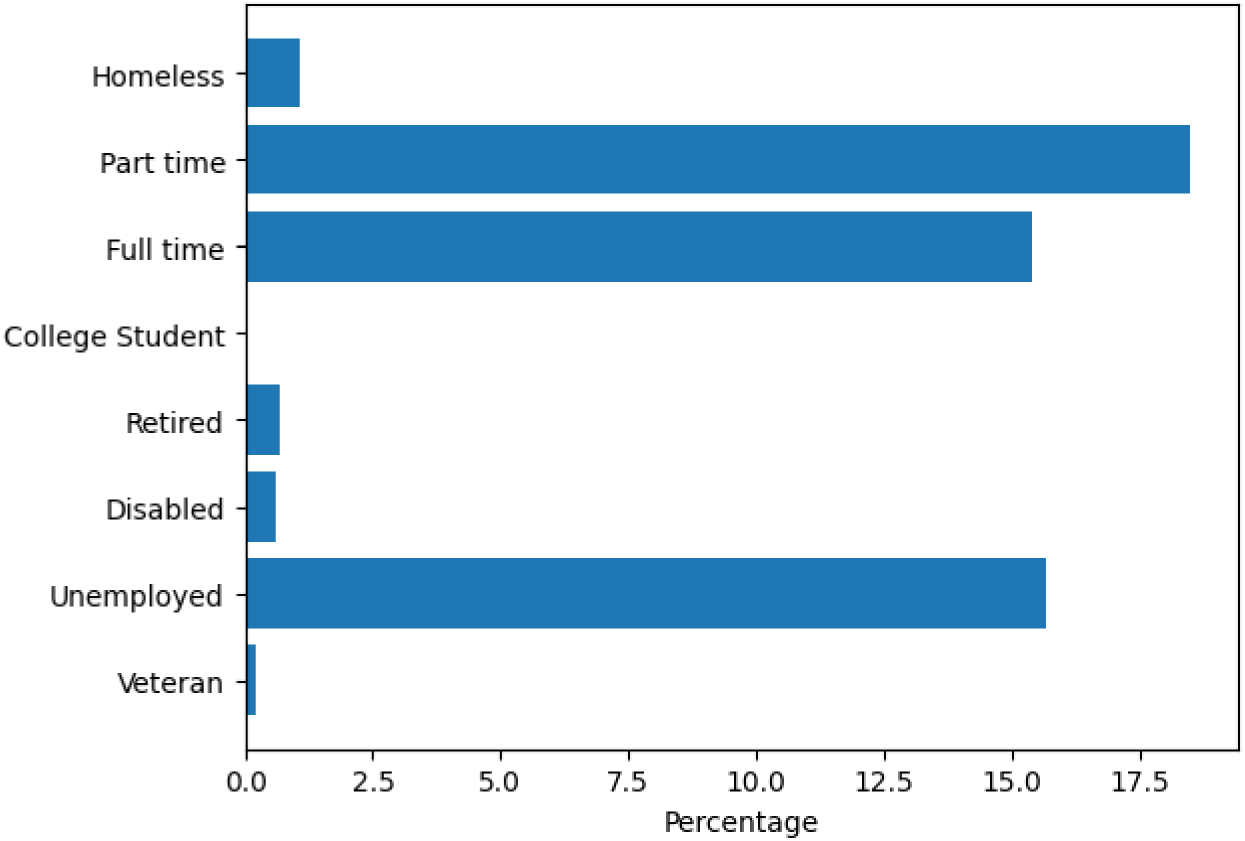
Elizabeth

DEMOGRAPHICS- ETHNIC/RACIAL IDENTITIES



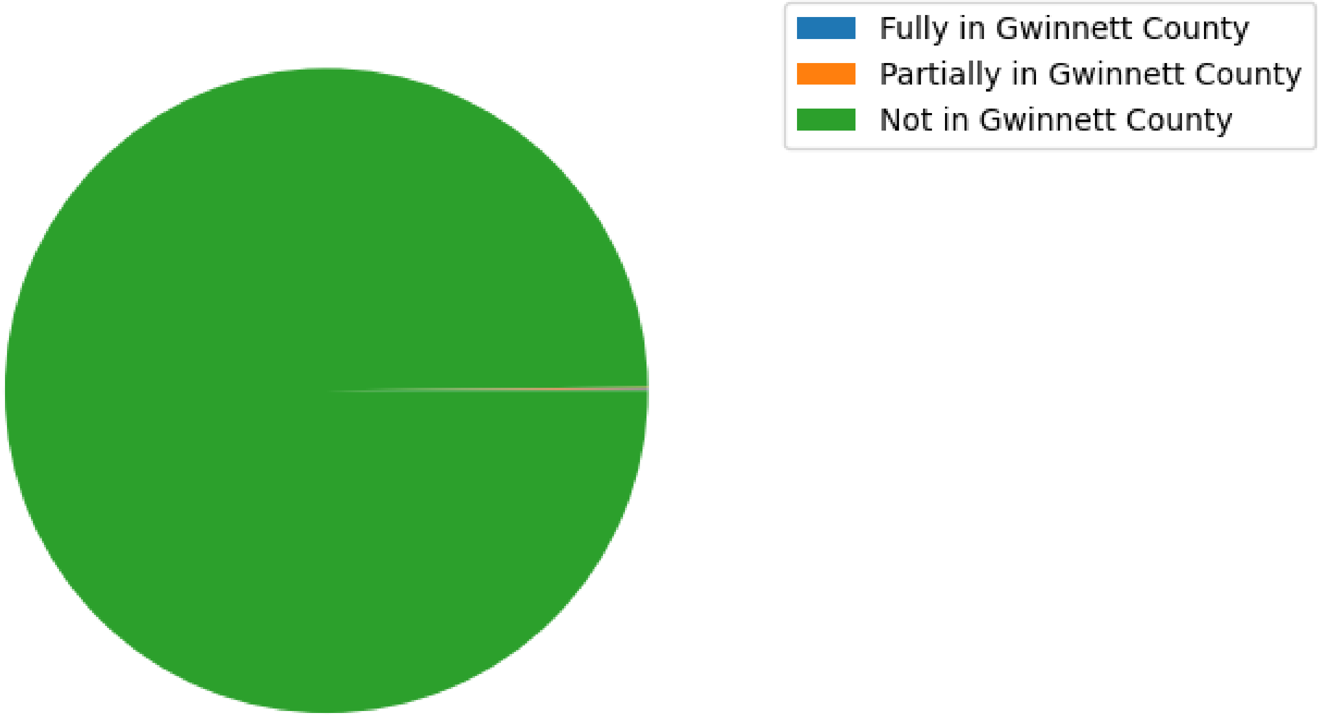
Elizabeth

DEMOGRAPHICS- CLIENT STATUS



Elizabeth

ARE SATISFEED CUSTOMERS GWINNETT RESIDENTS?



Elizabeth

PCA RESULTS

- Ran into problems because of column names and use of NaN

- Trying to rename columns failed and replace method failed; had to drop several columns

- Variance ultimately came from number of visits and household size; this is limited by lack of responses

Elizabeth

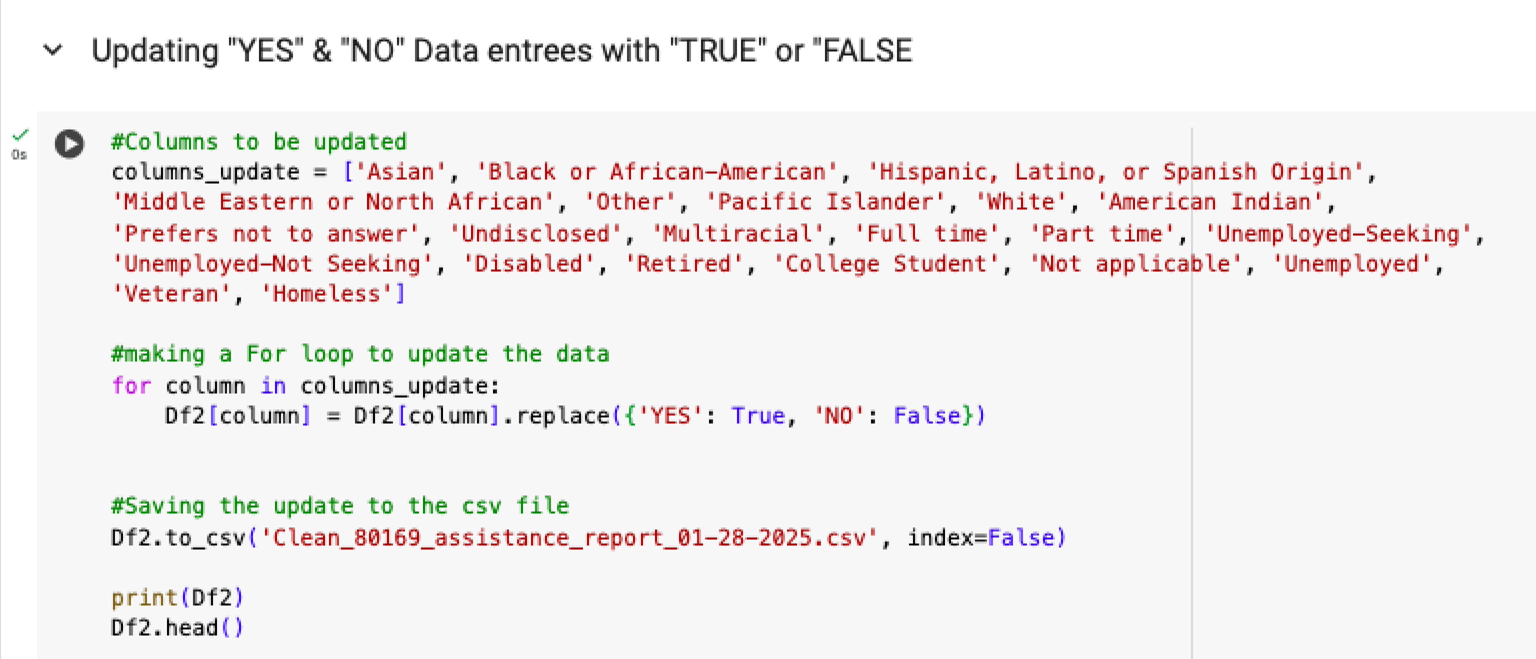
MULTIVARIATE LINEAR REGRESSION

- The original plan was to try and predict if customers would be repeats based on PCA components

- However, the PCA found only one relevant variable besides the number of customer visits

Damaris

UPDATING YES/NO COLUMNS TO TRUE/FALSE

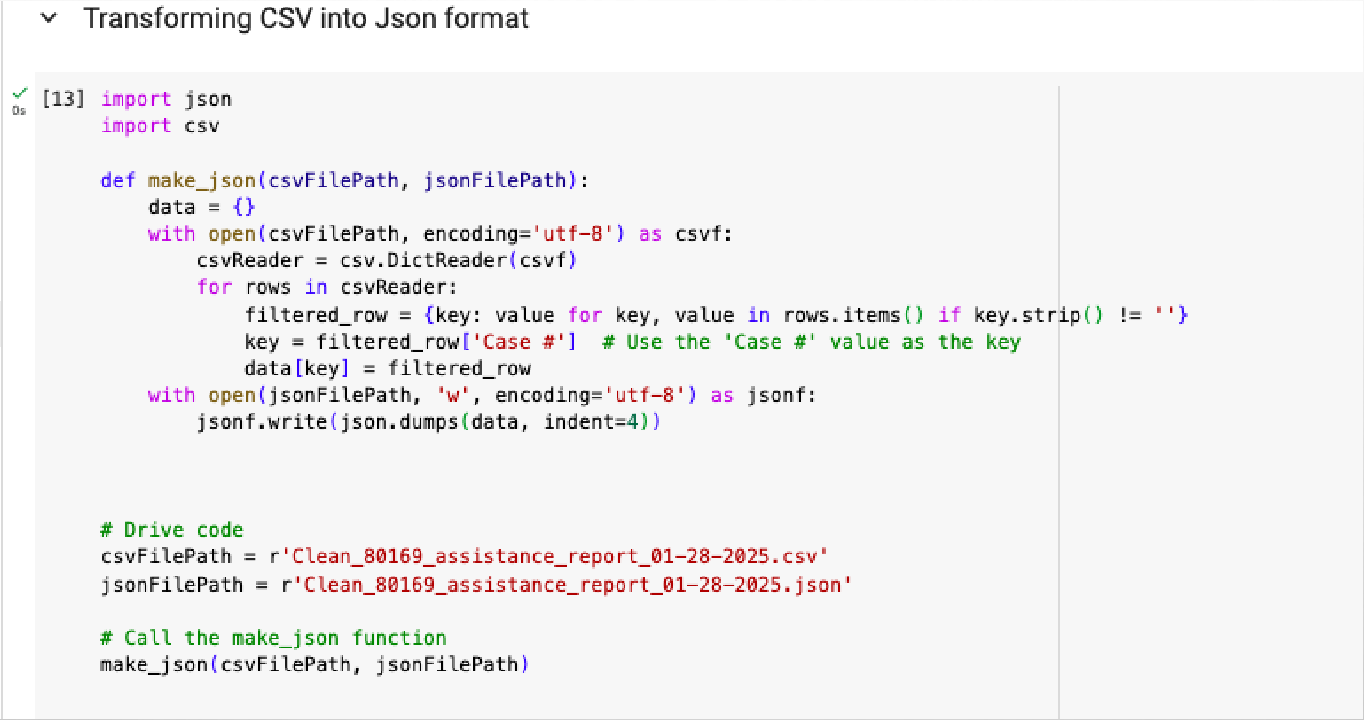


Updating "YES" & "NO" Data entrees with "TRUE" or "FALSE

```
#Columns to be updated
columns_update = ['Asian', 'Black or African-American', 'Hispanic, Latino, or Spanish Origin',
'Middle Eastern or North African', 'Other', 'Pacific Islander', 'White', 'American Indian',
'Prefers not to answer', 'Undisclosed', 'Multiracial', 'Full time', 'Part time', 'Unemployed-Seeking',
'Unemployed-Not Seeking', 'Disabled', 'Retired', 'College Student', 'Not applicable', 'Unemployed',
'Veteran', 'Homeless']

#making a For loop to update the data
for column in columns_update:
    Df2[column] = Df2[column].replace({'YES': True, 'NO': False})


#Saving the update to the csv file
Df2.to_csv('Clean_80169_assistance_report_01-28-2025.csv', index=False)

print(Df2)
Df2.head()
```

Damaris

TRANSFORMING CSV TO JSON FORMAT



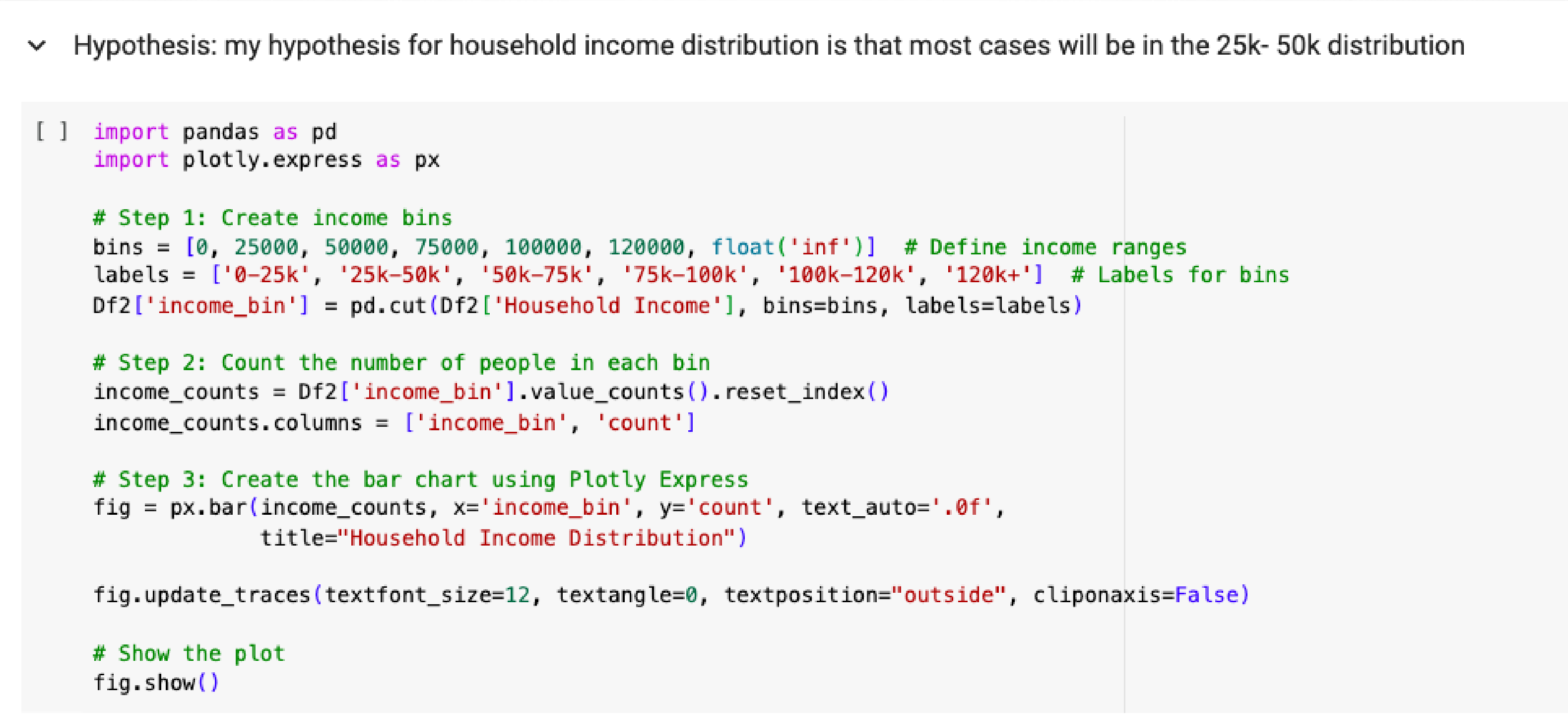
Transforming CSV into Json format

```python
import json
import csv

def make_json(csvFilePath, jsonFilePath):
    data = {}
    with open(csvFilePath, encoding='utf-8') as csvf:
        csvReader = csv.DictReader(csvf)
        for rows in csvReader:
            filtered_row = {key: value for key, value in rows.items() if key.strip() != ''}
            key = filtered_row['Case #']  # Use the 'Case #' value as the key
            data[key] = filtered_row
    with open(jsonFilePath, 'w', encoding='utf-8') as jsonf:
        jsonf.write(json.dumps(data, indent=4))



# Drive code
csvFilePath = r'Clean_80169_assistance_report_01-28-2025.csv'
jsonFilePath = r'Clean_80169_assistance_report_01-28-2025.json'

# Call the make_json function
make_json(csvFilePath, jsonFilePath)
```

Damaris

HOUSEHOLD INCOME DISTRIBUTION

- Hypothesis
- Code for graph



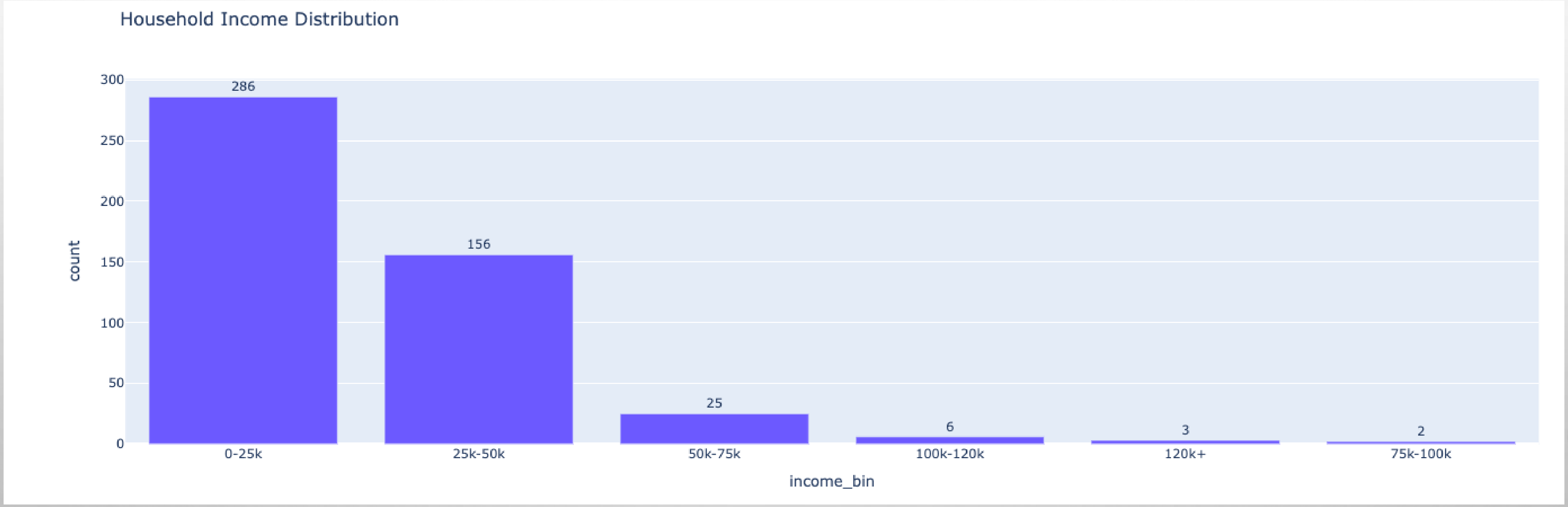
Hypothesis: my hypothesis for household income distribution is that most cases will be in the 25k- 50k distribution

```
[ ]  import pandas as pd
     import plotly.express as px

     # Step 1: Create income bins
     bins = [0, 25000, 50000, 75000, 100000, 120000, float('inf')]  # Define income ranges
     labels = ['0-25k', '25k-50k', '50k-75k', '75k-100k', '100k-120k', '120k+']  # Labels for bins
     Df2['income_bin'] = pd.cut(Df2['Household Income'], bins=bins, labels=labels)

     # Step 2: Count the number of people in each bin
     income_counts = Df2['income_bin'].value_counts().reset_index()
     income_counts.columns = ['income_bin', 'count']

     # Step 3: Create the bar chart using Plotly Express
     fig = px.bar(income_counts, x='income_bin', y='count', text_auto='.0f',
                  title="Household Income Distribution")

     fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=False)

     # Show the plot
     fig.show()
```

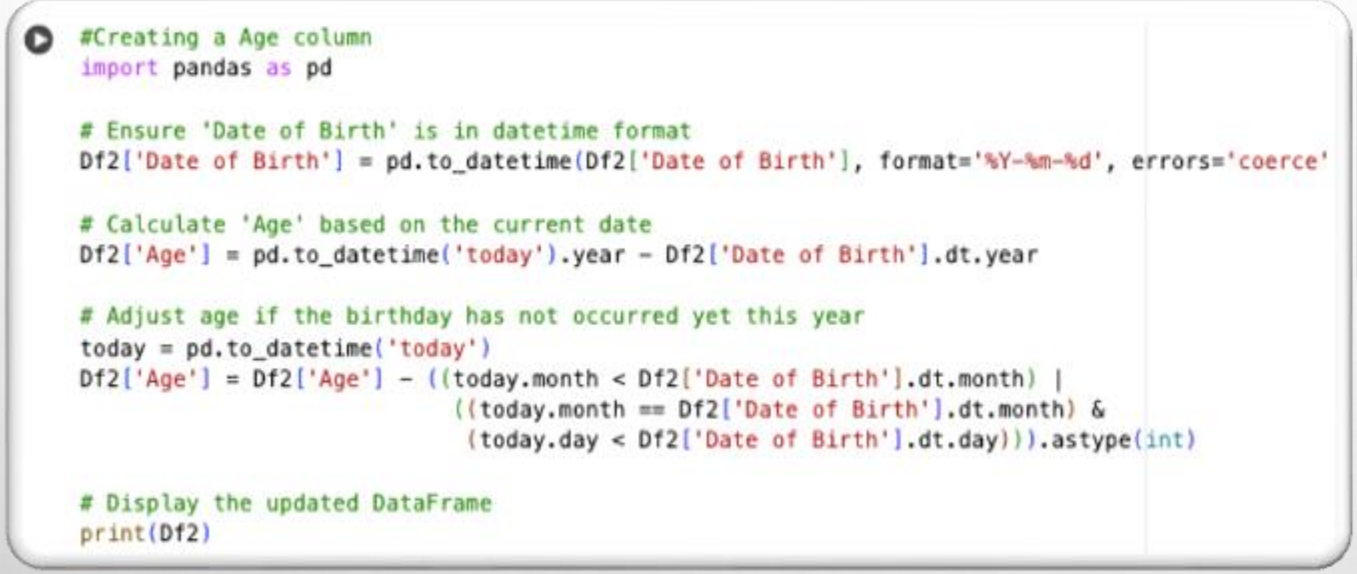
Damaris

HOUSEHOLD INCOME DISTRIBUTION

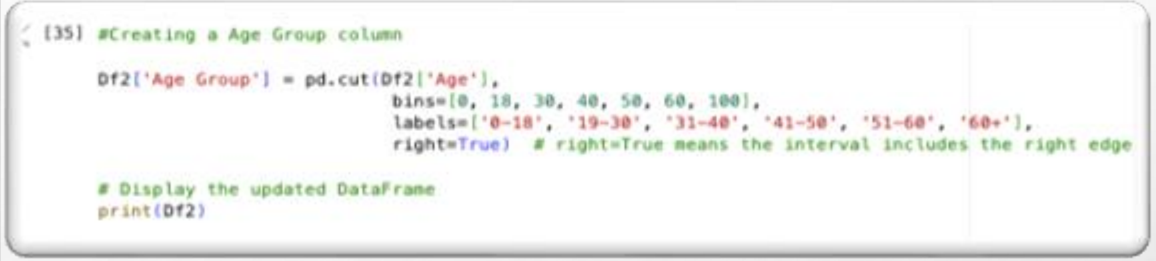
- My hypothesis is incorrect
- The top household income is 0-25k



Damaris



```python
#Creating a Age column
import pandas as pd

# Ensure 'Date of Birth' is in datetime format
Df2['Date of Birth'] = pd.to_datetime(Df2['Date of Birth'], format='%Y-%m-%d', errors='coerce'

# Calculate 'Age' based on the current date
Df2['Age'] = pd.to_datetime('today').year - Df2['Date of Birth'].dt.year

# Adjust age if the birthday has not occurred yet this year
today = pd.to_datetime('today')
Df2['Age'] = Df2['Age'] - ((today.month < Df2['Date of Birth'].dt.month) |
                            ((today.month == Df2['Date of Birth'].dt.month) &
                             (today.day < Df2['Date of Birth'].dt.day))).astype(int)

# Display the updated DataFrame
print(Df2)
```



```python
[35] #Creating a Age Group column

     Df2['Age Group'] = pd.cut(Df2['Age'],
                               bins=[0, 18, 30, 40, 50, 60, 100],
                               labels=['0-18', '19-30', '31-40', '41-50', '51-60', '60+'],
                               right=True)  # right=True means the interval includes the right edge

     # Display the updated DataFrame
     print(Df2)
```

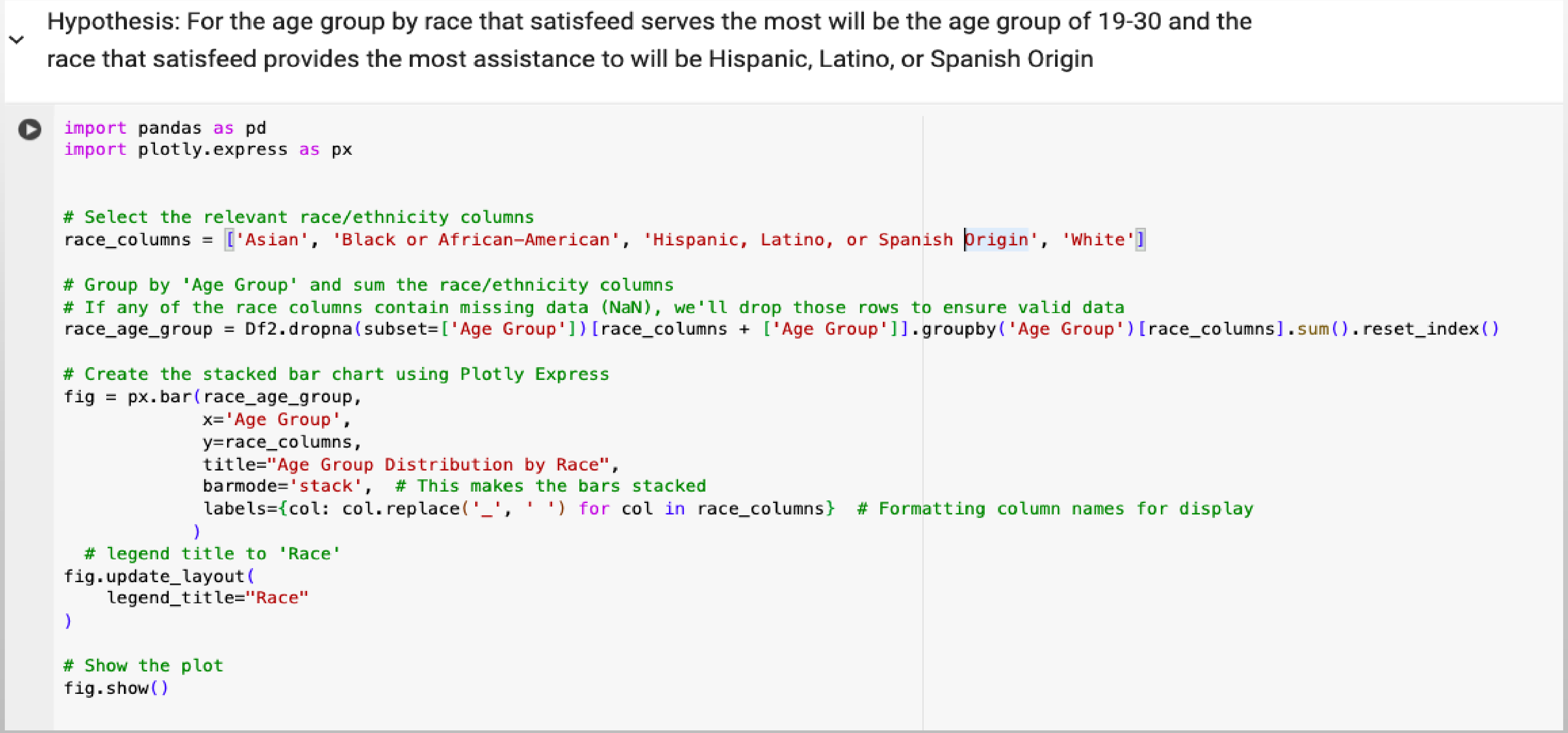
AGE GROUP DISTRIBUTION BY RACE

- AGE COLUMN AND AGE GROUP COLUMN CODE

Damaris

AGE GROUP DISTRIBUTION BY RACE

- HYPOTHESIS
- AGE GROUP DISTRIBUTION BY RACE CODE



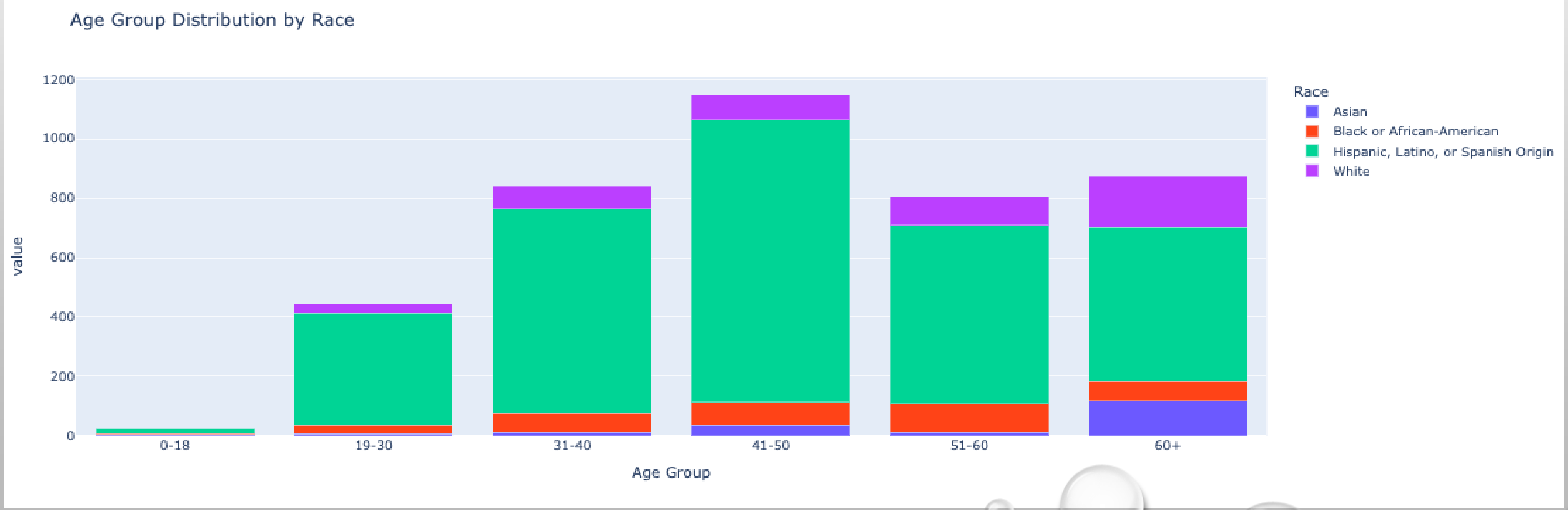
Hypothesis: For the age group by race that satisfeed serves the most will be the age group of 19-30 and the race that satisfeed provides the most assistance to will be Hispanic, Latino, or Spanish Origin

```
import pandas as pd
import plotly.express as px


# Select the relevant race/ethnicity columns
race_columns = ['Asian', 'Black or African-American', 'Hispanic, Latino, or Spanish Origin', 'White']

# Group by 'Age Group' and sum the race/ethnicity columns
# If any of the race columns contain missing data (NaN), we'll drop those rows to ensure valid data
race_age_group = Df2.dropna(subset=['Age Group'])[race_columns + ['Age Group']].groupby('Age Group')[race_columns].sum().reset_index()

# Create the stacked bar chart using Plotly Express
fig = px.bar(race_age_group,
             x='Age Group',
             y=race_columns,
             title="Age Group Distribution by Race",
             barmode='stack',  # This makes the bars stacked
             labels={col: col.replace('_', ' ') for col in race_columns}  # Formatting column names for display
            )
  # legend title to 'Race'
fig.update_layout(
    legend_title="Race"
)

# Show the plot
fig.show()
```

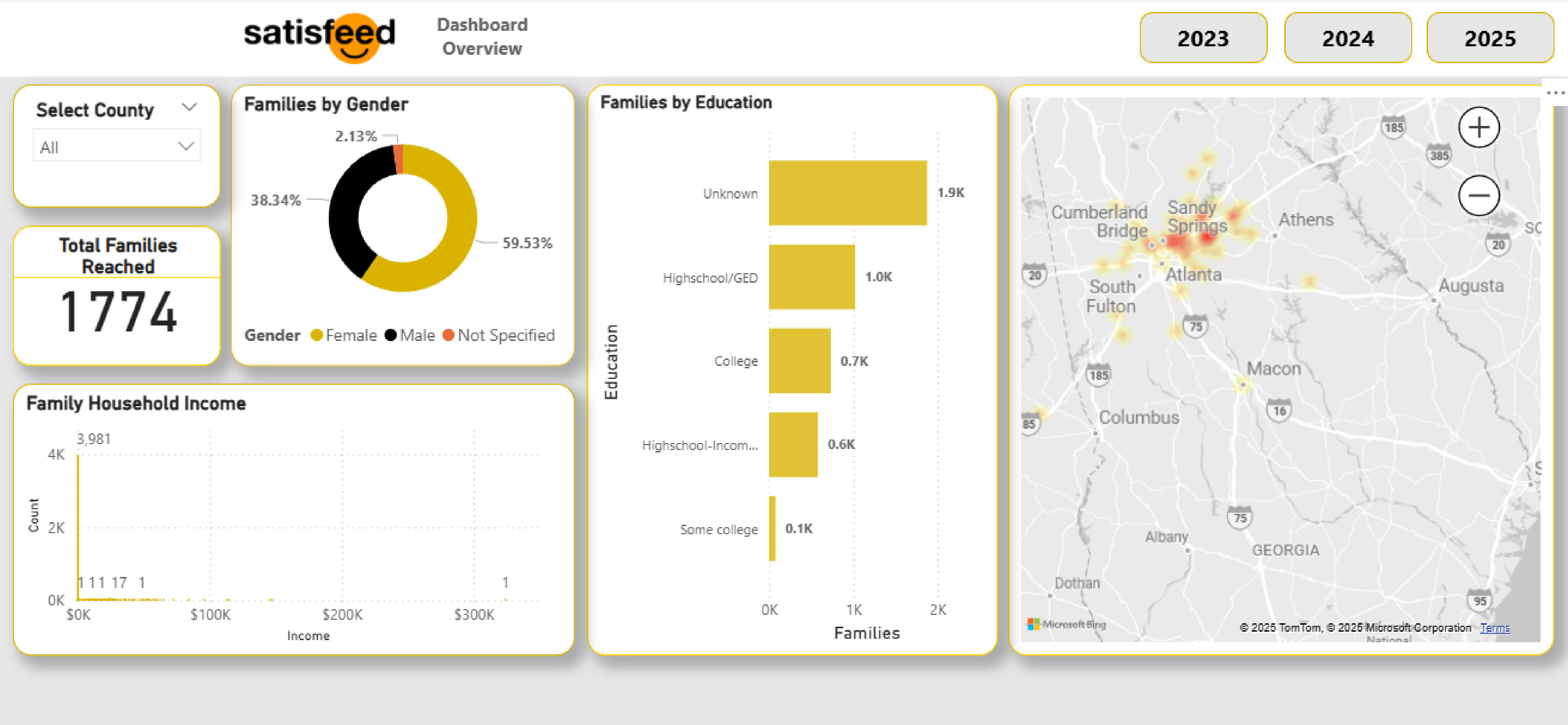
Damaris

AGE GROUP DISTRIBUTION BY RACE

- HYPOTHESIS WAS PARTIALLY INCORRECT
- TOP AGE GROUP IS 41-50
- TOP RACE IS HISPANIC, LATINO, OR SPANISH ORIGIN



ITERATION 1 OF SATISFEED DASHBOARD



satisfeed

Dashboard Overview

2023 | 2024 | 2025

Select County

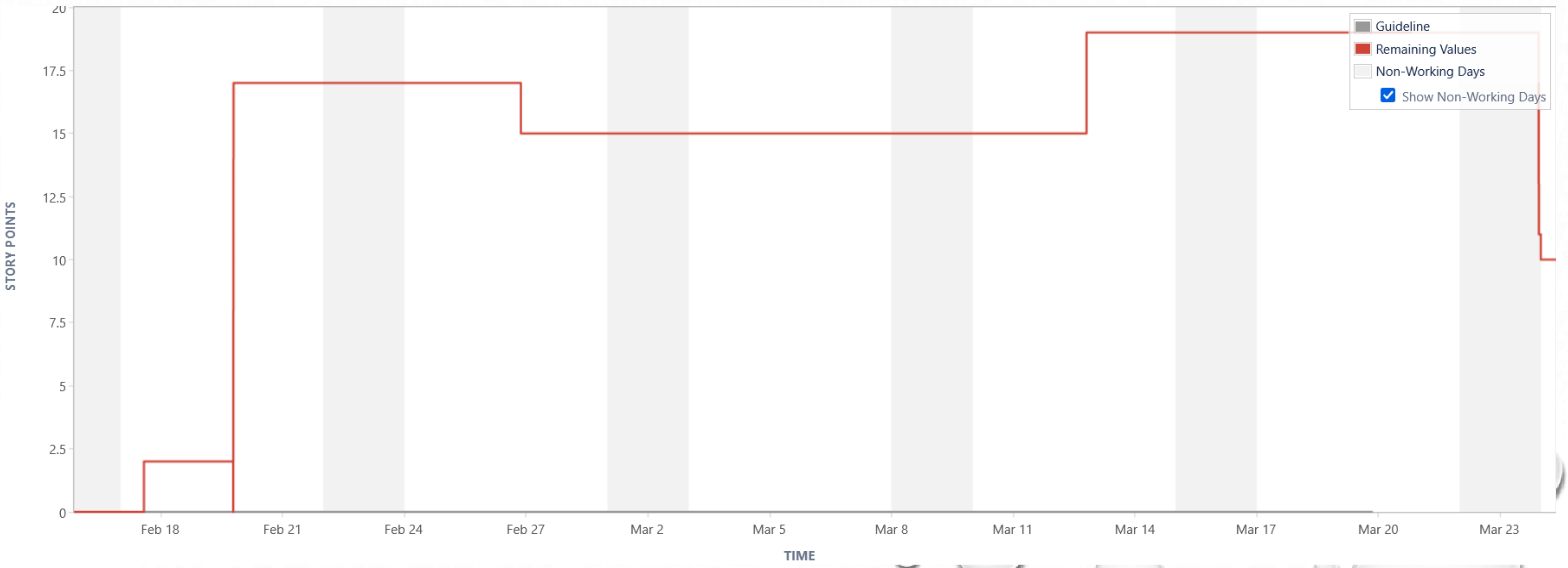
All

Total Families Reached

1774

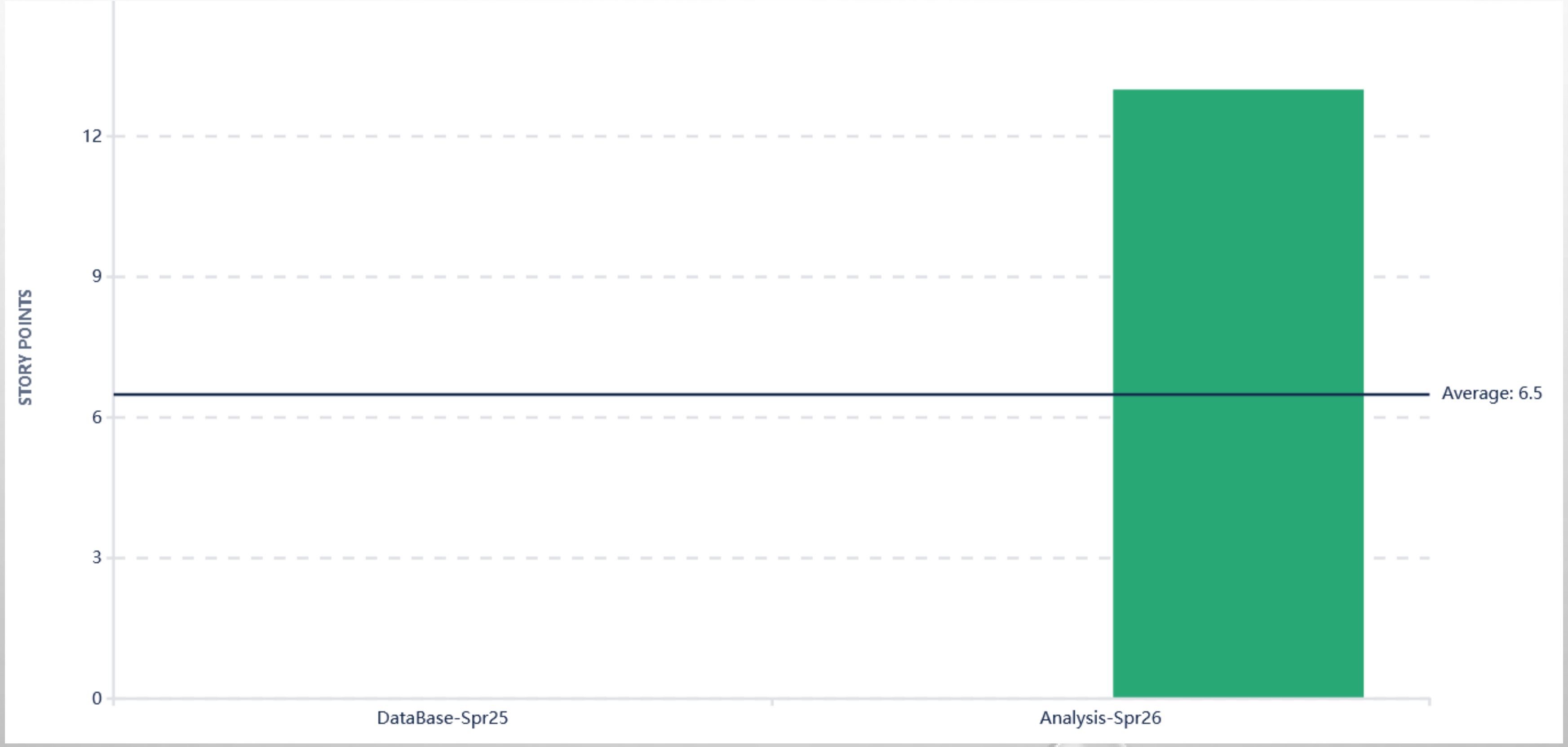
Develop some sort of trend line here to run analysis on how much food has been needed and how much is predicted to be needed

BURNDOWN



☑ Show Non-Working Days

VELOCITY


STORY POINTS
0
3
6
9
12
Average: 6.5
DataBase-Spr25
Analysis-Spr26