Neighborhood Obesity and WalkScore: Mapping the Relationship

Team Members:

- Mauricio Andrews
- Nathaniel Cervantez
- Isaac Gish
- Bryant Griessel
- Mark Helotie
- Manasi Shidhaye

Project Goals / Brainstorming

1. We decided to explore data within the field of healthcare.
2. We started to look at some COVID data -- many datasets available.
3. Came across some datasets on Data.gov for obesity
4. Finally arrived at the intersection of obesity and “walk score” of a neighborhood.

Hypothesis / Research Questions

Original Hypothesis:

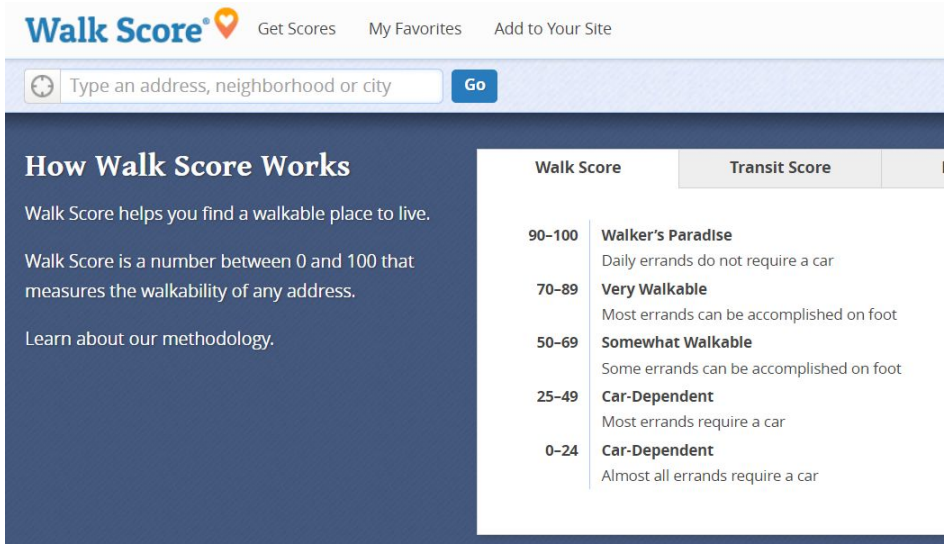
There should be a negative correlation between walkability and obesity,
indicating that neighborhoods with higher walkability scores will have lower obesity rates, and vice-versa.

Research Questions:

- What is the link, if any, between obesity and walkability scores of neighborhoods in the United States?
- Are there any potential confounding variables that could affect the relationship between walkability and obesity, and how will they be addressed in the analysis?
- What other factors may contribute to obesity rates within neighborhoods?

Dataset Information

https://data.cdc.gov/500-Cities-Places/500-Cities-Local-Data-for-Better-Health-2016-relea/9z78-nsfp



https://www.walkscore.com/methodology.shtml

Data Cleansing - Obesity

This full dataset was downloaded from https://catalog.data.gov/dataset/500-cities-local-data-for-better-health-2016-release

The original CSV download was 217MB. Unneeded rows and columns were manually removed, which resulted in a CSV of 4.5MB
(We were facing many size challenges in trying to upload the full CSV into GitHub, so we trimmed it down to what we needed.)

The resultant CSV has over 28,000 data points, encompassing data from all 50 states.

Some notes for the data in this CSV:

- Column F ("UniqueID") is comprised of the combination of "CityFIPS" (column M) and "TractFIPS" (column N).
- Column H ("Data_Value") is the actual obesity percentage of adults (18+).
- Column K ("Population2010") is the population of this specific census tract.
- Column L ("GeoLocation") is the latitude/longitude coordinates of the specific census tract.

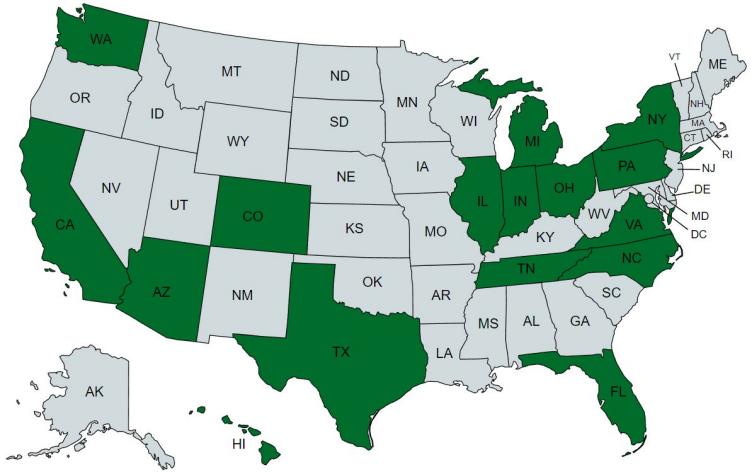
Data Cleansing - WalkScore

This dataset was obtained via API calls from https://www.walkscore.com/professional/api.php

It took several days to pull the data, because we faced limitations of 5,000 API calls per day / per key. Several team members tag-teamed this data retrieval process.

We ended up with over 20,000 data points, which were pulled from the 15 most populous states in the USA. (plus Hawaii, because who doesn't want to know more about Hawaii)

Here are the states we pulled data for:



Task Breakdown

1) Download 217MB master CSV from CDC.gov. Clean up this data to only include the needed data for Obesity.

2) Using the lat/long within the Obesity CSV, obtain the WalkScore for each row via an API call.

3) WalkScore is obtained per state; clean up this data and combine all outputs into one CSV.

4) Clean up the merged data to analyze statistical correlations.

5) Create visualizations, maps, and graphical output to analyze all data collected. (in light of the original hypothesis)

Code Snippets

[*]:
```
def getWalkScore(lat, lon, city):
    query_url = url+city+"&lat="+lat+"&lon="+lon+"&wsapikey="+walkscore_key
    walk_response=requests.get(query_url)
    walk_json = walk_response.json()
    return walk_json

for index, row in tx_data_df_cleaned.iterrows():
    try:

        lat=str(row["Lat"])
        long=str(row["Lon"])
        city=row["CityName"]
        walk_json=getWalkScore(lat,long,city)
        tx_data_df_cleaned.loc[index, "Walk Score"] = walk_json["walkscore"]
    except:
        print("Data not found. Skipping")
        pass
```

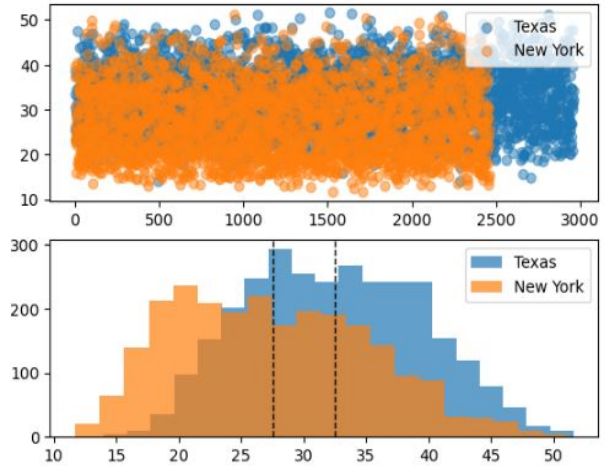
```
Data not found. Skipping
Data not found. Skipping
Data not found. Skipping
```

[22]:
```
# Now let's iterate thru that ONE state and get the data

for index,row in one_state.iterrows():

    try:
        lat = str(row["Lat"])
        lon = str(row["Lon"])
        query_url = url+"&lat="+lat+"&lon="+lon+"&wsapikey="+walkscore_key
        walk_response = requests.get(query_url)
        walk_json = walk_response.json()
        if walk_json["status"] == 41:
            print("Daily API quota exceeded")
            print("^^^^^^^^^^^^^^^^^^^^^^^^^")
            break

        print("Processing LAT %s, LON, %s" % (lat, lon))
        one_state.loc[index, "Walk Score"] = walk_json["walkscore"]

    except:
        print("Data not found. Skipping")
        pass
```

```
Processing LAT 47.636204596, LON, -122.275885393
Processing LAT 47.2242452813, LON, -122.497172711
Processing LAT 47.5643288754, LON, -122.134709858
Processing LAT 47.4131488108, LON, -122.24356832
Processing LAT 45.5967183312, LON, -122.518320907
Processing LAT 47.9417269181, LON, -122.203880637
Processing LAT 47.4503896809, LON, -122.178457056
Processing LAT 47.7067275502, LON, -122.366496857
Processing LAT 47.1939936737, LON, -122.496412414
Processing LAT 47.2791184551, LON, -122.155070317
Processing LAT 47.5537877711, LON, -122.310225765
Processing LAT 47.6717888713, LON, -117.400191522
Processing LAT 47.6658472901, LON, -117.271911536
Processing LAT 47.6051125357, LON, -122.324946395
Processing LAT 47.6589661327, LON, -122.3230895
Processing LAT 47.6645733026, LON, -117.208769317
Processing LAT 47.7078416216, LON, -117.387014933
Processing LAT 47.692321988, LON, -117.371213698
Processing LAT 47.3026458556, LON, -122.404407882
```

Code Snippets (cont'd.)

```python
# Creating a Scatter Plot of obesity data for one state compared to another
x = len(obesity_df.loc[obesity_df['State'] == "TX"].ObesityScore)
x2 = len(obesity_df.loc[obesity_df['State'] == "NY"].ObesityScore)
plt.subplot(2, 1, 1)
plt.scatter(range(x), obesity_df.loc[obesity_df['State'] == "TX"].ObesityScore, label="Texas", alpha = .5)
plt.scatter(range(x2), obesity_df.loc[obesity_df['State'] == "NY"].ObesityScore,label ="New York", alpha = .5)
plt.legend()

# Histogram Plot of above Data
plt.subplot(2, 1, 2)
plt.hist(obesity_df.loc[obesity_df['State'] == "TX"].ObesityScore, 20, alpha=0.7, label="Texas")
plt.hist(obesity_df.loc[obesity_df['State'] == "NY"].ObesityScore, 20, alpha=0.7, label="New York")
plt.axvline(obesity_df.loc[obesity_df['State'] == "TX"].ObesityScore.mean(), color='k', linestyle='dashed', linewidth=1)
plt.axvline(obesity_df.loc[obesity_df['State'] == "NY"].ObesityScore.mean(), color='k', linestyle='dashed', linewidth=1)
plt.legend()
plt.show()
```



[38]:
```python
# Calculate the quartiles for ObesityScore
obesity_quartiles = gdf["ObesityScore"].quantile([0.25, 0.5, 0.75])

# Calculate the quartiles for WalkScore
walk_quartiles = gdf["WalkScore"].quantile([0.25, 0.5, 0.75])

print("ObesityScore Quartiles:")
print(obesity_quartiles)

print("\nWalkScore Quartiles:")
print(walk_quartiles)
```

```
ObesityScore Quartiles:
0.25    23.0
0.50    28.1
0.75    34.6
Name: ObesityScore, dtype: float64

WalkScore Quartiles:
0.25    26.0
0.50    51.0
0.75    74.0
Name: WalkScore, dtype: float64
```

[5]:
```python
# Create a new column to identify locations with both very low or both very high WalkScore and ObesityScore
gdf = gdf.merge(grouped_data, on="State")

# Filter the GeoDataFrame to only include the locations based on the third quartiles.
outliers_df = gdf[(gdf["ObesityScore"] > 34.6) & (gdf["WalkScore"] > 74)]

# Create a scatter plot to visualize the outliers
plt.figure(figsize=(12, 8))
plt.scatter(outliers_df["ObesityScore"], outliers_df["WalkScore"], c="red", marker="o", edgecolors="black", alpha=0.75)

# Annotate the points with state labels
for i, row in outliers_df.iterrows():
    plt.annotate(row["State"], (row["ObesityScore"], row["WalkScore"]), textcoords="offset points", xytext=(5,5), ha='center')

plt.xlabel("Obesity Score (0-100%)")
plt.ylabel("Walk Score (0-100%)")
plt.title("Locations with Extreme ObesityScore and WalkScore (ObesityScore > 34.6, WalkScore > 74)")
plt.grid(True)
plt.show()
```

Code Snippets (cont'd.)

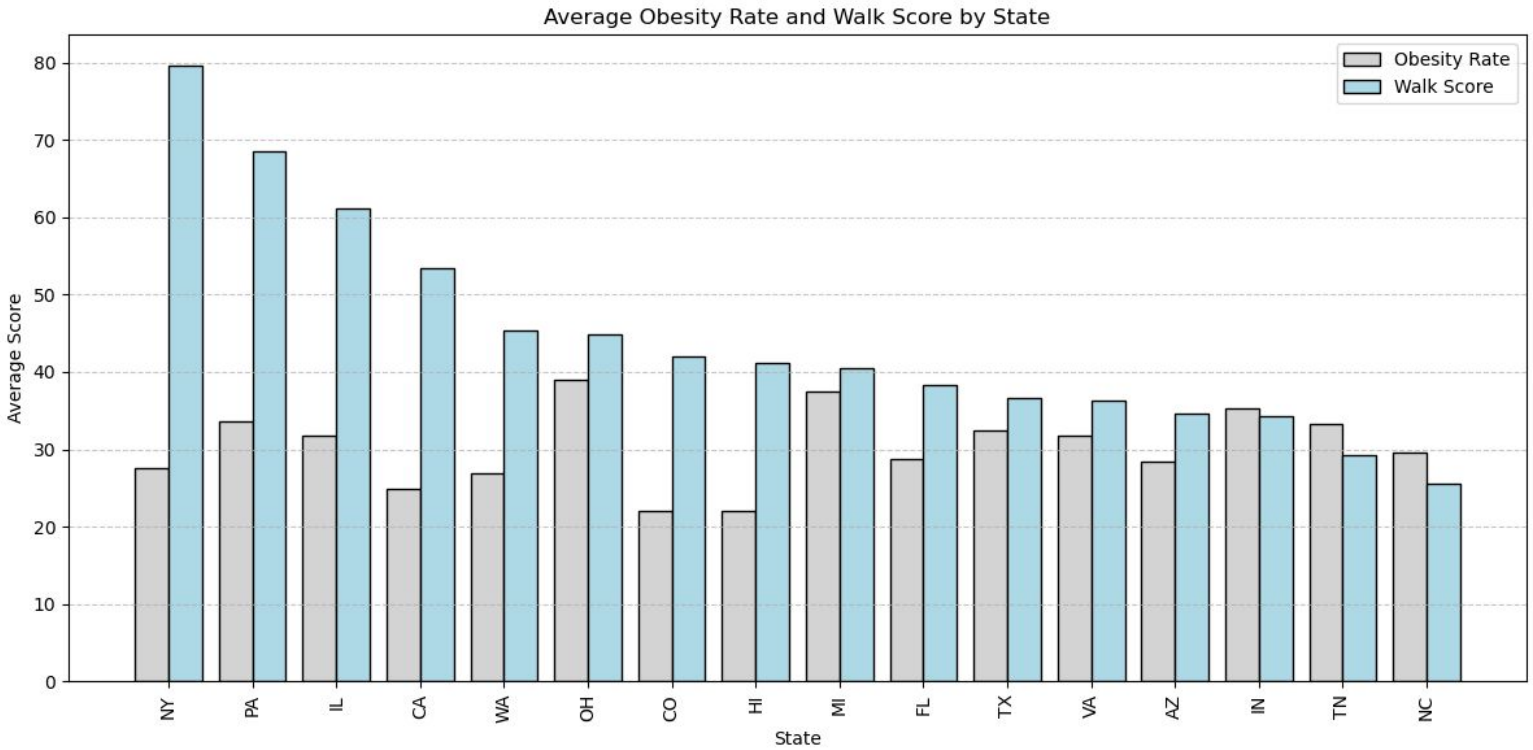
```
#Visualization 4 (Basic Scatterplot)

#Creating variables to match columns
walk_score = merged_df["WalkScore"]
obesity_rate = merged_df["ObesityScore"]

#Creating the scatterplot
plt.scatter(obesity_rate, walk_score, marker="o", facecolors="red", edgecolors="black",
            alpha=0.75,)

#Making lables for the axis and title
plt.ylabel("Walk Score (0-100)")
plt.xlabel("Obesity Rate (0-100%)")
plt.title("Walk Score vs Obesity Rate")

#Save Image
plt.savefig("../Images/Base_Scatterplot.png")

plt.show()
```

Analysis / Visualizations

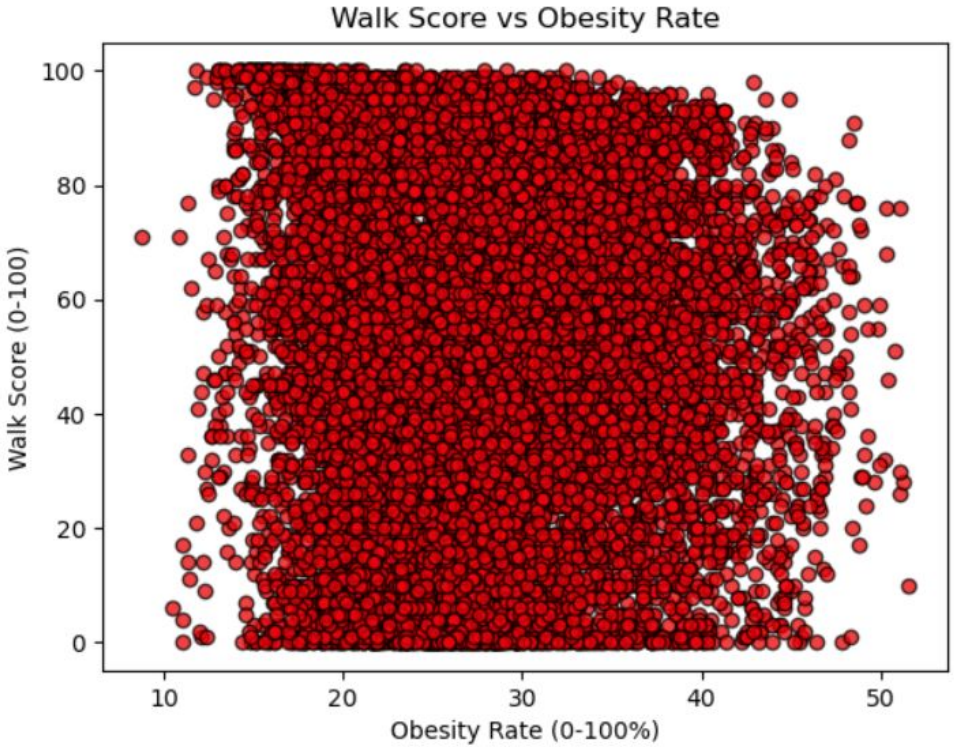
The following visualizations were created:

1. Bar chart comparing average Obesity Rates and Walk Score by state
2. Scatterplot comparing Obesity Rates and Walk Score
3. Scatter plot comparing largest 4 states
4. Map plot representation of walk/obesity scores
5. Map plot aggregated view of the data
6. Visual of locations with high/low Obesity and Walking scores
7. Box plots for individual states

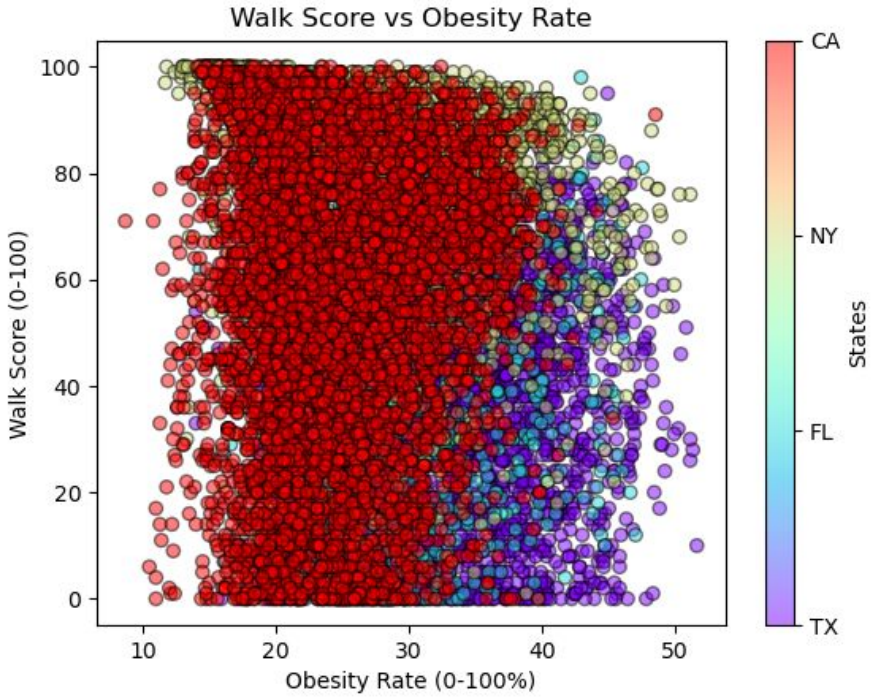
Visualization 1 - Bar graph to compare Average Obesity Rate and Walk Score by State



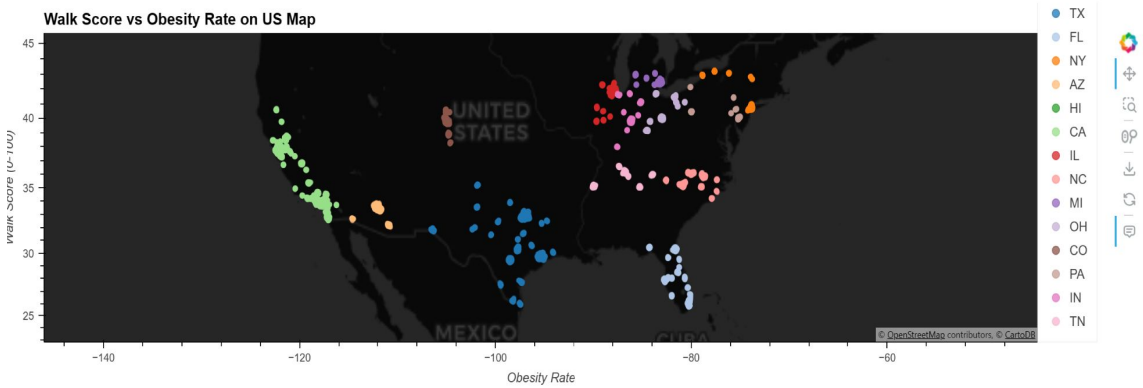
Visualization 2 – All States Walk Score vs Obesity Rate

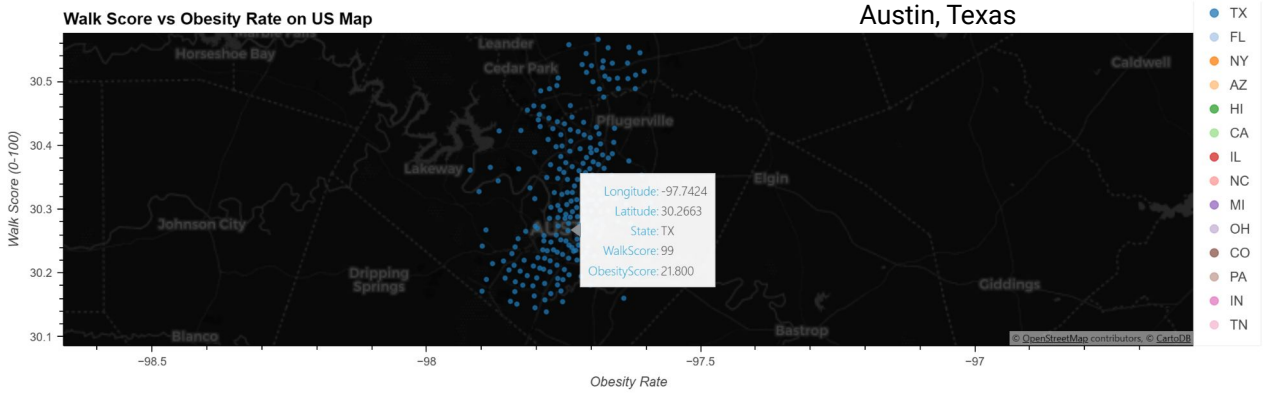


Visualization 3 - Comparing 4 Largest States

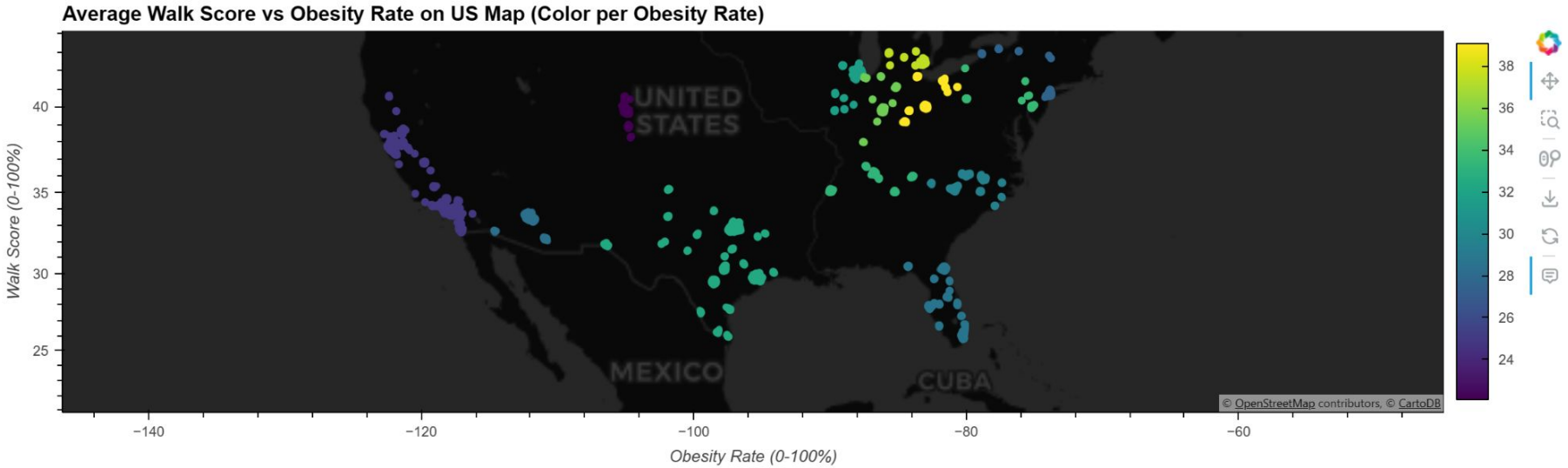


Visualization 4 -

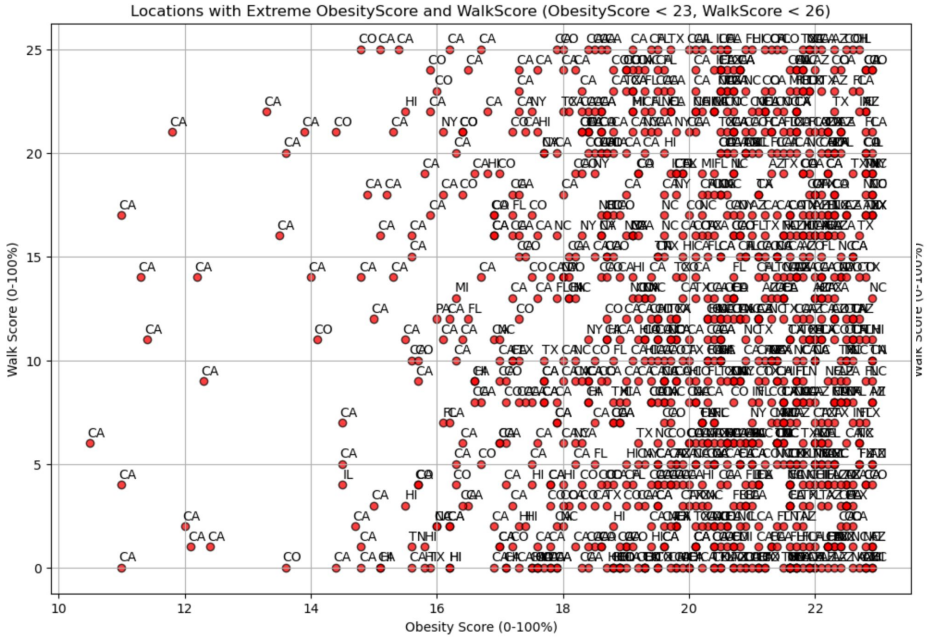




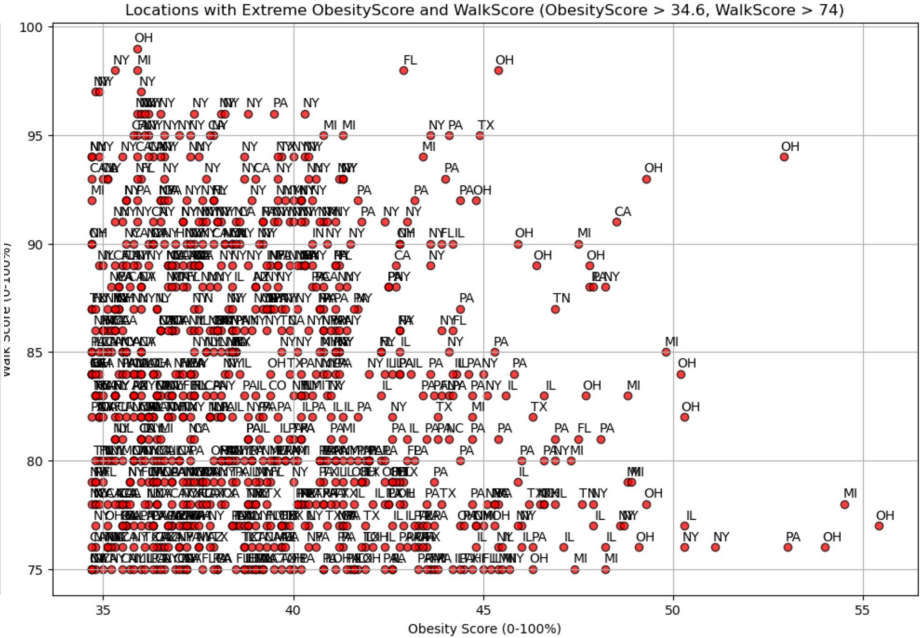
Visualization 5 -



Visualization 6 -

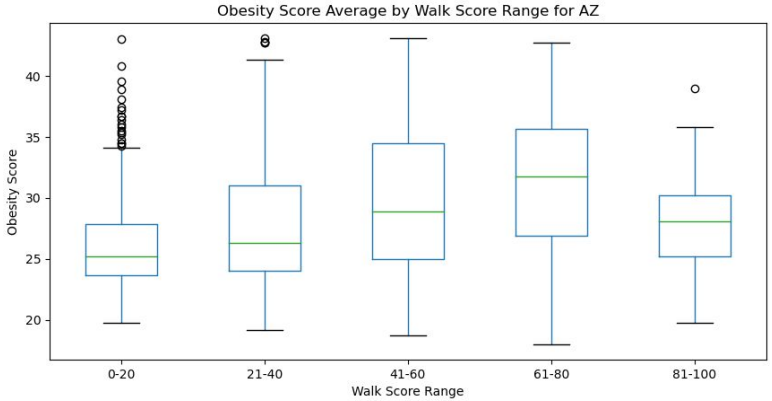


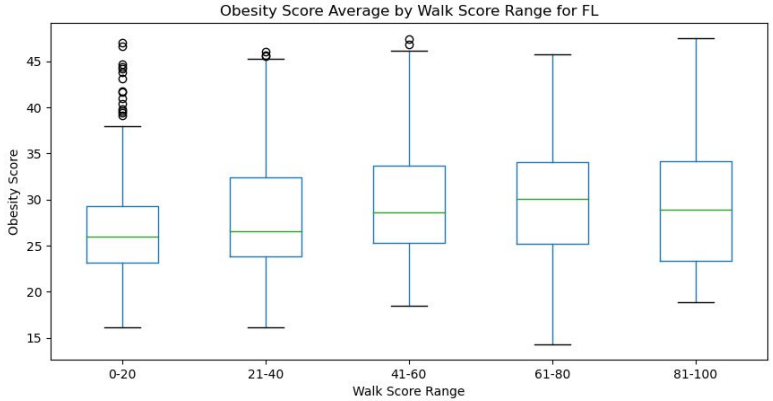
First Quartile

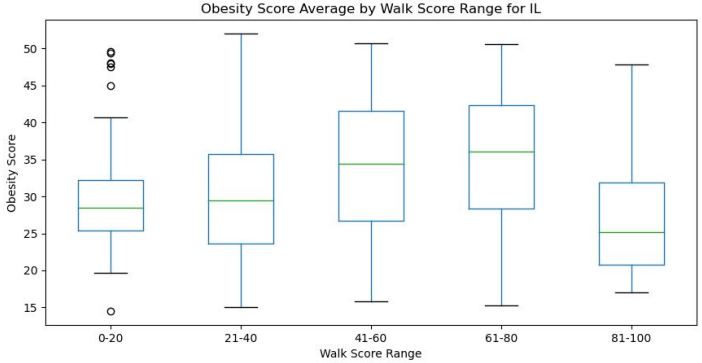


Third Quartile

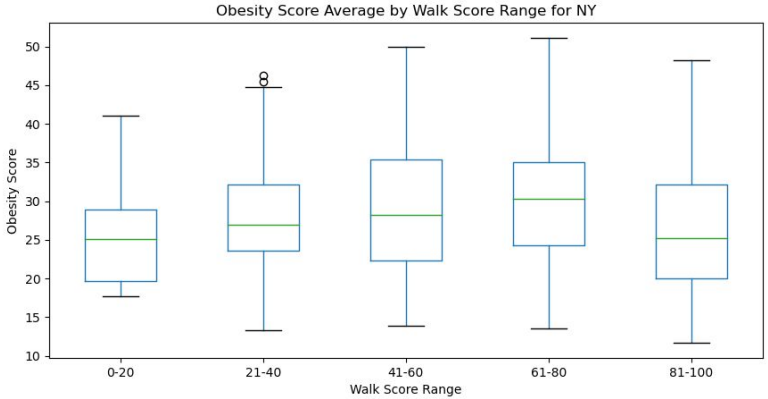
Box Plots

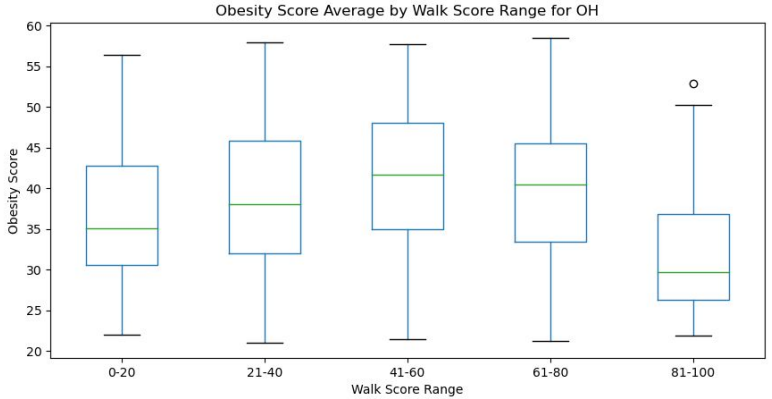

Obesity Score Average by Walk Score Range for AZ
Obesity Score
Walk Score Range
0-20
21-40
41-60
61-80
81-100

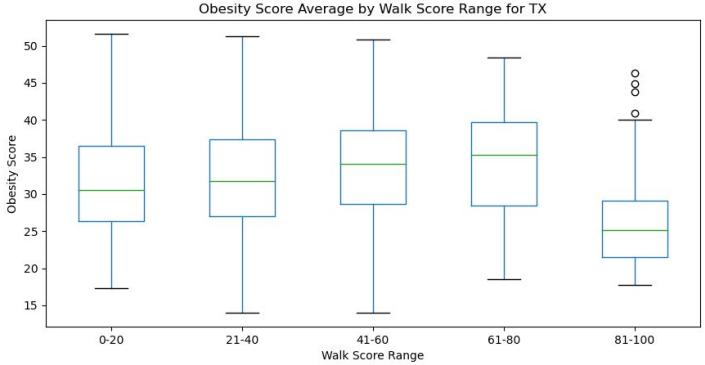

Obesity Score Average by Walk Score Range for FL
Obesity Score
Walk Score Range
0-20
21-40
41-60
61-80
81-100


Obesity Score Average by Walk Score Range for IL
Obesity Score
Walk Score Range
0-20
21-40
41-60
61-80
81-100

Box Plots (cont'd.)







Pearson Correlation Coefficient

```
# Extract data for Texas (TX)
tx_data = obesity_df.loc[obesity_df['State'] == "TX"].dropna()

# Extract ObesityScore and WalkScore columns
tx_obesity_scores = tx_data['ObesityScore']
tx_walk_scores = tx_data['WalkScore']

# Drop NaN values, if any
tx_obesity_scores = tx_obesity_scores.dropna()
tx_walk_scores = tx_walk_scores.dropna()
```

```
#Running pearson's corr coefficient test to determine correlation between obesity and walk score if any
stats.pearsonr(tx_obesity_scores, tx_walk_scores)
```

```
PearsonRResult(statistic=0.07049798887190445, pvalue=0.00015862303084408303)
```

```
# Extract data for New York (NY)
ny_data = obesity_df.loc[obesity_df['State'] == "NY"]

# Extract ObesityScore and WalkScore columns
ny_obesity_scores = ny_data['ObesityScore']
ny_walk_scores = ny_data['WalkScore']

# Drop NaN values, if any
ny_obesity_scores = ny_obesity_scores.dropna()
ny_walk_scores = ny_walk_scores.dropna()
```

```
#Running pearson's corr coefficient test to determine correlation between obesity and walk score if any
stats.pearsonr(ny_obesity_scores, ny_walk_scores)
```

```
PearsonRResult(statistic=-0.17360546277210986, pvalue=4.112649585531752e-18)
```

```
# Extract data for Florida (FL)
fl_data = obesity_df.loc[obesity_df['State'] == "FL"]

# Extract ObesityScore and WalkScore columns
fl_obesity_scores = fl_data['ObesityScore']
fl_walk_scores = fl_data['WalkScore']

# Drop NaN values, if any
fl_obesity_scores = fl_obesity_scores.dropna()
fl_walk_scores = fl_walk_scores.dropna()
```

```
#Running pearson's corr coefficient test to determine correlation between obesity and walk score if any
stats.pearsonr(fl_obesity_scores, fl_walk_scores)
```

```
PearsonRResult(statistic=0.21766212979222443, pvalue=2.367550357514103e-15)
```

- The Pearson correlation coefficient (r) is the most common way of measuring a linear correlation. It is a number between –1 and 1 that measures the strength and direction of the relationship between two variables.
- A low statistic shows a weak correlation between Walk Score and Obesity Rate
- A low p-value shows a possible correlation, however, a low statistic suggests that while there is a statistically significant relationship, the strength of that relationship is weak. This could indicate that other factors might be influencing the relationship.

Conclusion

Our original hypothesis was

> “There should be a negative correlation between walkability and obesity,
> indicating that neighborhoods with higher walkability scores will have lower obesity rates, and vice versa.”

After looking at our datasets and visualizations, we came to the conclusion that this is NOT true. (null hypothesis)

Further research:

- Determine other datasets to find correlations between walkability & obesity. i.e.: nearest gym, nearest McD, nearest grocery

Any questions for the Team?