Applied Project - Phoenix Sustainability Dashboard

Greenway: Locate Nearby Landfills and Transfer Stations

Ira A. Fulton Schools of Engineering

Arizona State University

IFT 593: Applied Project

Dr. Asmaa Elbadrawy

Due Date: 04/11/2024

Project Team:

Megha Jotangiya

Vrutik Adani

Harshita Mishra

Bhavya Atul Shah

Contents

Project Overview:

Sustainability is the practice of meeting current needs without compromising the ability of future generations to meet their own needs. This principle is crucial for several reasons, with one of the key aspects being environmental conservation. When we talk about sustainability, we're emphasizing the importance of using resources efficiently and reducing waste. This approach plays a significant role in preserving natural resources like clean air, water, forests, and biodiversity. By managing resources responsibly, we can protect ecosystems and minimize the impact of climate change.

The "Green Way: Locate Nearby Landfills & Transfer Stations" project contributes to sustainability in several ways:

1. **Efficient Waste Management:**
    - The project aims to enhance waste management efficiency by providing users with a web application to locate nearby landfills and transfer stations within the Phoenix city area.
    - Efficient waste management is crucial for sustainability as it reduces pollution, minimizes resource depletion, and prevents environmental degradation.
    - By enabling users to easily find nearby waste facilities, the project promotes proper waste disposal, which contributes to preserving natural resources and ecosystems.
    - Proper waste disposal also helps in reducing the release of harmful substances into the environment, thus safeguarding clean air, water, and soil for future generations.

2. **Reduces Carbon Footprint:**
    - One of the key benefits of the "Green Way" project is its contribution to reducing carbon footprints.
    - The project achieves this by facilitating easy access to waste facilities, which reduces the need for long-distance transportation of waste materials.
    - By minimizing transportation distances, the project helps in lowering vehicle emissions, which are a major contributor to climate change.
    - Lowering carbon emissions aids in climate change mitigation efforts and fosters environmental sustainability by curbing the rate of global warming and its associated impacts on ecosystems and human health.

3. **Supports Green Infrastructure:**
    - The "Green Way" project supports the adoption of green infrastructure practices by providing digital tools that enhance waste management efficiency and promote sustainable urban development.
    - Green infrastructure refers to natural or nature-based solutions for managing stormwater, reducing urban heat islands, improving air quality, and enhancing overall environmental quality.
    - Through interactive maps and geolocation services, the project showcases the importance of green infrastructure in waste management and urban planning.
    - By promoting green infrastructure adoption, the project encourages the use of sustainable practices that reduce resource consumption, improve resilience to environmental challenges, and enhance the quality of life for communities.

Scope and Objective:

The scope of the "Green Way" web application is to provide users with a convenient tool for locating nearby landfills and transfer stations within the Phoenix city area. The objective is to streamline waste disposal processes and promote sustainable waste management practices by offering users easy access to essential waste facilities.

Key Features and Functionalities:

1. **User Address Validation:**
    - Users can input their address into the application.
    - The application validates the entered address against an active_service_addresses.csv file to ensure accuracy and eligibility.
    - Address validation helps prevent errors and ensures that users receive relevant information based on their location.
2. **Location Display:**
    - Upon successful address validation, the application displays the entered address to users.
    - It also provides service area information, indicating which landfills and transfer stations are within proximity to the user's address.
    - Additionally, the application includes a link to the bulk-trash collection schedule, providing users with further waste management resources.
    - The user's address is visually represented on a map using the OpenStreetMap API, enhancing the user experience and facilitating navigation.
3. **Distance Calculation:**
    - The application utilizes the Mapbox Direction API to calculate the distance between the user's address and nearby landfills and transfer stations.
    - Distance calculation helps determine the nearest station to the user, enabling efficient waste disposal planning.
4. **Nearest Station Search:**
    - Users can initiate a search for the nearest landfill or transfer station by clicking on the designated button within the application.
    - The application calculates the distance to each station based on the user's address and displays the nearest station prominently.
    - It also shows the shortest driving route between the user's address and the selected station on the map, facilitating navigation.

5. **Detailed Nearest Station Information:**
    - Upon selecting a station, the application provides users with detailed information about it.
    - This includes the station's name, address, operating hours, holidays, and directions for reaching it.
    - Additionally, the application displays the distance in miles from the user's address to the selected station, assisting users in making informed decisions about waste disposal options.
6. **Error Handling:**
    - The application implements robust error handling mechanisms to address unexpected errors that may occur during address validation, map loading, or API interactions.
    - Error handling ensures smooth functionality and enhances the user experience by minimizing disruptions and providing clear guidance in case of technical issues.

By incorporating these key features and functionalities, the "Green Way" web application aims to empower users with the tools and information necessary to make sustainable choices regarding waste disposal. It prioritizes user experience, accuracy, and reliability to promote efficient waste management practices and contribute to environmental sustainability in the Phoenix city area.

Data Sources:

1. **Public_Works_Solid_Waste_Active_Service_Addresses.csv:**
    - This dataset was obtained from the official Phoenix Open Data website at link.
    - It contains information about active service addresses related to solid waste management in the Phoenix city area.
    - The dataset likely includes details such as addresses, service status, and other relevant information necessary for validating user input addresses within the application.

Column Name	Data Type	Description
service address	object	Address of the property
City	object	City name
State	object	State abbreviation
Zip	object	Zip code
INCITYLIMIT	object	Indicator of city limits
REFUSE	object	Refuse collection service
RECYCLE	object	Recycling collection service
BULK_TRASH	object	Bulk trash collection service
GREEN_ORG	object	Green waste/organic waste collection service
ELIGIBLEGO	object	Eligibility for services
QUARTERSECTION	object	Quarter section
PROPERTY_TYPE	object	Property type
PREM_DESCRIPTION	object	Property description
NUMBER_OF_CONTAINERS	int	Number of waste containers
IS_IN_ALLEY	object	Indicator if property is in an alley
LIVINGUNITS	int	Number of living units
SERVICE_AREA	object	Service area or zone
GIS_X_COORDINATE	float	X-coordinate in GIS format
GIS_Y_COORDINATE	float	Y-coordinate in GIS format

Data Dictionary: Public_Works_Solid_Waste_Active_Service_Addresses.csv

2. **Transfer_Station_Dataframe.csv:**
    - This file was created by the project team and contains data specific to landfills and transfer stations within the Phoenix city area.
    - It serves as a reference for the locations and details of these waste facilities, including their names, addresses, and possibly operational hours or other relevant information.
    - The data from this file is utilized by the application to provide users with information about nearby waste disposal options.

Column Name	Description
Name	Name of the location or station
Station_Address	Address of the station
Latitude	Latitude coordinates of the location
Longitude	Longitude coordinates of the location

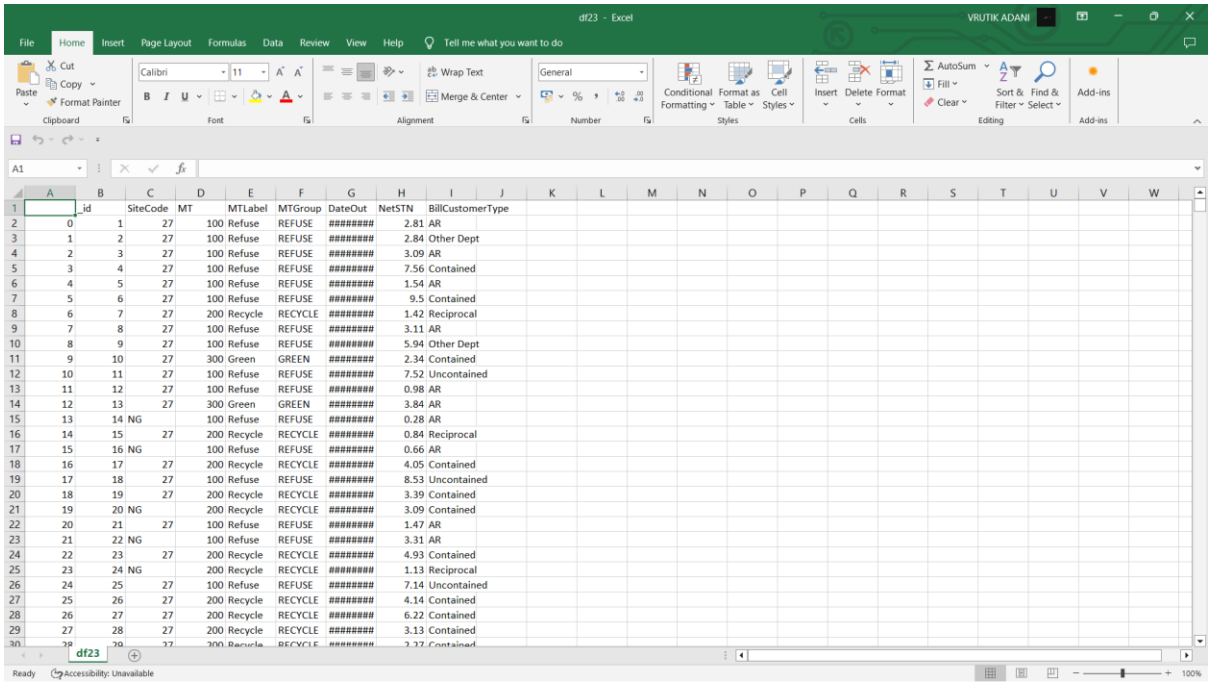
Data Dictionary: Transfer_Station_Dataframe.csv

3. **City_Limit_Dark_Outline.geojson:**
    - The project utilizes the "City_Limit_Dark_Outline.geojson" file sourced from the Phoenix Open Data portal at link.
    - This file contains geographical data representing the boundary of the Phoenix city area.
    - It is used to visualize the extent of the city area within the application, providing users with context regarding the coverage area for waste management services.
    - The boundary data helps users understand which waste facilities fall within the jurisdiction of Phoenix and are accessible through the application.


```
{
"type": "FeatureCollection",
"name": "City_Limit_Dark_Outline",
"crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
"features": [
{ "type": "Feature", "id": 2563, "properties": { "OBJECTID": 2563, "NAME": "City of Phoenix" }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ -112.145716962238453, 33.914708785788051 ],
[ -112.147714269392196, 33.914709790387505 ], [ -112.148043893770648, 33.914709910871075 ], [ -112.152230805771495, 33.914715481647868 ], [ -112.152227490058678, 33.918354975547501 ], [ -112.169692066848114,
33.918357615928223 ], [ -112.187127217297657, 33.918359842282122 ], [ -112.203537085681191, 33.918359689815482 ], [ -112.203548190957861, 33.914727986842763 ], [ -112.203577870704734, 33.913752299052128 ],
[ -112.212139876600745, 33.913720973725738 ], [ -112.220701888478004, 33.913689163132624 ], [ -112.220742112820588, 33.906485718031703 ], [ -112.220782380210693, 33.899274525563655 ], [ -112.221085625013515,
33.89928476654655 ], [ -112.220987640317276, 33.892041305596734 ], [ -112.220984910032513, 33.891839478074161 ], [ -112.220891777006642, 33.884953146849682 ], [ -112.220888762920296, 33.88473061226442 ],
[ -112.220940621050332, 33.877546602996404 ], [ -112.220964247813654, 33.874273173103533 ], [ -112.207921711486833, 33.874158832981387 ], [ -112.207964499511434, 33.877491798228235 ], [ -112.203568525391546,
33.877472919849055 ], [ -112.203575416145185, 33.873851934284772 ], [ -112.212273937855002, 33.873859754282648 ], [ -112.220968009618005, 33.873961959503568 ], [ -112.220992471927192, 33.870362577140547 ],
[ -112.221032590658155, 33.863123605377567 ], [ -112.221072707463662, 33.855884627882588 ], [ -112.221085376392438, 33.848568757475754 ], [ -112.220999791943115, 33.841383992909051 ], [ -112.220999822466524,
33.840478398253687 ], [ -112.220999691799022, 33.840430387546533 ], [ -112.220996754296976, 33.840253921968433 ], [ -112.221046401756922, 33.840267683315496 ], [ -112.221094842366483, 33.840282884930161 ],
[ -112.221173697259189, 33.84030913337547 ], [ -112.221291130163308, 33.840352732028293 ], [ -112.224231221936876, 33.841528672469721 ], [ -112.228272770735018, 33.843147773824171 ], [ -112.228502206531189,
33.843226567156108 ], [ -112.228740841043447, 33.843291635192 ], [ -112.228886964022465, 33.843322247485759 ], [ -112.229098663779013, 33.843354494439758 ], [ -112.229283936625279, 33.843372316800014 ],
[ -112.229384184267971, 33.843377572706828 ], [ -112.229484592538583, 33.843379906791718 ], [ -112.229635234347413, 33.843377919542725 ], [ -112.229714170006289, 33.843374248172395 ], [ -112.229829768289191,
33.843365248332994 ], [ -112.229963617515423, 33.843350609402059 ], [ -112.230108232549455, 33.843327955186442 ], [ -112.230258276226451, 33.843298441538828 ], [ -112.230447999487623, 33.843249793944246 ],
[ -112.230680674701802, 33.843173623411467 ], [ -112.230864559833591, 33.843099368610034 ], [ -112.231066172979453, 33.843000661595944 ], [ -112.231246931203415, 33.842895667375579 ], [ -112.231363433808355,
33.842818802562462 ], [ -112.231548586805605, 33.842677333611604 ], [ -112.231680311173648, 33.842559333088019 ], [ -112.231805700969232, 33.842431031885639 ], [ -112.231891601814567, 33.842331506206179 ],
[ -112.231945570835393, 33.842263001322571 ], [ -112.232021511100598, 33.842156987794709 ], [ -112.232096534352976, 33.842038087521907 ], [ -112.232143059411129, 33.841955140582009 ], [ -112.232193112781488,
33.841855467312811 ], [ -112.232258994127085, 33.841721622142792 ], [ -112.232277729433396, 33.841647891037681 ], [ -112.232305767241868, 33.841561215331552 ], [ -112.232346746353628, 33.841480752435354 ],
[ -112.234116542588922, 33.834155060600509 ], [ -112.235890116179633, 33.827100410657778 ], [ -112.221003980568454, 33.827129575993241 ], [ -112.216322358899461, 33.827104577509658 ], [ -112.216330641728082,
33.823226278270369 ], [ -112.212311278418909, 33.823200642669605 ], [ -112.207958243285645, 33.823173226964862 ], [ -112.207959709759976, 33.819552123712917 ], [ -112.212310483701557, 33.819574733229288 ],
[ -112.216667710740268, 33.819597695717988 ], [ -112.216652158943112, 33.826831961723499 ], [ -112.221003006877112, 33.826854757936054 ], [ -112.228678190719549, 33.826843009126428 ], [ -112.233065613321656,
33.826835944627184 ], [ -112.235955691956552, 33.826831202370506 ], [ -112.238504327809736, 33.826826971502925 ], [ -112.252991265006116, 33.826897108997372 ], [ -112.240369097181537, 33.812390124333419 ],
[ -112.238427875173343, 33.812386812548809 ], [ -112.238232203660118, 33.80887265060144 ], [ -112.233929972880134, 33.808810949642854 ], [ -112.233985674184865, 33.805265159192807 ], [ -112.229603430126261,
33.805161407127954 ], [ -112.220904322704115, 33.804954986759599 ], [ -112.217647534201122, 33.804968744382663 ], [ -112.217874407054381, 33.804551863574361 ], [ -112.218051965281617, 33.804281111548305 ],
[ -112.218208717794326, 33.803996555989578 ], [ -112.218414918552639, 33.803607907613625 ], [ -112.218604731906296, 33.803281680725917 ], [ -112.218864533188246, 33.802788932622668 ], [ -112.219195056341988,
33.802341062020429 ], [ -112.219517275192118, 33.801893213656356 ], [ -112.219665434590924, 33.801528964064367 ], [ -112.219718197284365, 33.801158798153992 ], [ -112.219694655051157, 33.800723950750537 ],
[ -112.219664247140869, 33.800423881722452 ], [ -112.219619534610743, 33.800229777729733 ], [ -112.219608087570265, 33.800102434323129 ], [ -112.219629751475452, 33.800004374218744 ], [ -112.219662444797166,
33.799912412796367 ], [ -112.219680398325352, 33.799802110242246 ], [ -112.219672589263396, 33.799667370608731 ], [ -112.219657439478794, 33.799532649000291 ], [ -112.219623872240817, 33.799376533774904 ],
[ -112.219561045, 33.79924499073342 ], [ -112.219465154768514, 33.799101277258579 ], [ -112.21933997242855, 33.798972951387569 ], [ -112.219203829115429, 33.798856903818198 ], [ -112.219115291373683,
33.798716236291135 ], [ -112.219109555299838, 33.798578163421716 ], [ -112.219173060581781, 33.798452699999948 ], [ -112.219273588964214, 33.798365711473721 ], [ -112.219397689899736, 33.798221743943017 ],
[ -112.219497777703126, 33.797990857388614 ], [ -112.219497027533748, 33.797785555877226 ], [ -112.219424690033136, 33.797463730230856 ], [ -112.219328800255156, 33.797196456699119 ], [ -112.219220994011877,
33.796980939494215 ], [ -112.219089374333308, 33.796602576777126 ], [ -112.218993532697894, 33.796345210117131 ], [ -112.218921442088558, 33.796092737583471 ], [ -112.218831237231839, 33.795751140683771 ],
```


Data: City_Limit_Dark_Outline.geojson

4. **Df23:**
    - This dataset has been used for Tableau Visualizations.

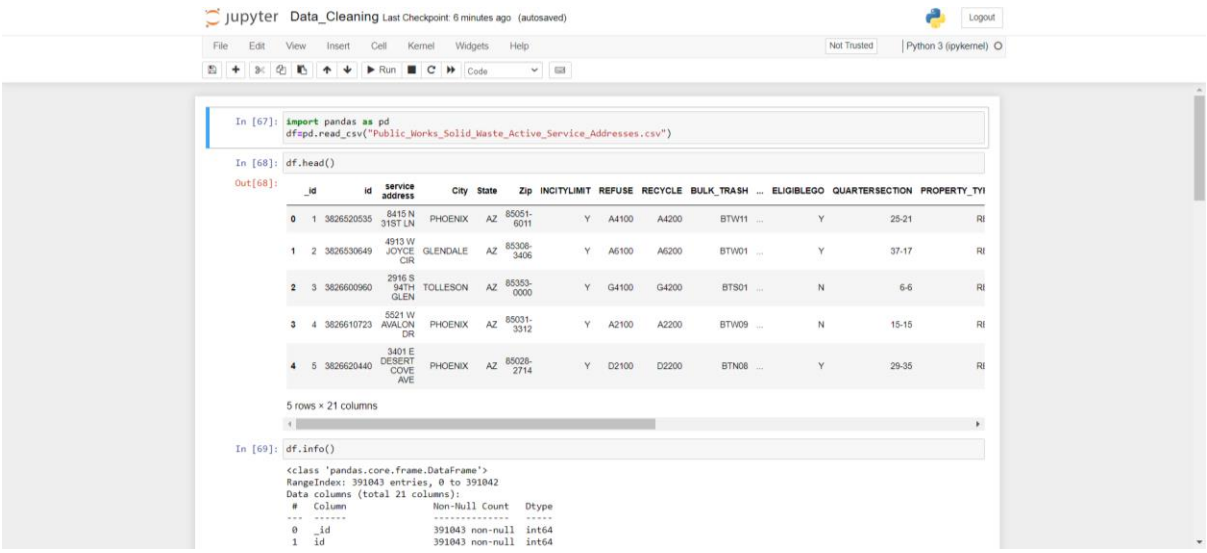


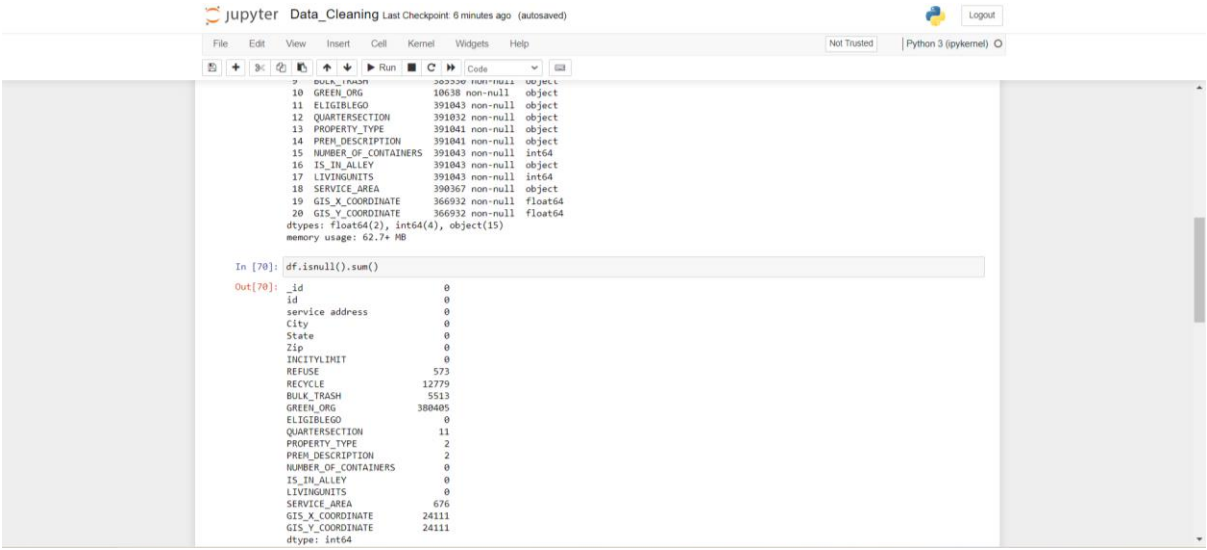
	A	B	C	D	E	F	G	H	I
1		_id	SiteCode	MT	MTLabel	MTGroup	DateOut	NetSTN	BillCustomerType
2	0	1	27	100	Refuse	REFUSE	########	2.81	AR
3	1	2	27	100	Refuse	REFUSE	########	2.84	Other Dept
4	2	3	27	100	Refuse	REFUSE	########	3.09	AR
5	3	4	27	100	Refuse	REFUSE	########	7.56	Contained
6	4	5	27	100	Refuse	REFUSE	########	1.54	AR
7	5	6	27	100	Refuse	REFUSE	########	9.5	Contained
8	6	7	27	200	Recycle	RECYCLE	########	1.42	Reciprocal
9	7	8	27	100	Refuse	REFUSE	########	3.11	AR
10	8	9	27	100	Refuse	REFUSE	########	5.94	Other Dept
11	9	10	27	300	Green	GREEN	########	2.34	Contained
12	10	11	27	100	Refuse	REFUSE	########	7.52	Uncontained
13	11	12	27	100	Refuse	REFUSE	########	0.98	AR
14	12	13	27	300	Green	GREEN	########	3.84	AR
15	13	14	NG	100	Refuse	REFUSE	########	0.28	AR
16	14	15	27	200	Recycle	RECYCLE	########	0.84	Reciprocal
17	15	16	NG	100	Refuse	REFUSE	########	0.66	AR
18	16	17	27	200	Recycle	RECYCLE	########	4.05	Contained
19	17	18	27	100	Refuse	REFUSE	########	8.53	Uncontained
20	18	19	27	200	Recycle	RECYCLE	########	3.39	Contained
21	19	20	NG	200	Recycle	RECYCLE	########	3.09	Contained
22	20	21	27	100	Refuse	REFUSE	########	1.47	AR
23	21	22	NG	100	Refuse	REFUSE	########	3.31	AR
24	22	23	27	200	Recycle	RECYCLE	########	4.93	Contained
25	23	24	NG	200	Recycle	RECYCLE	########	1.13	Reciprocal
26	24	25	27	100	Refuse	REFUSE	########	7.14	Uncontained
27	25	26	27	200	Recycle	RECYCLE	########	4.14	Contained
28	26	27	27	200	Recycle	RECYCLE	########	6.22	Contained
29	27	28	27	200	Recycle	RECYCLE	########	3.13	Contained

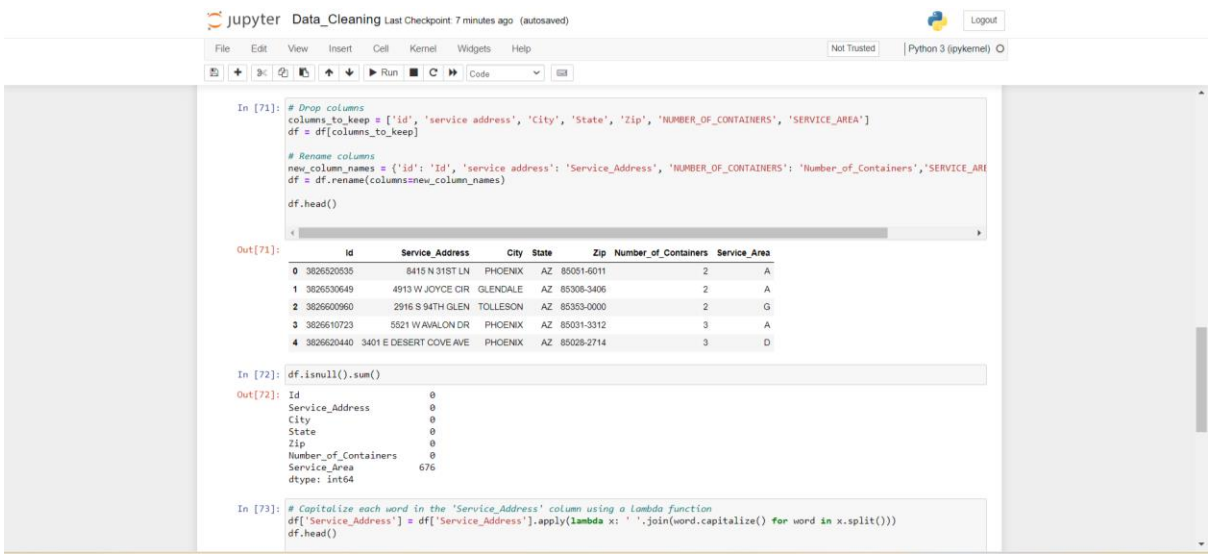
Contents of df23.csv

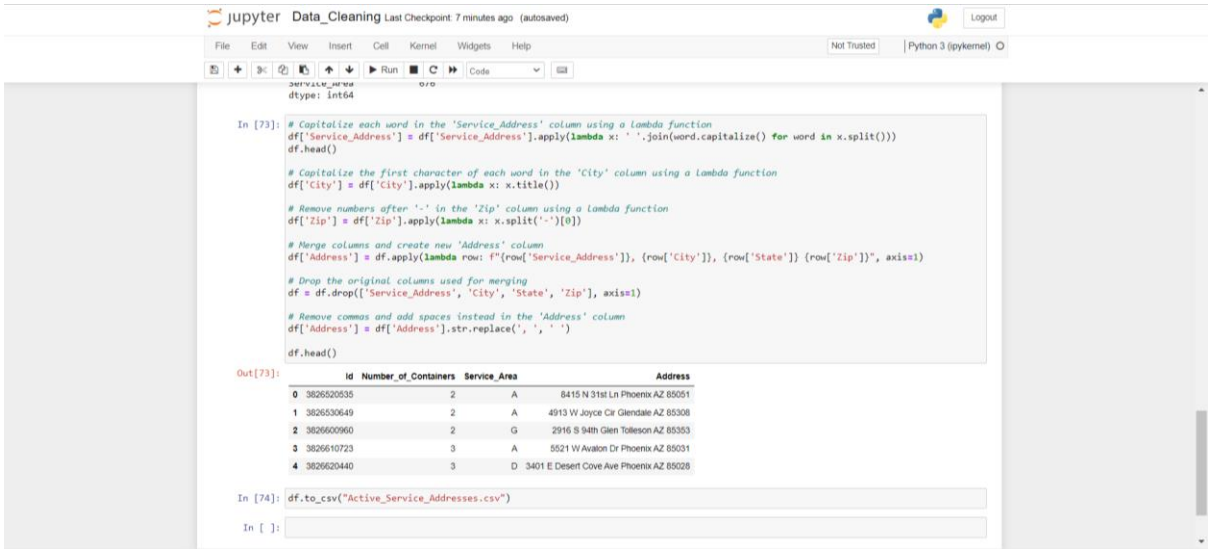
By leveraging these data sources, the "Green Way" web application is able to provide users with accurate and relevant information about waste management services within the Phoenix city area. The combination of address validation, facility data, and geographical boundaries enhances the functionality and usefulness of the application, ultimately supporting the project's objective of promoting sustainable waste management practices.

Data Cleaning:

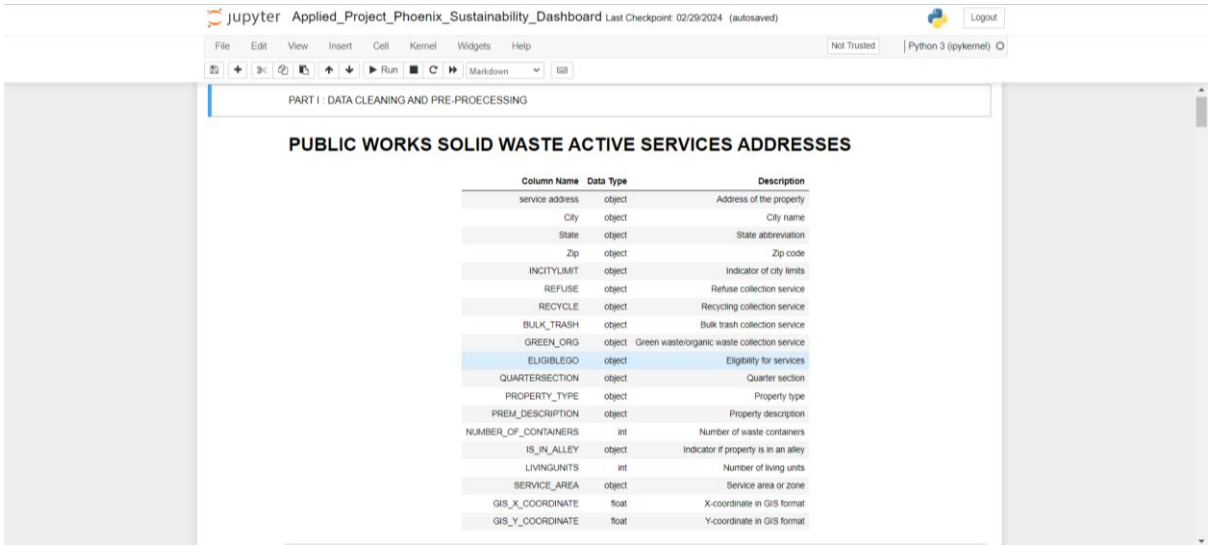





jupyter Data_Cleaning Last Checkpoint: 7 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [71]:
# Drop columns
columns_to_keep = ['id', 'service address', 'City', 'State', 'Zip', 'NUMBER_OF_CONTAINERS', 'SERVICE_AREA']
df = df[columns_to_keep]
# Rename columns
new_column_names = {'id': 'Id', 'service address': 'Service_Address', 'NUMBER_OF_CONTAINERS': 'Number_of_Containers','SERVICE_AR
df = df.rename(columns=new_column_names)
df.head()
Out[71]:
Id Service_Address City State Zip Number_of_Containers Service_Area
0 3826520535 8415 N 31ST LN PHOENIX AZ 85051-6011 2 A
1 3826530649 4913 W JOYCE CIR GLENDALE AZ 85308-3406 2 A
2 3826600960 2916 S 94TH GLEN TOLLESON AZ 85353-0000 2 G
3 3826610723 5521 W AVALON DR PHOENIX AZ 85031-3312 3 A
4 3826620440 3401 E DESERT COVE AVE PHOENIX AZ 85028-2714 3 D
In [72]:
df.isnull().sum()
Out[72]:
Id 0
Service_Address 0
City 0
State 0
Zip 0
Number_of_Containers 0
Service_Area 676
dtype: int64
In [73]:
Capitalize each word in the 'Service_Address' column using a lambda function
df['Service_Address'] = df['Service_Address'].apply(lambda x: ' '.join(word.capitalize() for word in x.split()))
df.head()


jupyter Data_Cleaning Last Checkpoint: 7 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
dtype: int64
In [73]:
Capitalize each word in the 'Service_Address' column using a lambda function
df['Service_Address'] = df['Service_Address'].apply(lambda x: ' '.join(word.capitalize() for word in x.split()))
df.head()
# Capitalize the first character of each word in the 'City' column using a lambda function
df['City'] = df['City'].apply(lambda x: x.title())
Remove numbers after '-' in the 'Zip' column using a lambda function
df['Zip'] = df['Zip'].apply(lambda x: x.split('-')[0])
Merge columns and create new 'Address' column
df['Address'] = df.apply(lambda row: f"{row['Service_Address']}, {row['City']}, {row['State']} {row['Zip']}", axis=1)
# Drop the original columns used for merging
df = df.drop(['Service_Address', 'City', 'State', 'Zip'], axis=1)
# Remove commas and add spaces instead in the 'Address' column
df['Address'] = df['Address'].str.replace(', ', ' ')
df.head()
Out[73]:
Id Number_of_Containers Service_Area Address
0 3826520535 2 A 8415 N 31st Ln Phoenix AZ 85051
1 3826530649 2 A 4913 W Joyce Cir Glendale AZ 85308
2 3826600960 2 G 2916 S 94th Glen Tolleson AZ 85353
3 3826610723 3 A 5521 W Avalon Dr Phoenix AZ 85031
4 3826620440 3 D 3401 E Desert Cove Ave Phoenix AZ 85028
In [74]:
df.to_csv("Active_Service_Addresses.csv")
In [ ]:

Initial Data Preprocessing:



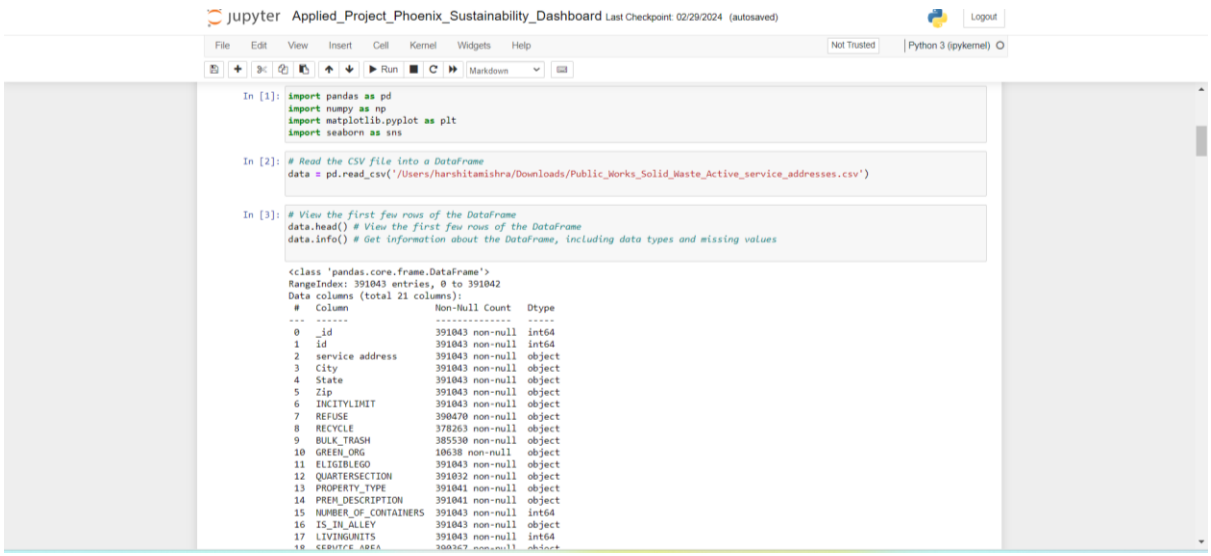
jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

PART I : DATA CLEANING AND PRE-PROECESSING

PUBLIC WORKS SOLID WASTE ACTIVE SERVICES ADDRESSES

Column Name	Data Type	Description
service address	object	Address of the property
City	object	City name
State	object	State abbreviation
Zip	object	Zip code
INCITYLIMIT	object	Indicator of city limits
REFUSE	object	Refuse collection service
RECYCLE	object	Recycling collection service
BULK_TRASH	object	Bulk trash collection service
GREEN_ORG	object	Green waste/organic waste collection service
ELIGIBLEGO	object	Eligibility for services
QUARTERSECTION	object	Quarter section
PROPERTY_TYPE	object	Property type
PREM_DESCRIPTION	object	Property description
NUMBER_OF_CONTAINERS	int	Number of waste containers
IS_IN_ALLEY	object	Indicator if property is in an alley
LIVINGUNITS	int	Number of living units
SERVICE_AREA	object	Service area or zone
GIS_X_COORDINATE	float	X-coordinate in GIS format
GIS_Y_COORDINATE	float	Y-coordinate in GIS format



jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)

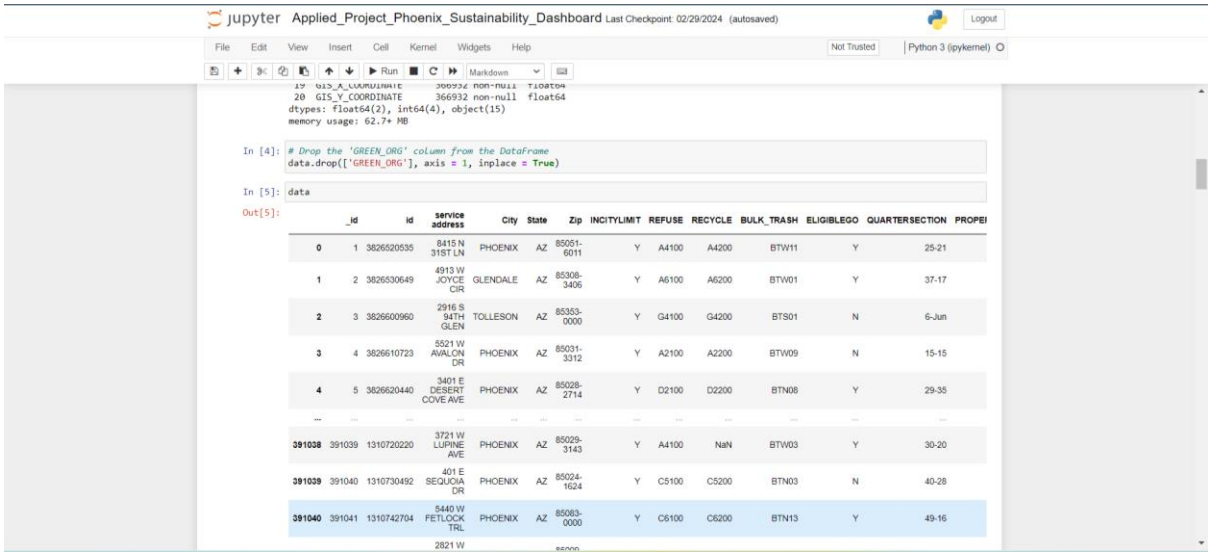
File Edit View Insert Cell Kernel Widgets Help

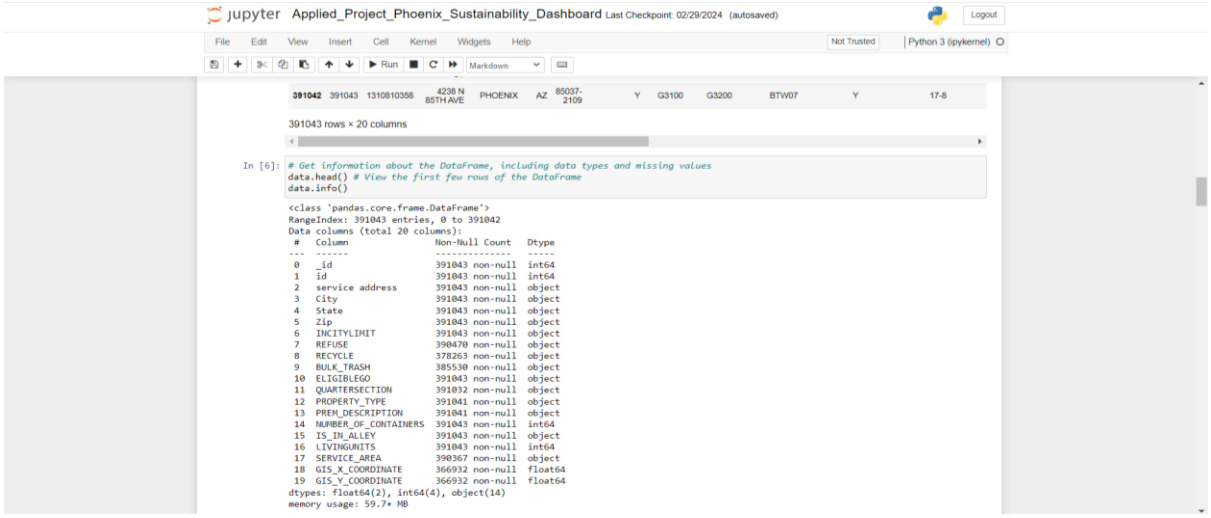
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

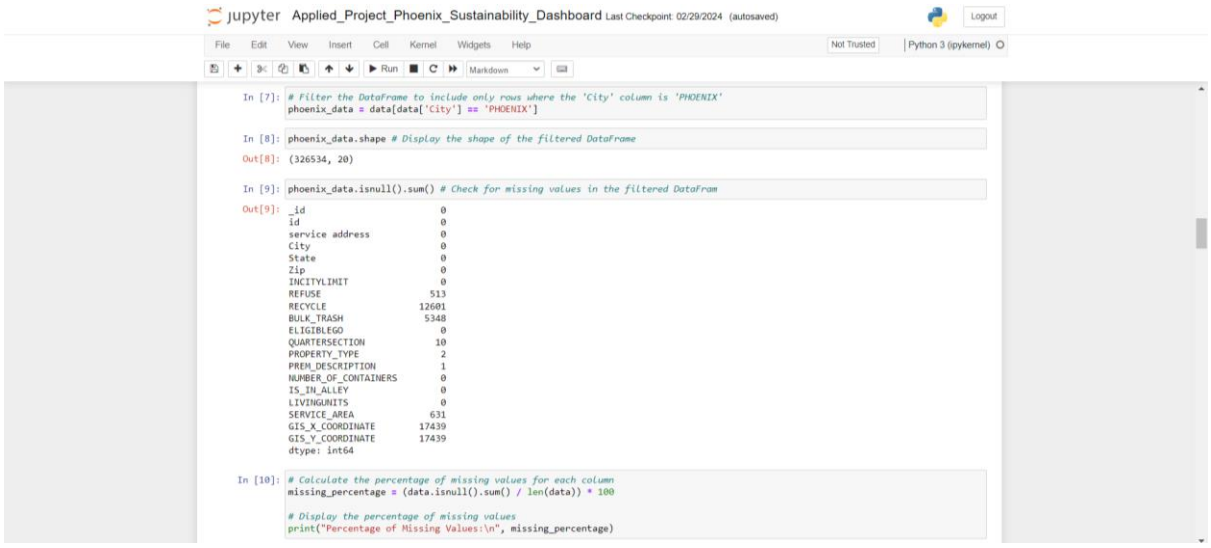
In [2]:
```
# Read the CSV file into a DataFrame
data = pd.read_csv('/Users/harshitamishra/Downloads/Public_Works_Solid_Waste_Active_service_addresses.csv')
```

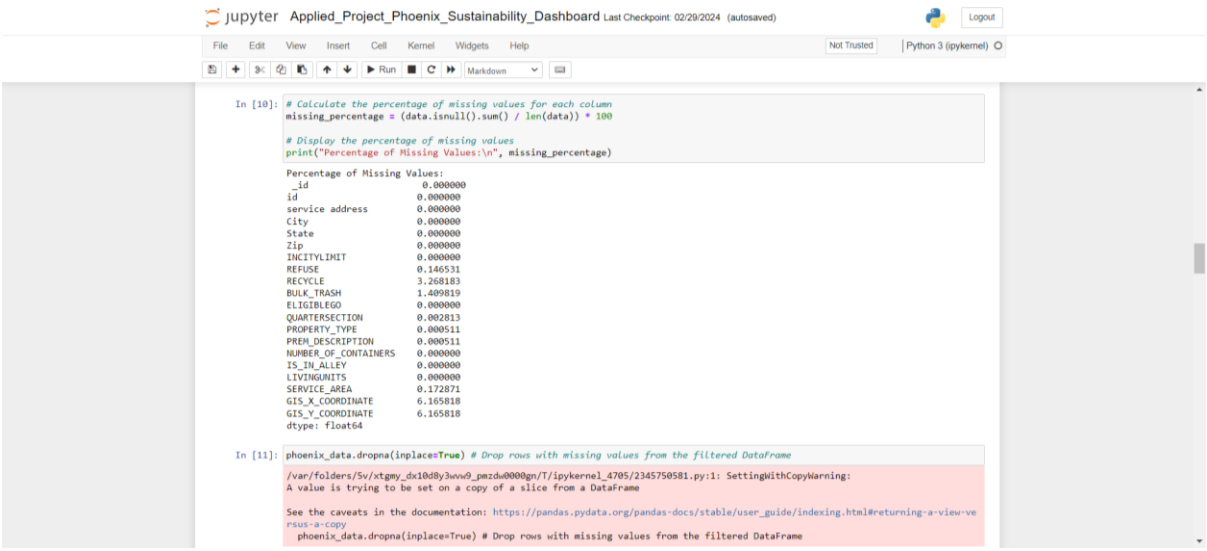
In [3]:
```
# View the first few rows of the DataFrame
data.head() # View the first few rows of the DataFrame
data.info() # Get information about the DataFrame, including data types and missing values
```

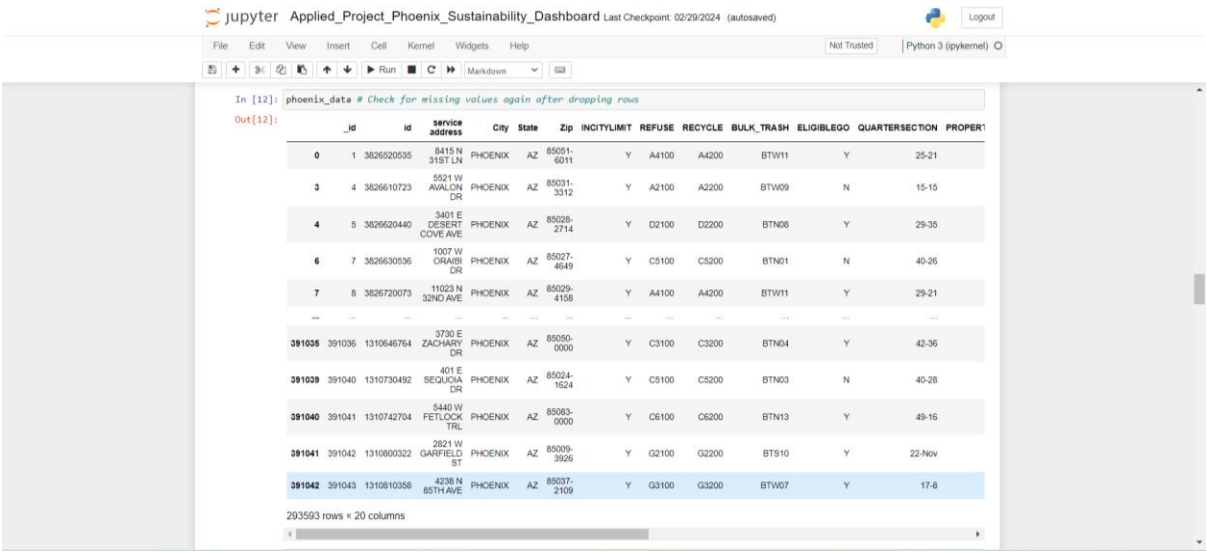
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 391043 entries, 0 to 391042
Data columns (total 21 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   _id                  391043 non-null  int64
 1   Id                   391043 non-null  int64
 2   service address      391043 non-null  object
 3   City                 391043 non-null  object
 4   State                391043 non-null  object
 5   Zip                  391043 non-null  object
 6   INCITYLIMIT          391043 non-null  object
 7   REFUSE               390470 non-null  object
 8   RECYCLE              378263 non-null  object
 9   BULK_TRASH           385530 non-null  object
 10  GREEN_ORG            10638 non-null   object
 11  ELIGIBLEGO           391043 non-null  object
 12  QUARTERSECTION       391032 non-null  object
 13  PROPERTY_TYPE        391041 non-null  object
 14  PREM_DESCRIPTION     391041 non-null  object
 15  NUMBER_OF_CONTAINERS 391043 non-null  int64
 16  IS_IN_ALLEY          391043 non-null  object
 17  LIVINGUNITS          391043 non-null  int64
 18  SERVICE_AREA         390367 non-null  object
```

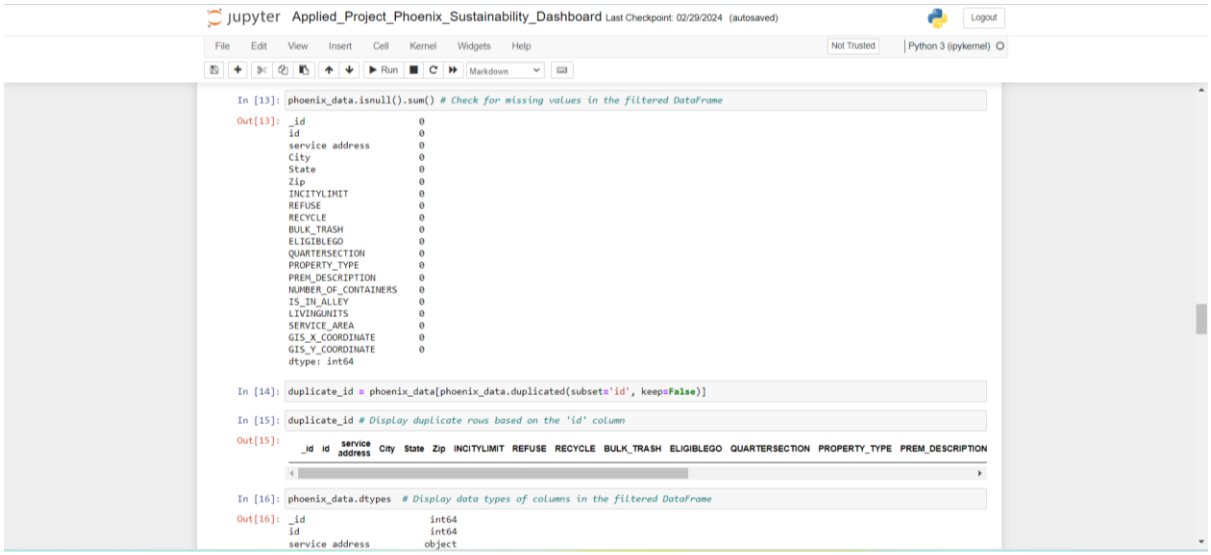

jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
20 GIS_Y_COORDINATE 366932 non-null float64
dtypes: float64(2), int64(4), object(15)
memory usage: 62.7+ MB
In [4]:
# Drop the 'GREEN_ORG' column from the DataFrame
data.drop(['GREEN_ORG'], axis = 1, inplace = True)
In [5]:
data
Out[5]:
_id id service address City State Zip INCITYLIMIT REFUSE RECYCLE BULK_TRASH ELIGIBLEGO QUARTERSECTION PROPEI
0 1 3826520535 8415 N 31ST LN PHOENIX AZ 85051-6011 Y A4100 A4200 BTW11 Y 25-21
1 2 3826530649 4913 W JOYCE CIR GLENDALE AZ 85308-3406 Y A6100 A6200 BTW01 Y 37-17
2 3 3826600960 2916 S 94TH GLEN TOLLESON AZ 85353-0000 Y G4100 G4200 BTS01 N 6-Jun
3 4 3826610723 5521 W AVALON DR PHOENIX AZ 85031-3312 Y A2100 A2200 BTW09 N 15-15
4 5 3826620440 3401 E DESERT COVE AVE PHOENIX AZ 85028-2714 Y D2100 D2200 BTN08 Y 29-35
391038 391039 1310720220 3721 W LUPINE AVE PHOENIX AZ 85029-3143 Y A4100 NaN BTW03 Y 30-20
391039 391040 1310730492 401 E SEQUOIA DR PHOENIX AZ 85024-1624 Y C5100 C5200 BTN03 N 40-28
391040 391041 1310742704 5440 W FETLOCK TRL PHOENIX AZ 85083-0000 Y C6100 C6200 BTN13 Y 49-16
2821 W

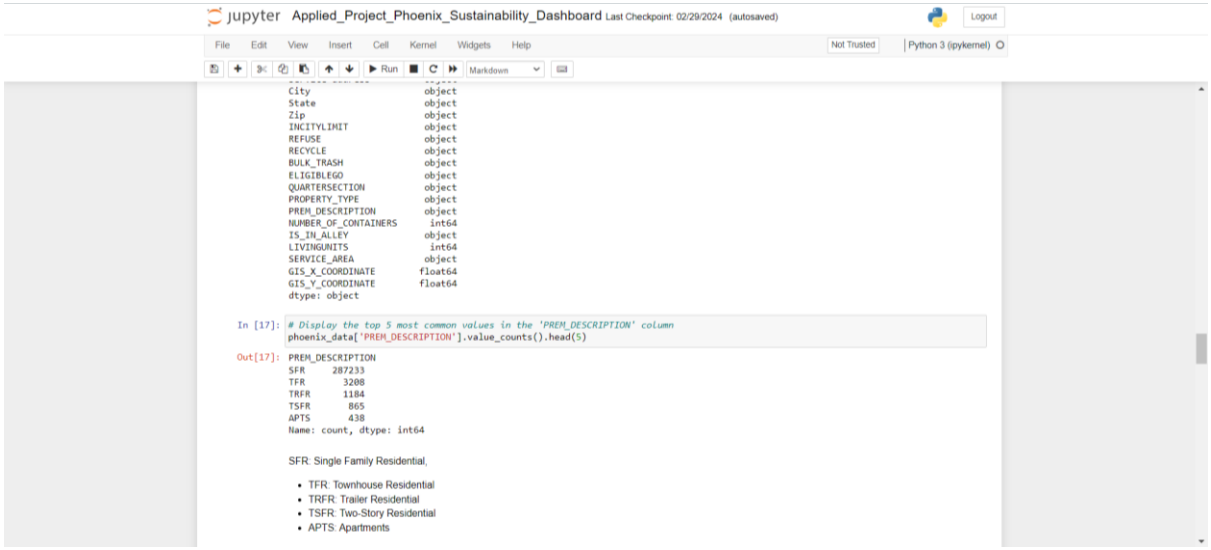

jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
391042 391043 1310810358 4238 N 85TH AVE PHOENIX AZ 85037-2109 Y G3100 G3200 BTW07 Y 17-8
391043 rows × 20 columns
In [6]:
Get information about the DataFrame, including data types and missing values
data.head() # View the first few rows of the DataFrame
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 391043 entries, 0 to 391042
Data columns (total 20 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 _id 391043 non-null int64
1 id 391043 non-null int64
2 service address 391043 non-null object
3 City 391043 non-null object
4 State 391043 non-null object
5 Zip 391043 non-null object
6 INCITYLIMIT 391043 non-null object
7 REFUSE 390470 non-null object
8 RECYCLE 378263 non-null object
9 BULK_TRASH 385530 non-null object
10 ELIGIBLEGO 391043 non-null object
11 QUARTERSECTION 391032 non-null object
12 PROPERTY_TYPE 391041 non-null object
13 PREM_DESCRIPTION 391041 non-null object
14 NUMBER_OF_CONTAINERS 391043 non-null int64
15 IS_IN_ALLEY 391043 non-null object
16 LIVINGUNITS 391043 non-null int64
17 SERVICE_AREA 390367 non-null object
18 GIS_X_COORDINATE 366932 non-null float64
19 GIS_Y_COORDINATE 366932 non-null float64
dtypes: float64(2), int64(4), object(14)
memory usage: 59.7+ MB


jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [7]:
# Filter the DataFrame to include only rows where the 'City' column is 'PHOENIX'
phoenix_data = data[data['City'] == 'PHOENIX']
In [8]:
phoenix_data.shape # Display the shape of the filtered DataFrame
Out[8]:
(326534, 20)
In [9]:
phoenix_data.isnull().sum() # Check for missing values in the filtered DataFram
Out[9]:
_id 0
id 0
service address 0
City 0
State 0
Zip 0
INCITYLIMIT 0
REFUSE 513
RECYCLE 12601
BULK_TRASH 5348
ELIGIBLEGO 0
QUARTERSECTION 10
PROPERTY_TYPE 2
PREM_DESCRIPTION 1
NUMBER_OF_CONTAINERS 0
IS_IN_ALLEY 0
LIVINGUNITS 0
SERVICE_AREA 631
GIS_X_COORDINATE 17439
GIS_Y_COORDINATE 17439
dtype: int64
In [10]:
# Calculate the percentage of missing values for each column
missing_percentage = (data.isnull().sum() / len(data)) * 100
# Display the percentage of missing values
print("Percentage of Missing Values:\n", missing_percentage)


jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [10]:
# Calculate the percentage of missing values for each column
missing_percentage = (data.isnull().sum() / len(data)) * 100
# Display the percentage of missing values
print("Percentage of Missing Values:\n", missing_percentage)
Percentage of Missing Values:
_id 0.000000
id 0.000000
service address 0.000000
City 0.000000
State 0.000000
Zip 0.000000
INCITYLIMIT 0.000000
REFUSE 0.146531
RECYCLE 3.268183
BULK_TRASH 1.409819
ELIGIBLEGO 0.000000
QUARTERSECTION 0.002813
PROPERTY_TYPE 0.000511
PREM_DESCRIPTION 0.000511
NUMBER_OF_CONTAINERS 0.000000
IS_IN_ALLEY 0.000000
LIVINGUNITS 0.000000
SERVICE_AREA 0.172871
GIS_X_COORDINATE 6.165818
GIS_Y_COORDINATE 6.165818
dtype: float64
In [11]:
phoenix_data.dropna(inplace=True) # Drop rows with missing values from the filtered DataFrame
/var/folders/5v/xtgmy_dx10d8y3wvw9_pmzdw0000gn/T/ipykernel_4705/2345750581.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
phoenix_data.dropna(inplace=True) # Drop rows with missing values from the filtered DataFrame

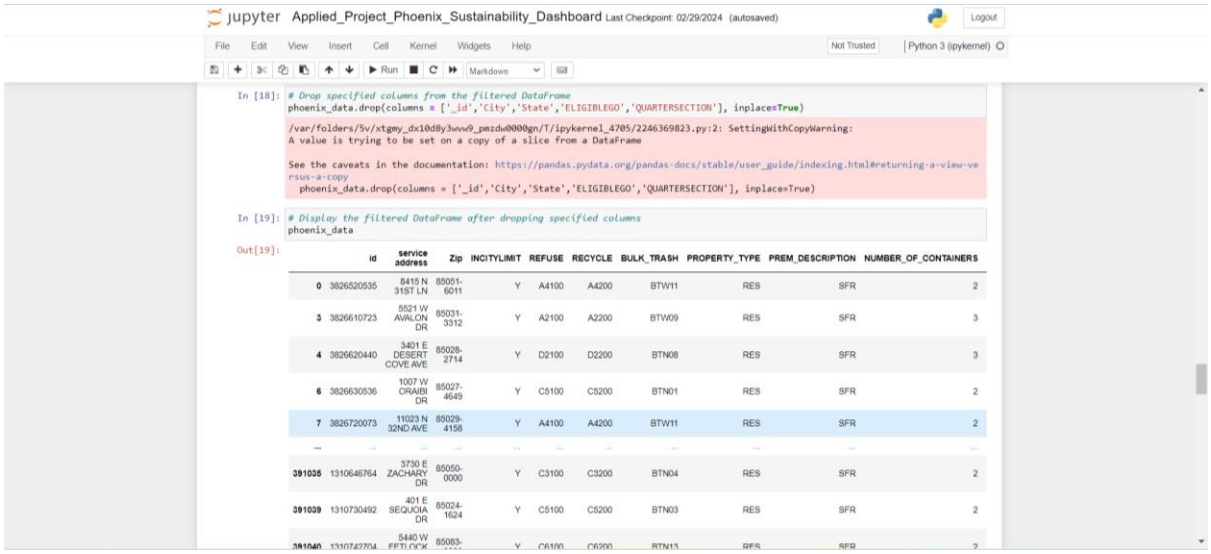

jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [12]:
phoenix_data # Check for missing values again after dropping rows
Out[12]:
293593 rows × 20 columns





SFR: Single Family Residential,

- TFR: Townhouse Residential
- TRFR: Trailer Residential
- TSFR: Two-Story Residential
- APTS: Apartments



jupyter Applied_Project_Phoenix_Sustainability_Dashboard Last Checkpoint: 02/29/2024 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Markdown

In [18]:
```
# Drop specified columns from the filtered DataFrame
phoenix_data.drop(columns = ['_id','City','State','ELIGIBLEGO','QUARTERSECTION'], inplace=True)
```

```
/var/folders/5v/xtgmy_dx10d8y3wvw9_pmzdw0000gn/T/ipykernel_4705/2246369823.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  phoenix_data.drop(columns = ['_id','City','State','ELIGIBLEGO','QUARTERSECTION'], inplace=True)
```

In [19]:
```
# Display the filtered DataFrame after dropping specified columns
phoenix_data
```

Out[19]:

	id	service address	Zip	INCITYLIMIT	REFUSE	RECYCLE	BULK_TRASH	PROPERTY_TYPE	PREM_DESCRIPTION	NUMBER_OF_CONTAINERS
0	3826520535	8415 N 31ST LN	85051-6011	Y	A4100	A4200	BTW11	RES	SFR	2
3	3826610723	5521 W AVALON DR	85031-3312	Y	A2100	A2200	BTW09	RES	SFR	3
4	3826620440	3401 E DESERT COVE AVE	85028-2714	Y	D2100	D2200	BTN08	RES	SFR	3
6	3826630536	1007 W ORAIBI DR	85027-4649	Y	C5100	C5200	BTN01	RES	SFR	2
7	3826720073	11023 N 32ND AVE	85029-4158	Y	A4100	A4200	BTW11	RES	SFR	2
...	...	...	...	...	...	...	...	...	...	...
391035	1310646764	3730 E ZACHARY DR	85050-0000	Y	C3100	C3200	BTN04	RES	SFR	2
391039	1310730492	401 E SEQUOIA DR	85024-1624	Y	C5100	C5200	BTN03	RES	SFR	2
391040	1310742704	5440 W EETLOCK	85083-	Y	C5100	C5200	BTN13	RES	SFR	2

Code for Website:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Green Way: Locate Nearby Landfills & Transfer Stations</title>
<link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css" />
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<style>
    body, html {
        margin: 0;
        padding: 0;
        height: 100%;
    }
    .logo-title-container {
        display: flex;
        align-items: center;
        padding: 10px;
        background-color: #355E3B;
        color: white;
    }
    .logo-title-container img {
        width: 40px; /* Adjust the width of the logo */
        margin-right: 10px; /* Adjust margin as needed */
    }
    h1 {
        margin: 0;
        font-size: 24px;
```

```
    }

    #mapContainer {
        width: 75%;
        height: auto;
        position: absolute;
        top: 0;
        left: 0;
        right: 0;
        bottom: 0;
    }

    #addressInput {
        width: 100%;
        padding: 10px;
        margin-bottom: 10px;
        border: 2px solid green;
        border-radius: 4px;
        box-sizing: border-box;
    }
    .white-box {
        width: 70%; /* Set the width to 70% */
        max-width: 360px; /* Adjust the max-width as needed */
        background-color: white;
        padding: 15px;
        box-sizing: border-box;
        z-index: 999;
        position: absolute;
        top: 0; /* Adjust top position as needed */
```

```
        right: 0; /* Adjust right position as needed */
        border-radius: 0; /* Rounded corners */
        overflow: auto;
    }
    .transfer-details-box {
        display: none;
        margin-top: 5px;
        background-color: #f0f0f0; /* Light grey background color */
        padding: 10px; /* Optional: Add padding to the result box */
        border-radius: 5px;/* Optional: Add rounded corners */
    }
    .arrow-submit,
    .additional-btn {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
    background-color: #355E3B;
    color: white;
    border: none; /* Remove border */
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    box-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2); /* Add box shadow */
    /* Additional styles for h6 appearance */
    font-size: 1em;
    font-weight: bold;
    text-align: center;
    font-family: Vivaldi;
}
```

```
.search-container {
    display: flex;
    flex-direction: column; /* Stack elements vertically */
    align-items: center;
    justify-content: center;

}
.result-box {
    display: none; /* Initially hide the result box */
    margin-top: 5px;
    background-color: #f0f0f0; /* Light grey background color */
    padding: 10px; /* Optional: Add padding to the result box */
    border-radius: 5px; /* Optional: Add rounded corners */
    margin-bottom: 10px;
}
#result-box p {
    margin-bottom: 0;
}
#validationResult {
margin-top: 10px;
margin-bottom: 10px;
display: none; /* Hide the element by default */
}
#enteredAddress,
#stationName {
    color: #355E3B;
    font-weight: bold;
}
.image-title-container {
```

```
    display: flex;
    align-items: center;
    }
    .image-title-container img {
        margin-right: 10px; /* Adjust the margin as needed */
    }
    .station-distance {
    color: grey;
    margin-left: 130px;
    font-weight: lighter;
    }


</style>
</head>
<body>
    <div class="white-box" id="whiteBox">
        <!-- Content inside the white box -->
        <div class="search-container">
            <div class="image-title-container">
                <img src="file:///Users/meghajotangiya/Desktop/Green%20Way-Locate%20Nearby%20Landfills%20&%20Transfer%20Stations/light-green-location-arrow-pin-icon.svg" alt="Image Description">
                <h3 style="font-size: 23px; margin-top: 0; margin-bottom: 5px; color: #355E3B;">Find Drop-Off Location Near You</h3>
            </div>
            <input type="text" id="addressInput" name="addressInput" placeholder="3401 E Desert Cove Ave Phoenix AZ 85028" required>
            <p id="validationResult" style="display: none;"></p>
            <button class="arrow-submit" onclick="validateAddress()">Show Details</button>
        </div>
        <div class="result-box" id="resultBox">
```

```
        <p id="enteredAddress" style="margin-top: 0; font-size: 16px;"></p>
        <p id="serviceArea" style="margin-top: -10px;"></p>
        <p id="collectionSchedule" style="margin-top: -10px; margin-bottom: 0;"></p>
      </div>
      <button class="additional-btn" onclick="showNearestStation()">Landfill/Transfer Station</button>
      <!-- Transfer station details section -->
      <div class="transfer-details-box" id="transferStationDetails">
        <p id="stationName" style="margin-top: 0; font-size: 16px;"></p>
        <p id="stationDistance" style="margin-top: 0; color: grey; margin-left: 30px;"></p>
        <p id="stationAddress" style="margin-top: -10px; "></p>
        <h5 style="font-family: Vivaldi; margin-top: 0; margin-bottom: 0; font-size: 16px; font-weight: bold;">Hours</h5>
        <p id="stationHours" style="margin-top: 0; color: gray; "></p> <!-- Add this line for displaying hours -->
        <h5 style="font-family: Vivaldi; margin-top: 0; margin-bottom: 0; font-size: 16px; font-weight: bold;">Holidays</h5>
        <p id="stationHolidays" style="margin-top: 0; color: gray;"></p>
        <h5 style="font-family: Vivaldi; margin-top: 0; margin-bottom: 0; font-size: 16px; font-weight: bold;">Direction</h5>
        <p id="stationDirection" style="margin-top: 0; margin-bottom: 0; color: gray;"></p>
      </div>
    </div>
    <div id="mapContainer" class="map-container"></div>

  <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
  <script src='https://api.mapbox.com/mapbox-gl-js/v2.6.1/mapbox-gl.js'></script>
  <link href='https://api.mapbox.com/mapbox-gl-js/v2.6.1/mapbox-gl.css' rel='stylesheet' />


  <script>
```

```
// Create new map with marker and tooltip
let map = L.map('mapContainer').setView([33.5987844812527,-112.077087523346], 10);


// Add a tile layer (OpenStreetMap tiles)
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'}).addTo(map);


// Load and add the GeoJSON boundary data to the map
var geojsonUrl = 'http://127.0.0.1:8080/City_Limit_Dark_Outline.geojson'; // Update the URL with your GeoJSON file URL
fetch(geojsonUrl)
  .then(response => response.json())
  .then(data => {
    L.geoJSON(data, {
      style: {
        color: 'green', // Adjust color if needed
        weight: 2, // Adjust line thickness if needed
        fillOpacity: 0,
        fillColor: 'none'

      }
    }).addTo(map);

    // Fit the map to the GeoJSON boundary layer
    map.fitBounds(L.geoJSON(data).getBounds());
  })
  .catch(error => {
    console.error('Error fetching GeoJSON:', error);
  });
```

```
let marker = null;
let stations = [
{
    name: 'North Gateway Station',
    address: '30205 N Black Canyon Hwy \n Phoenix AZ 85085',
    lat: 33.7605925,
    lon: -112.1163813,
    hours: {
        mondayToFriday: '5:30 a.m. to 5 p.m.',
        saturday: '6 a.m. to 3 p.m.',
        sunday: 'Closed'
    },
    holidays: [
        "Martin Luther King Jr. Day",
        "President's Day",
        "Cesar Chavez Day",
        "Memorial Day",
        "Juneteenth",
        "Independence Day",
        "Labor Day",
        "Indigenous Peoples' Day",
        "Veterans Day (observed by the City on November 10, 2023)",
        "Thanksgiving and the day after",
        "Christmas",
        "New Year's",
        "Christmas Eve - Open 6 a.m. - 12 p.m.",
        "Regular hours on New Year's Eve"
    ]
```

```
  },
  {
    name: '27th Ave Station',
    address: '3060 S 27th Ave \n Phoenix AZ 85009',
    lat: 33.4169941,
    lon: -112.1186168,
    hours: {
      mondayToFriday: '5:30 a.m. to 5 p.m.',
      saturday: '6 a.m. to 3 p.m.',
      sunday: 'Closed'
    },
    holidays: [
      "Martin Luther King Jr. Day",
      "President's Day",
      "Cesar Chavez Day",
      "Memorial Day",
      "Juneteenth",
      "Independence Day",
      "Labor Day",
      "Indigenous Peoples' Day",
      "Veterans Day (observed by the City on November 10, 2023)",
      "Thanksgiving and the day after",
      "Christmas",
      "New Year's",
      "Christmas Eve - Open 6 a.m. - 12 p.m.",
      "Regular hours on New Year's Eve"
    ]
  },
  {
```

```
        name: '85 Landfill',
        address: '28361 W Patterson Road \n Buckeye AZ 85326',
        lat: 33.1890375,
        lon: -112.6686124,
        hours: {
            mondayToFriday: '5:30 a.m. to 5 p.m.',
            saturday: '6 a.m. to 3 p.m.',
            sunday: 'Closed'
        },
        holidays: [
            "Martin Luther King Jr. Day",
            "President's Day",
            "Cesar Chavez Day",
            "Memorial Day",
            "Juneteenth",
            "Independence Day",
            "Labor Day",
            "Indigenous Peoples' Day",
            "Veterans Day (observed by the City on November 10, 2023)",
            "Thanksgiving and the day after",
            "Christmas",
            "New Year's",
            "Christmas Eve - Open 6 a.m. - 12 p.m.",
            "Regular hours on New Year's Eve"
        ]
    }
];
async function validateAddress() {
```

```
const addressInput = document.getElementById('addressInput').value.trim().toLowerCase();

try {
    const response = await fetch('http://127.0.0.1:8080/active-service-addresses.csv');
    if (!response.ok) {
        throw new Error('Failed to fetch CSV data.');
    }
    const csvData = await response.text();
    const rows = csvData.split('\n');
    const headers = rows[0].split(',').map(header => header.trim()); // Trim whitespace from headers
    const addressIndex = headers.indexOf('Address');
    const serviceAreaIndex = headers.indexOf('Service_Area');

    let isValid = false;
    let validatedAddress = '';
    let serviceArea = '';

    for (let i = 1; i < rows.length; i++) {
        const rowData = rows[i].split(',').map(item => item.trim()); // Trim whitespace from row data
        const address = rowData[addressIndex].toLowerCase();

        if (address === addressInput) {
            isValid = true;
            validatedAddress = address;
            serviceArea = rowData[serviceAreaIndex];

            document.getElementById('resultBox').style.display = 'block'; // Show the result box
```

```
const validationResult = document.getElementById('validationResult');
const enteredAddress = document.getElementById('enteredAddress');


enteredAddress.innerText = rowData[addressIndex]; // Display entered address
document.getElementById('serviceArea').innerHTML = 'Service Area: ' + serviceArea; // Display service area
document.getElementById('collectionSchedule').innerHTML = '<a href="https://www.phoenix.gov/publicworkssite/Documents/2024%20Bulk%20Trash%20Collection%20Schedule_English.pdf" target="_blank">Bulk-Trash Collection Day</a>';



// Fetch latitude and longitude using OpenStreetMap API
const osmResponse = await fetch(https://nominatim.openstreetmap.org/search?format=json&q=${validatedAddress});
if (!osmResponse.ok) {
  throw new Error('Failed to fetch latitude and longitude.');
}
const osmData = await osmResponse.json();
const { lat, lon } = osmData[0];


// Define the custom icon using L.icon
const customIcon = L.icon({
  iconUrl: 'file:///Users/meghajotangiya/Desktop/Green%20Way-Locate%20Nearby%20Landfills%20&%20Transfer%20Stations/location-pin-user-dot-icon.svg', // Replace 'path_to_your_custom_icon.svg' with the actual path or URL to your SVG icon
  iconSize: [35, 35], // Adjust the size of the icon as needed
  iconAnchor: [16, 32] // Adjust the anchor point if necessary
});


if (map) {
```

```
            map.remove();
        }

        map = L.map('mapContainer').setView([33.5987844812527,-112.077087523346], 13);
        L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'}).addTo(map);
        marker = L.marker([lat, lon], { icon: customIcon }).addTo(map);
        marker.bindTooltip('<strong>Your Address: </strong>'+rowData[addressIndex]).openTooltip();

        break;
      }
    }

    if (!isValid) {
      const validationResult = document.getElementById('validationResult');
      validationResult.style.display = 'block';
      validationResult.innerText = 'We could not find this address.';
      validationResult.style.color = 'red';
    }
  } catch (error) {
    console.error('Error:', error);
  }
}

async function showNearestStation() {
  const addressInput = document.getElementById('addressInput').value.trim().toLowerCase(); // Convert input to lowercase for case-insensitive comparison
```

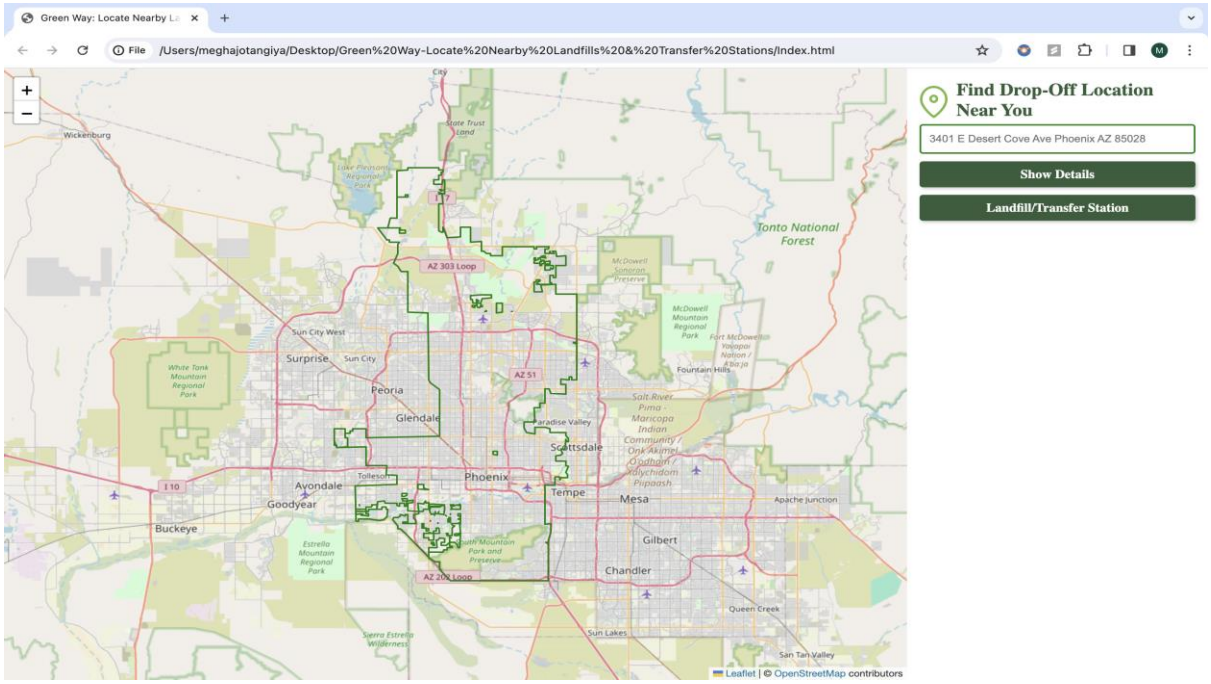
```
    try {
        // Fetch latitude and longitude using OpenStreetMap API
        const osmResponse = await fetch(https://nominatim.openstreetmap.org/search?format=json&q=${addressInput});
        if (!osmResponse.ok) {
            throw new Error('Failed to fetch latitude and longitude.');
        }
        const osmData = await osmResponse.json();
        const { lat, lon } = osmData[0];

        // Calculate distance to each station using Mapbox API
        const distances = await Promise.all(stations.map(async (station) => {
            const mapboxResponse = await fetch(https://api.mapbox.com/directions/v5/mapbox/driving/${lon},${lat};${station.lon},${station.lat}?access_token=##########################);
            if (!mapboxResponse.ok) {
                throw new Error('Failed to calculate distance.');
            }
            const mapboxData = await mapboxResponse.json();
            return mapboxData.routes[0].distance;
        }));

        // Find the station with the shortest distance
        const nearestIndex = distances.indexOf(Math.min(...distances));
        const nearestStation = stations[nearestIndex];

        // Create new map only if it's not already initialized
        if (!map) {
            map = L.map('mapContainer').setView([33.5987844812527,-112.077087523346], 13);
```

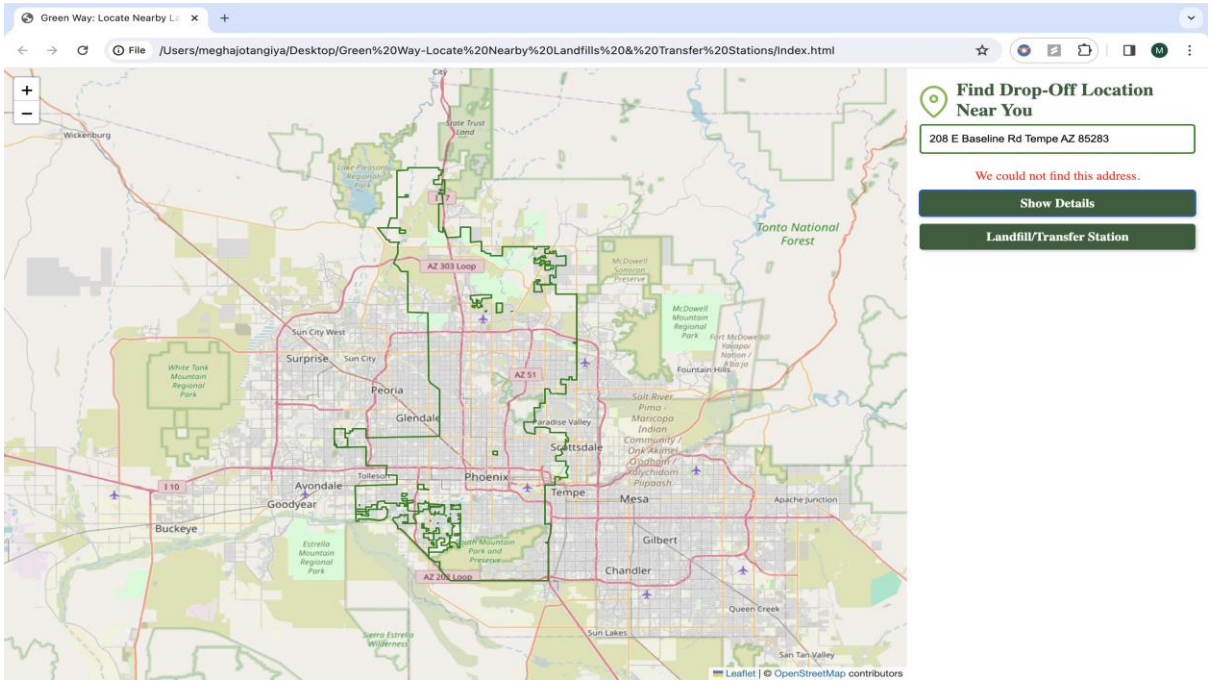
```
        L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
            attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
        }).addTo(map);
    }
    // Display the nearest station on the map
    if (nearestStation) {
        // Fetch route geometry using Mapbox Directions API and display on the map
        const routeResponse = await fetch(https://api.mapbox.com/directions/v5/mapbox/driving/${lon},${lat};${nearestStation.lon},${nearestStation.lat}?geometries=geojson&steps=true&access_token=##########################);
        const routeData = await routeResponse.json();
        const route = routeData.routes[0].geometry;

        // Add the route to the map
        L.geoJSON(route).addTo(map);

        // Add marker for the nearest station
        const stationMarker = L.marker([nearestStation.lat, nearestStation.lon]).addTo(map);
        stationMarker.bindTooltip('<strong>Name: </strong>'+nearestStation.name + '<br>' + '<strong>Station Address: </strong>'+nearestStation.address).openTooltip();

        // Convert distance from meters to miles
        const distanceInMiles = distances[nearestIndex] * 0.000621371;

        // Update transfer station details on the web page with distance in miles
        document.getElementById('transferStationDetails').style.display = 'block';
        document.getElementById('stationName').innerHTML = `
            <span>${nearestStation.name}</span>
            <span class="station-distance">${distanceInMiles.toFixed(2)}mi</span>
```

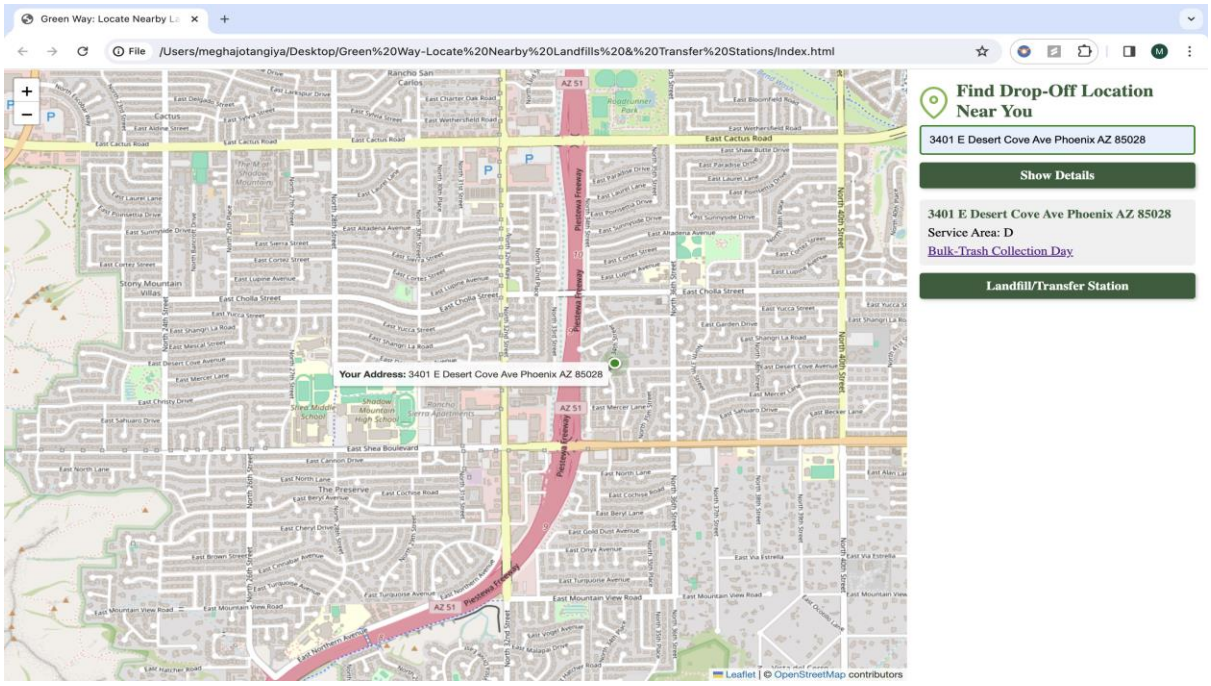
```
        `;
        document.getElementById('stationAddress').innerHTML = nearestStation.address ;
        document.getElementById('stationHours').innerHTML = 'Monday-Friday: ' + nearestStation.hours.mondayToFriday + '<br>' +
                                    'Saturday: ' + nearestStation.hours.saturday + '<br>' +
                                    'Sunday: ' + nearestStation.hours.sunday;
        document.getElementById('stationHolidays').innerHTML = nearestStation.holidays.join('<br>');
        document.getElementById('stationDistance').innerHTML = '';

        stationMarker.setIcon(L.icon({
          iconUrl: 'file:///Users/meghajotangiya/Desktop/Green%20Way-Locate%20Nearby%20Landfills%20&%20Transfer%20Stations/location-pin-trash.svg', // Provide the path to your custom marker image
          iconSize: [35, 35],
          iconAnchor: [16, 32]
        }));
        // Fetch direction using Mapbox Directions API and display it
        const directionResponse = await fetch(https://api.mapbox.com/directions/v5/mapbox/driving/${lon},${lat};${nearestStation.lon},${nearestStation.lat}?steps=true&access_token=####################);
        const directionData = await directionResponse.json();
        const steps = directionData.routes[0].legs[0].steps;
        const directions = steps.map(step => step.maneuver.instruction);

        // Update transfer station details on the web page with directions
        document.getElementById('stationDirection').innerHTML =  directions.join('<br>');
      }
    } catch (error) {
      console.error('Error:', error);
    }
```

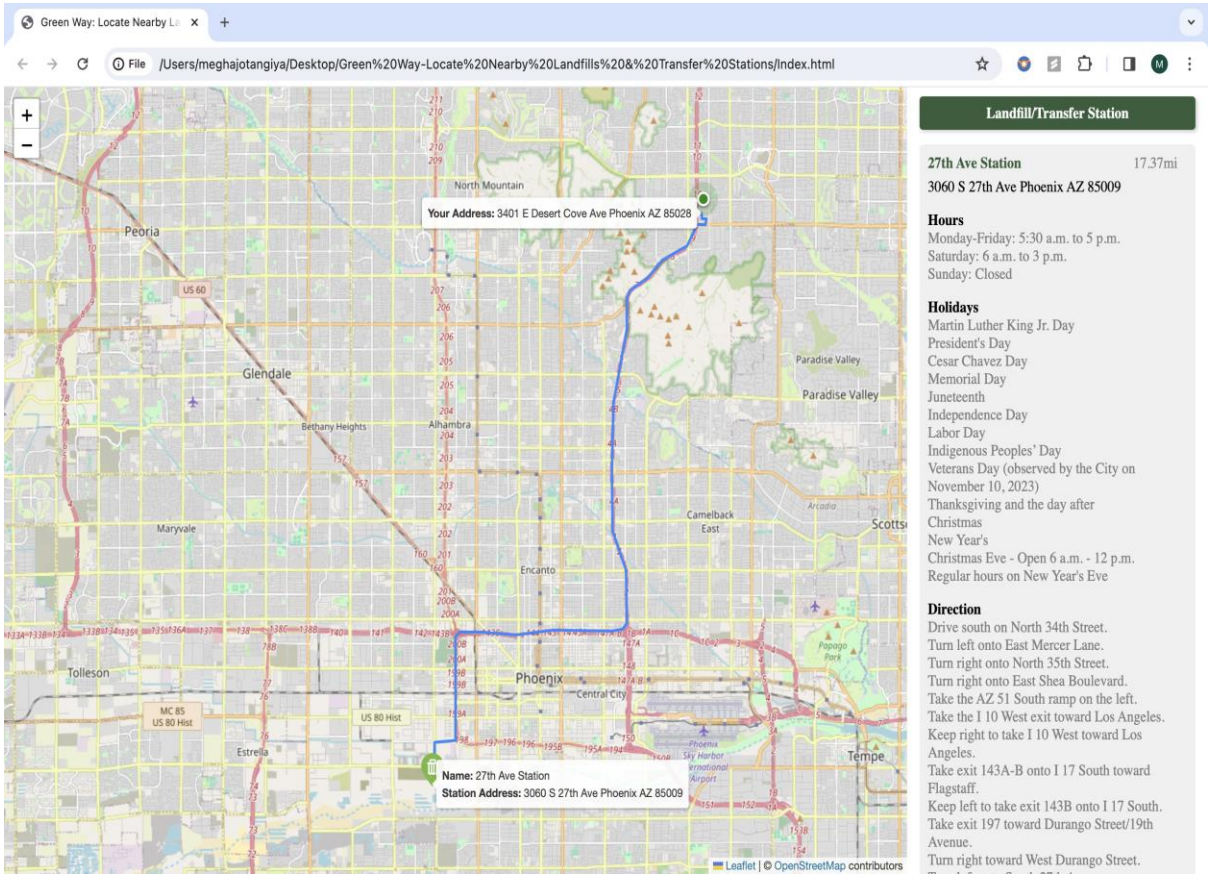
```
}
 // Function to update map container height based on white box height
 function updateMapContainerHeight() {
        const whiteBox = document.getElementById('whiteBox');
        const whiteBoxHeight = whiteBox.offsetHeight;
        const transferDetailsBox = document.getElementById('transferStationDetails');
        const transferDetailsHeight = transferDetailsBox.offsetHeight;
        const webpageHeight = window.innerHeight;
        const mapContainer = document.getElementById('mapContainer');

        // Calculate total height of white box including transfer station details
        const totalWhiteBoxHeight = whiteBoxHeight + transferDetailsHeight;

        // Check if total white box height exceeds webpage height
        if (totalWhiteBoxHeight > webpageHeight) {
            // Set the map container's height to match the total white box height
            mapContainer.style.height = ${totalWhiteBoxHeight}px;
        } else {
            // Set default height for the map container
            mapContainer.style.height = 'auto';
        }
    }
    // Call the updateMapContainerHeight function when the window is resized or when the white box content changes
    window.addEventListener('resize', updateMapContainerHeight);
    document.addEventListener('DOMContentLoaded', updateMapContainerHeight); // Call when the DOM content is loaded
</script>
</body>
</html>
```

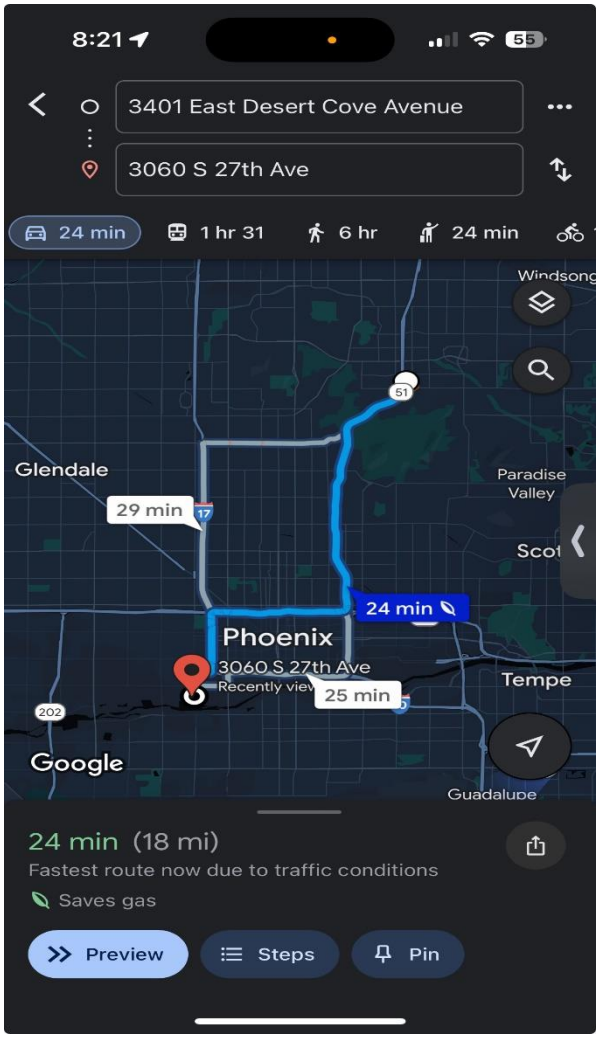
Screenshots:





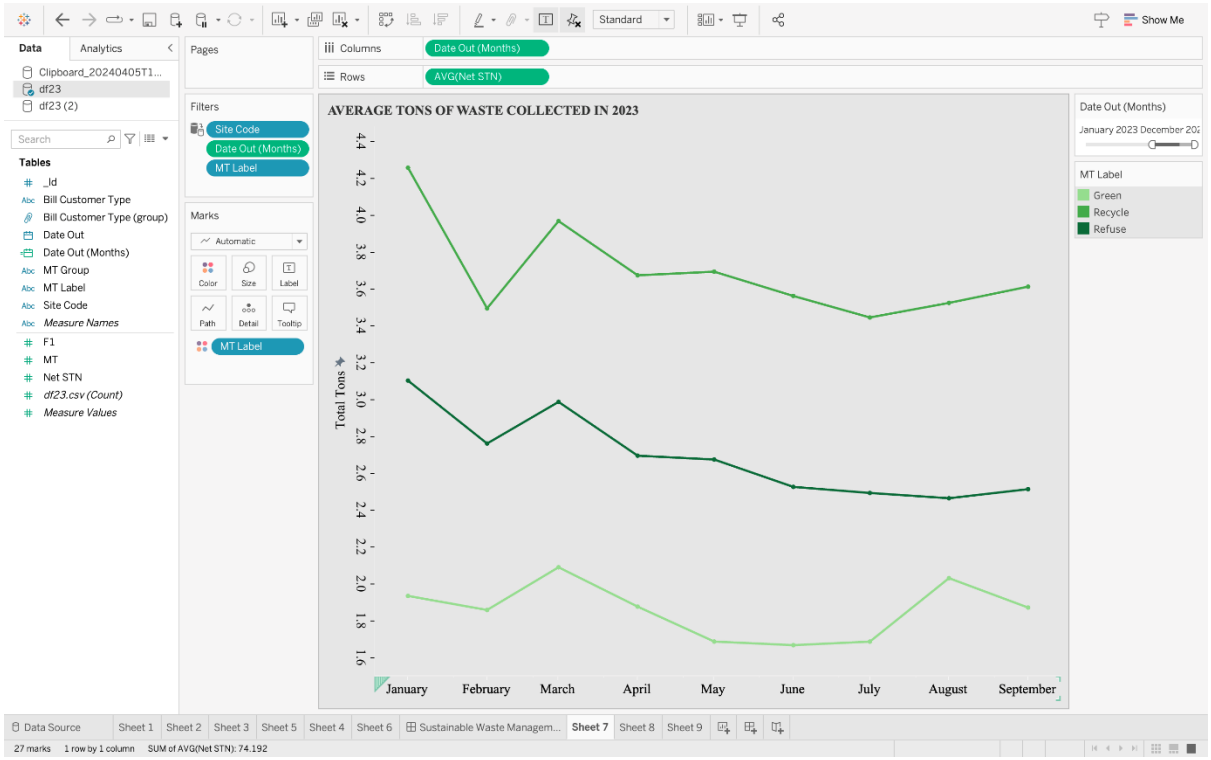

Green Way: Locate Nearby La
File
/Users/meghajotangiya/Desktop/Green%20Way-Locate%20Nearby%20Landfills%20&%20Transfer%20Stations/Index.html
Find Drop-Off Location Near You
3401 E Desert Cove Ave Phoenix AZ 85028
Show Details
3401 E Desert Cove Ave Phoenix AZ 85028
Service Area: D
Bulk-Trash Collection Day
Landfill/Transfer Station
Your Address: 3401 E Desert Cove Ave Phoenix AZ 85028
Leaflet | © OpenStreetMap contributors


Green Way: Locate Nearby La
File
/Users/meghajotangiya/Desktop/Green%20Way-Locate%20Nearby%20Landfills%20&%20Transfer%20Stations/Index.html
Landfill/Transfer Station
27th Ave Station
17.37mi
3060 S 27th Ave Phoenix AZ 85009
Hours
Monday-Friday: 5:30 a.m. to 5 p.m.
Saturday: 6 a.m. to 3 p.m.
Sunday: Closed
Holidays
Martin Luther King Jr. Day
President's Day
Cesar Chavez Day
Memorial Day
Juneteenth
Independence Day
Labor Day
Indigenous Peoples' Day
Veterans Day (observed by the City on November 10, 2023)
Thanksgiving and the day after Christmas
New Year's
Christmas Eve - Open 6 a.m. - 12 p.m.
Regular hours on New Year's Eve
Direction
Drive south on North 34th Street.
Turn left onto East Mercer Lane.
Turn right onto North 35th Street.
Turn right onto East Shea Boulevard.
Take the AZ 51 South ramp on the left.
Take the I 10 West exit toward Los Angeles.
Keep right to take I 10 West toward Los Angeles.
Take exit 143A-B onto I 17 South toward Flagstaff.
Keep left to take exit 143B onto I 17 South.
Take exit 197 toward Durango Street/19th Avenue.
Turn right toward West Durango Street.
Your Address: 3401 E Desert Cove Ave Phoenix AZ 85028
Name: 27th Ave Station
Station Address: 3060 S 27th Ave Phoenix AZ 85009
Leaflet | © OpenStreetMap contributors

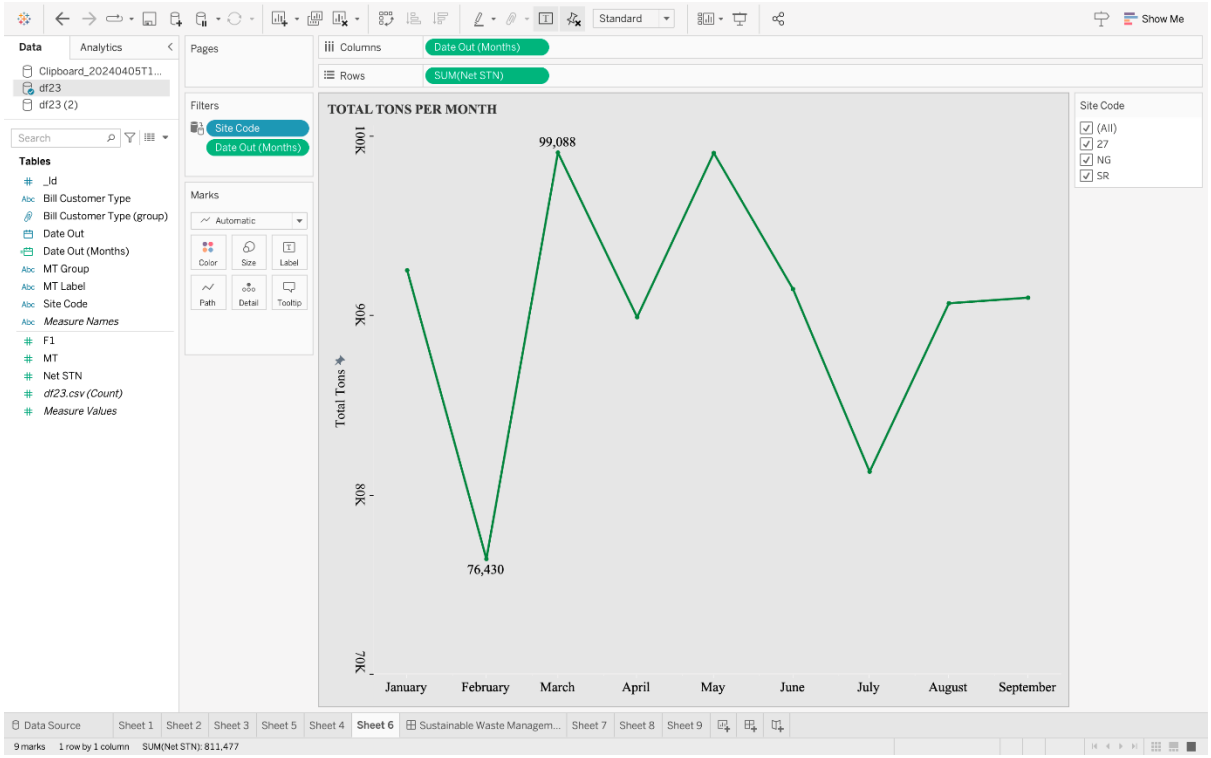


The Google Maps screenshot & the webpage, both display the distance from one location to the other. As we can see, the accuracy of both of these functions is almost similar with same routes. Also, the website displays public holidays on which the station might be closed.

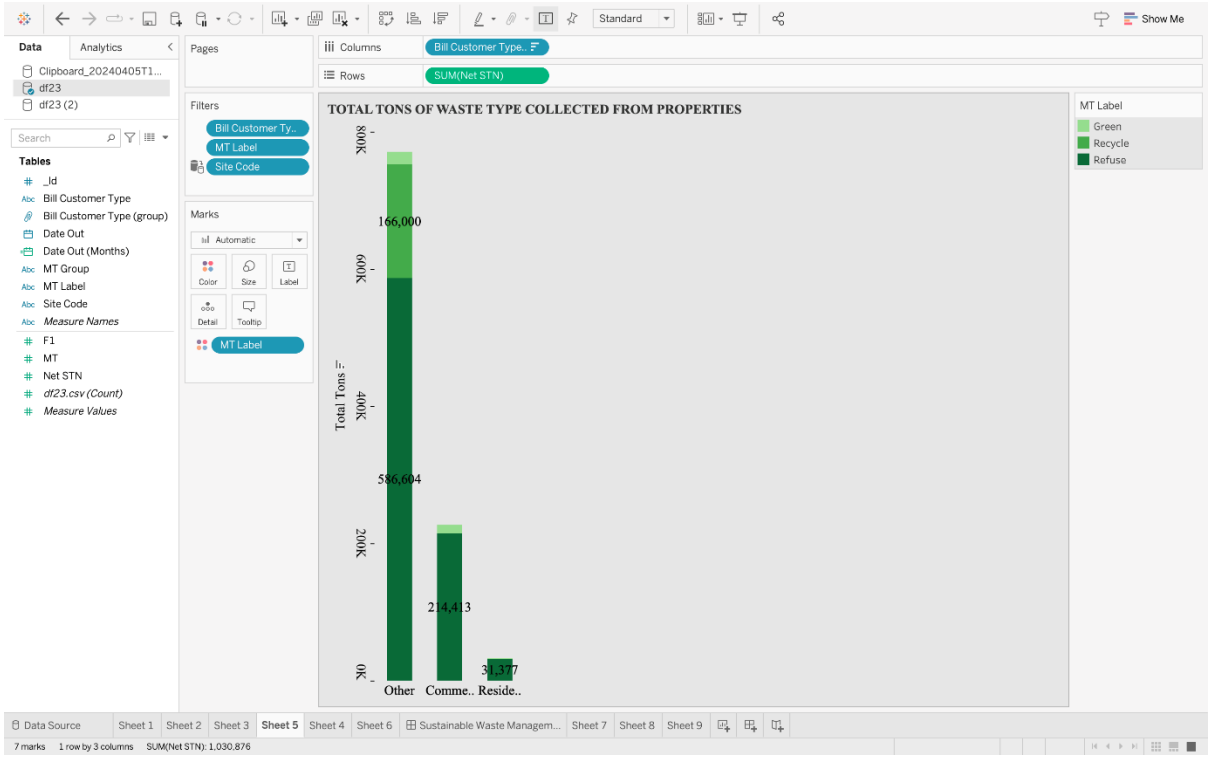
Tableau Visualization:



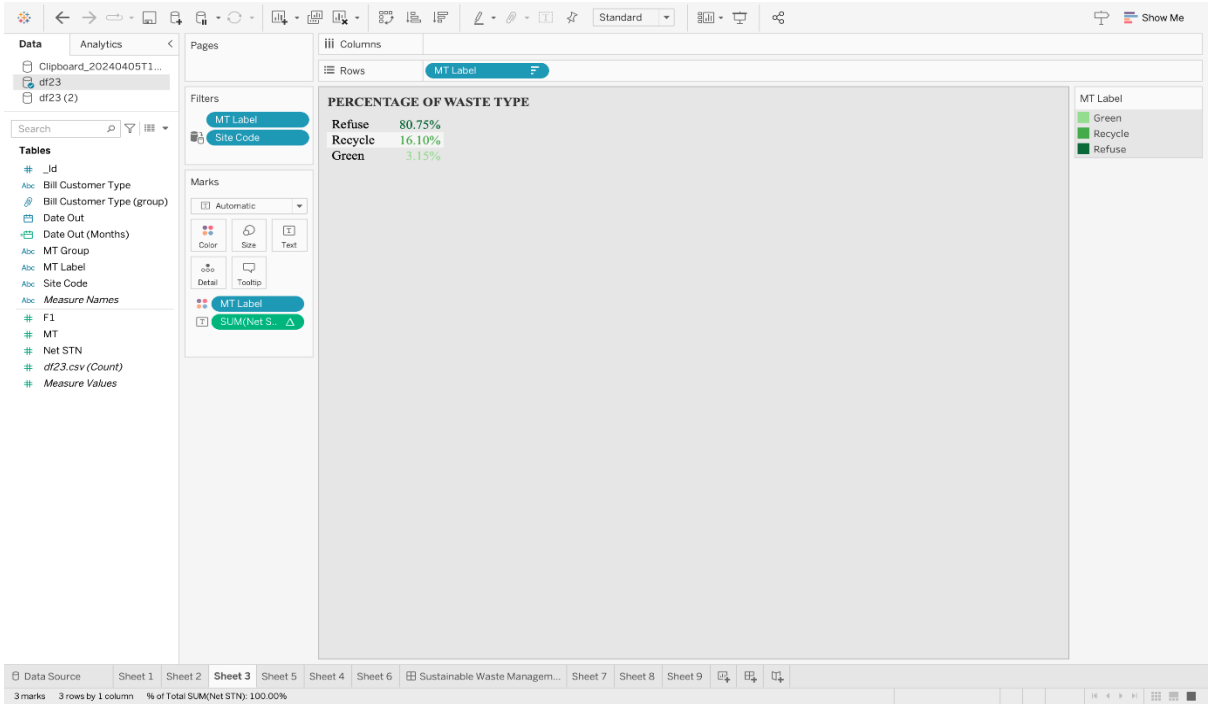
The presented line chart offers a comprehensive analysis of the average monthly waste collection volumes categorized by waste type - Recycle, Refuse, and Green - within Phoenix City. Through a meticulous aggregation and categorization of collection data, the visualization provides valuable insights into the distribution and trends of waste collection across different months. By segmenting the data according to waste type, it enables a nuanced understanding of the relative contributions of Recycle, Refuse, and Green waste to the overall waste stream on a monthly basis. This detailed analysis facilitates the identification of patterns, anomalies, and potential areas for improvement in waste management practices. By visualizing the average tons of waste collected per month, categorized by waste type. This comprehensive portrayal of waste collection dynamics serves as a valuable resource for urban planners, waste management authorities, and policymakers, aiding in informed decision-making and strategic planning aimed at fostering a more efficient and environmentally sustainable waste management infrastructure within the city



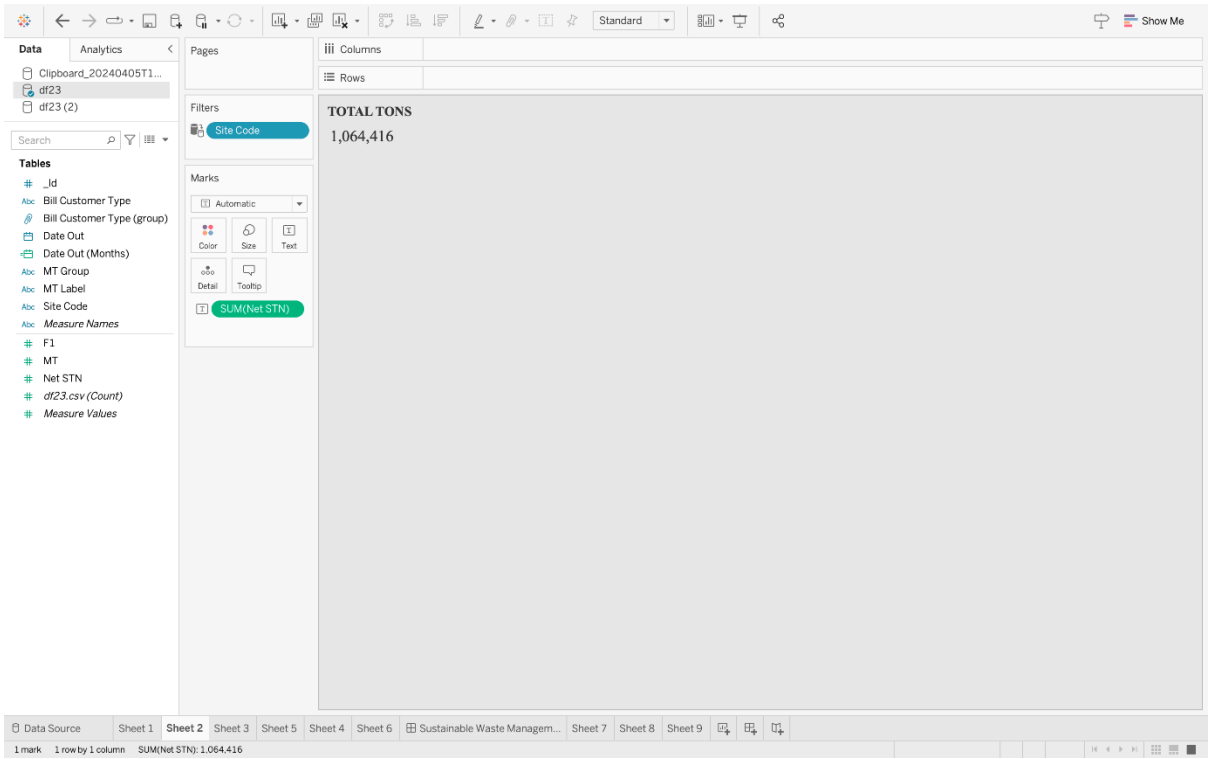
The line chart provides a temporal overview of monthly waste collection trends in Phoenix City, showcasing fluctuations in tonnage across different months. March emerges as the peak month, with a total collection of 99,000 tons, reflecting potentially heightened seasonal waste generation or specific events impacting waste volumes. Conversely, February records the lowest collection at 76,000 tons, suggesting potential factors such as reduced consumption or waste generation during this period. Analyzing these variations can inform waste management strategies, resource allocation, and operational planning to optimize collection efficiency and ensure timely waste disposal services. Understanding monthly waste generation patterns aids in maintaining effective waste management practices and promoting environmental sustainability in Phoenix City.



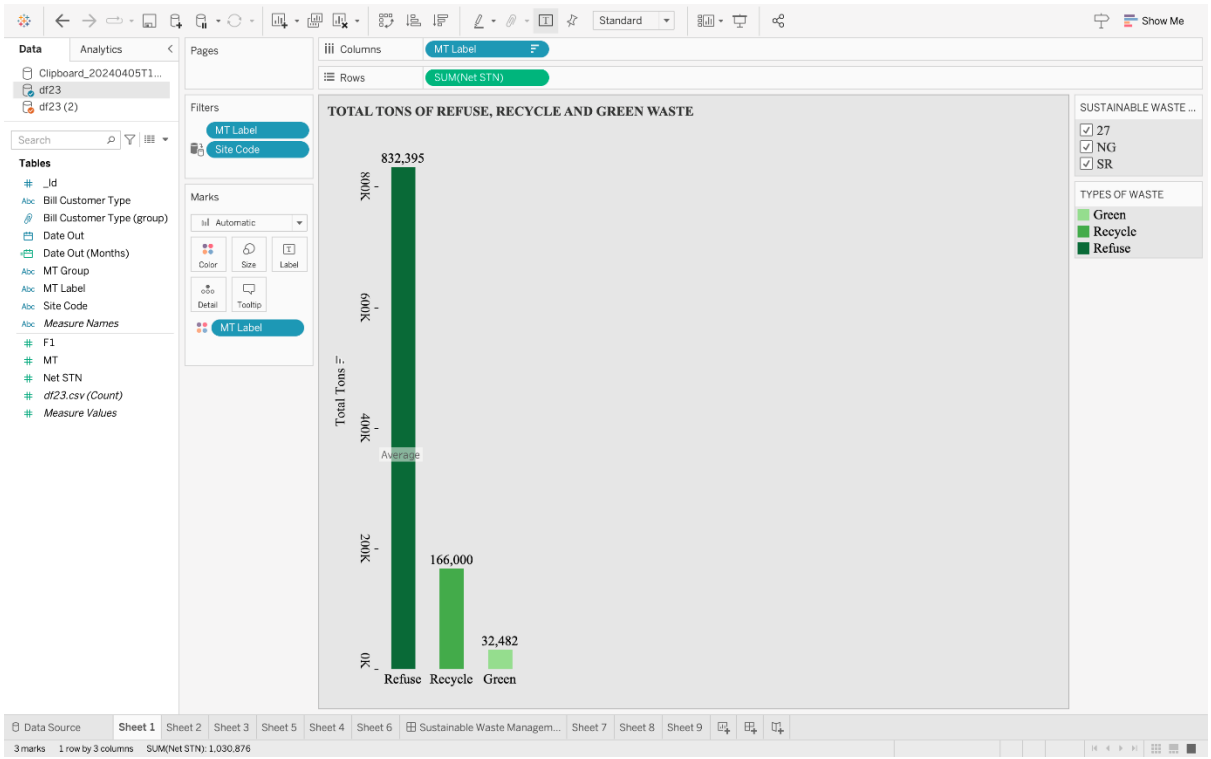
The categorical bar chart underscores the waste management dynamics across various property types in Phoenix City, with refuse waste dominating each category. This signifies the substantial volume of non-recyclable materials generated across residential, commercial, and industrial sectors. Addressing the management of refuse waste remains a critical challenge, necessitating comprehensive strategies for waste diversion and resource recovery. While specific tonnages for recycle and green waste are not delineated, their inclusion underscores the city's multifaceted approach to sustainable waste management. Promoting recycling initiatives and implementing green waste diversion programs are essential steps towards mitigating environmental impacts and fostering a circular economy ethos within Phoenix City.



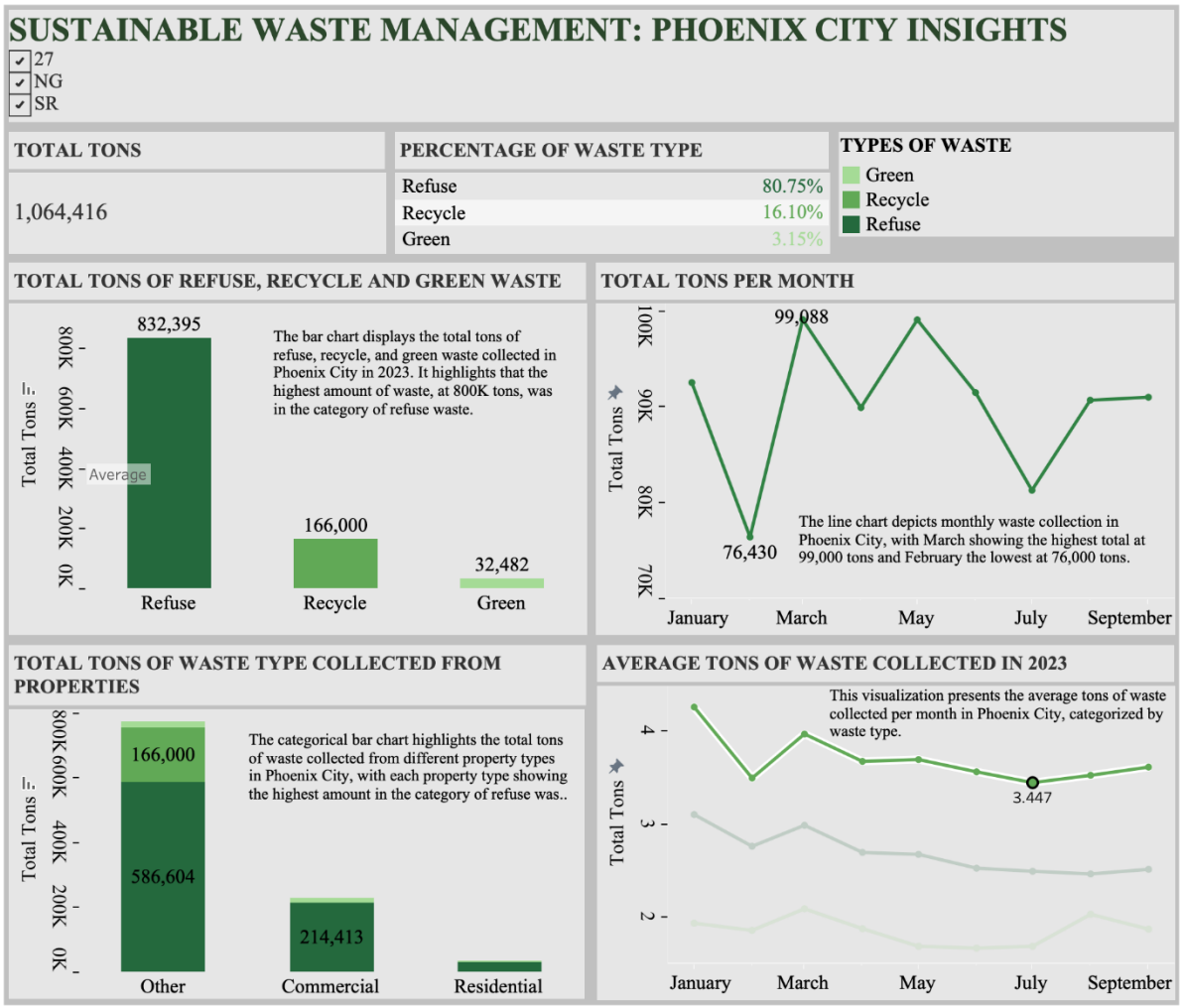
This graph displays the Percentage of Waste Type with Refuse being the highest type of waste collected with 80.75%, followed by Recycle (16.10) and Green (3.15%)



The total tons of waste collected till 2023 is 1,064,416.



The bar chart presents a comprehensive overview of waste management in Phoenix City for the year 2023, showcasing the total tons of refuse, recycle, and green waste collected. Notably, the data underscores the predominant volume of refuse waste, towering at an impressive 800,000 tons, marking it as the most substantial category among the three. This visualization provides crucial insights into the distribution of waste types, crucial for informed decision-making and sustainable urban planning.



As the culmination of our project, we are pleased to present our comprehensive dashboard, offering a detailed analysis of waste management dynamics in Phoenix City for the year 2023. Our dashboard encompasses a series of visualizations meticulously crafted to provide stakeholders with actionable insights into the city's waste landscape.

The first visualization lays the groundwork by presenting the total tons of waste collected, providing a foundational understanding of the scale of waste management efforts throughout the year.

In the second visualization, we delve deeper into the composition of waste types - Refuse, Recycle, and Green - presenting their respective percentages of contribution to the overall waste stream. This comparative analysis sheds light on the relative significance of each waste category in the broader context of waste management.

Our third visualization, a bar chart, highlights the total tons of refuse, recycle, and green waste collected, with a striking emphasis on the dominance of refuse waste, totaling 800,000 tons. This finding underscores the critical importance of targeted interventions to manage non-recyclable waste effectively.

Moving forward, our fifth visualization, a categorical bar chart, unveils the distribution of waste across different property types, revealing a consistent trend of refuse waste being the most prevalent category across diverse sectors.

The sixth and seventh visualizations, represented as line charts, offer insights into temporal trends in waste collection. While the former highlights monthly fluctuations in collection volumes, the latter presents the average tons of waste collected per month, categorized by waste type.

Together, these visualizations form a comprehensive narrative, empowering with the data-driven insights necessary for informed decision-making and strategic planning in the realm of sustainable waste management. We believe that this project contributes significantly to the ongoing discourse on environmental sustainability and underscores the imperative for concerted action in addressing contemporary waste challenges.

Insights & Actions:

Insights:

1. **Operational Efficiency**:

   By analyzing the total tons collected at landfills and transfer stations in 2023, you can assess the operational efficiency of these waste processing facilities. Identifying peak periods of waste collection can help in resource allocation and scheduling.

2. **Waste Composition:**

   The percentage table showing the breakdown of waste types (refuse, recycle, and green waste) gives insights into the city's waste composition. This information is valuable for designing recycling programs and promoting green initiatives.

3. **Waste Management Trends:**

   The bar chart depicting the total tons of refuse, recycle, and green waste collected provides a comparative view of waste management trends. Identifying trends can aid in evaluating the effectiveness of waste reduction strategies.

4. **Monthly Variations:**

   The line chart displaying total tons per month allows you to identify seasonal variations in waste generation and collection. This insight is useful for planning and adjusting waste management operations accordingly.

5. **Residential Contribution:**

   Analyzing the total tons of waste type collected from properties helps in understanding the contribution of residential areas to overall waste generation. It can guide targeted outreach programs to encourage responsible waste disposal practices among residents.

6. **Monthly Average:**

   Calculating the average tons of waste collected each month provides a benchmark for assessing the consistency of waste management efforts. Deviations from the average can highlight areas that require attention or improvement.

Actions:

1. **Operational Efficiency Improvement:**
   - Implement optimized scheduling during peak waste collection periods at landfills and transfer stations to improve operational efficiency.
   - Invest in technology and infrastructure upgrades to streamline waste processing and reduce bottlenecks.

2. **Waste Composition Management:**
   - Develop targeted recycling programs based on the waste composition data to increase recycling rates and reduce landfill waste.
   - Promote green initiatives such as composting for organic waste to divert it from landfills and encourage sustainable waste disposal practices.

3. **Effective Waste Reduction Strategies:**
   - Analyze waste management trends to identify areas where waste reduction strategies can be most effective, such as reducing refuse or increasing recycling of specific materials.
   - Collaborate with local businesses and industries to implement waste minimization practices and reduce waste generation at the source.

4. **Seasonal Variation Planning:**
   - Use insights from seasonal waste variations to optimize waste collection routes, adjust staffing levels, and allocate resources efficiently during peak periods.

5. **Community Engagement and Education:**
    - Launch targeted outreach programs based on residential waste contributions to educate residents about proper waste sorting, recycling techniques, and the importance of waste reduction.
    - Organize community clean-up events, recycling drives, and educational workshops to foster a culture of sustainability and environmental responsibility.

6. **Continuous Monitoring and Improvement:**
    - Regularly monitor and evaluate waste management efforts using metrics such as monthly average waste collection to identify areas for improvement.
    - Implement feedback mechanisms and gather input from stakeholders to adapt strategies, address challenges, and achieve continuous improvement in waste management practices.

Tableau Public Link:

https://public.tableau.com/app/profile/vrutik.sanjay.adani/viz/SustainableWasteManagement-PhoenixCityInsights_171289461