Syringe Surveillance: Understanding Syringe Data in NYC

Nahallah Champagne

CS301 Introduction to Data Science

Project 1

Report:

https://docs.google.com/document/d/1X4128rEwovh6mNZ9mDBYtfqzHV2j-JR1CtIPbi3BDro/edit

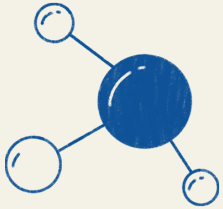
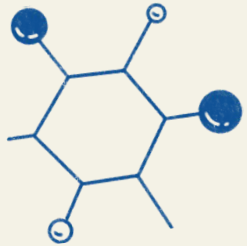


Table of contents

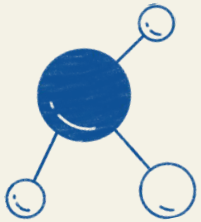


01

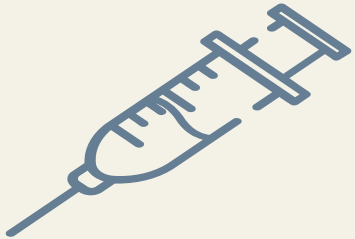
Overview

What is my project all about? Why is it important and exciting?!

Introduction



Syringe litter is a concern for all New Yorkers. The City is working to clean up syringe litter and to educate community members about how to get rid of syringes safely and prevent needlestick injury. Data gathered from NYC Open Data in creating this report, that aims to give a detailed overview on that data gathered from 2017-2023



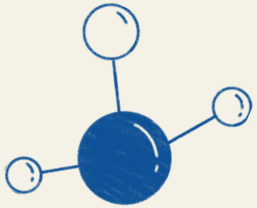
02

Data Collection and Preprocessing

Processing the data, and cleaning it up to better fit my project.

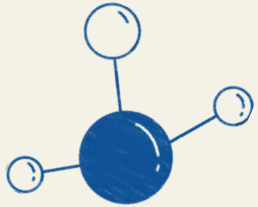
Data Collection and Preprocessing

- NYC Open Data: Summary of Syringe Data in NYC Parks
- Dataset had 23 columns and 30K rows



```
#-------------------------Data Collection and Preprocessing-------------------------------------#
import pandas as pd
import numpy as np

#Load dataset into DataFrame
syringe_df = pd.read_csv('Summary_of_Syringe_Data_in_NYC_Parks_20240306.csv')

#Drop these columns from the Syringe Dataset. Since most of the cells in these columns are empty.
syringe_df = syringe_df.drop('kiosk_number', axis=1)
syringe_df = syringe_df.drop('kiosk_syringes', axis=1)
syringe_df = syringe_df.drop('kiosk_type', axis=1)
syringe_df = syringe_df.drop('kiosk_site', axis=1)

#I want to stort the data in some way, sorting it in decsending order by year seems like a good idea.
syringe_df = syringe_df.sort_values(by='collected_date', ascending=False)

#Display the first 5 rows of the specific rows
syringe_df.head()
```

Data Collection and Preprocessing



2]:

collected_date	time_of_day	year	month	month_text	week	group	location	ground_syringes	total_syringes	precinct	borough	district	property_type	createc
12/31/2023 12:00:00 AM	AM	2023	12	Dec	53	Parks	Aqueduct Walk	135.0	135.0	52.0	Bronx	X-07	ZONE	2024· 06:17:33.00
12/31/2023 12:00:00 AM	AM	2023	12	Dec	53	Parks	St. James Park	78.0	78.0	52.0	Bronx	X-07	PARK	2024· 06:19:43.00
12/31/2023 12:00:00 AM	AM	2023	12	Dec	53	BBP	Beanstalk Playground	5.0	5.0	46.0	Bronx	X-05	PARK	2023· 11:36:37.00
12/31/2023 12:00:00 AM	AM	2023	12	Dec	53	BBP	Slattery Playground	3.0	3.0	46.0	Bronx	X-05	PARK	2023· 11:34:29.00
12/31/2023 12:00:00 AM	AM	2023	12	Dec	53	Parks	People's Park	2.0	2.0	40.0	Bronx	X-01	PARK	2024· 10:55:07.00

03

Data Visualization

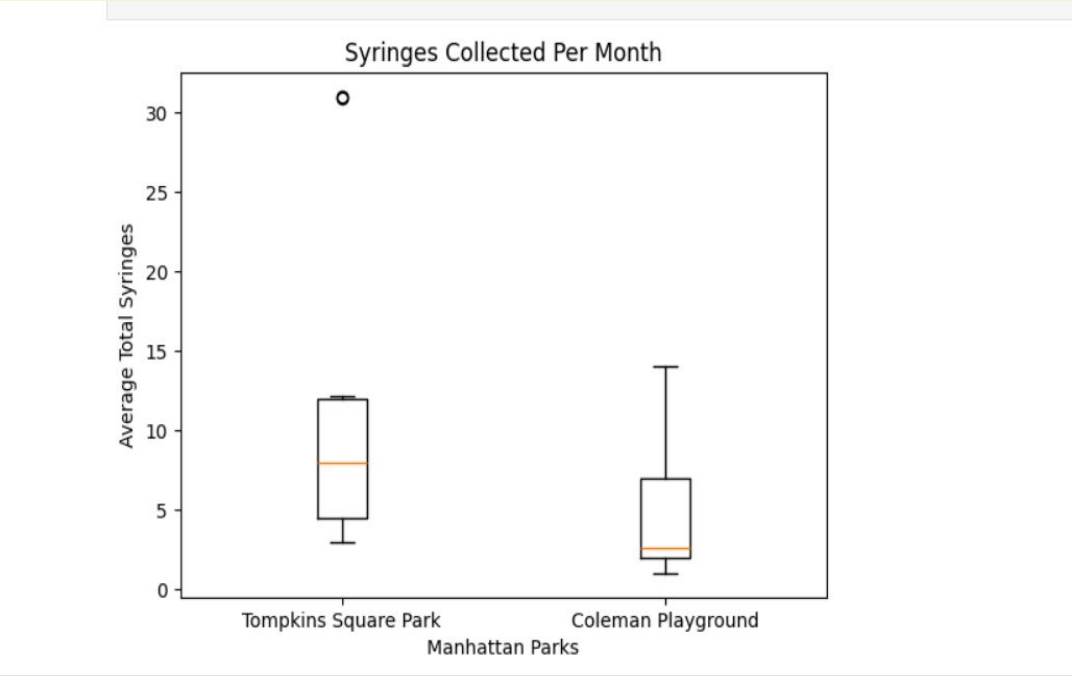
Different Data Visualization techniques I used to represent and illustrate the data.

Data Visualization - Box Plot

[72]:

```
#-------------------------Data Visualization-------------------------------------#
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#------Box Plot-------

#Find what is the average amount of syringes that are found each month. Reset the index from the way it was before
avg_syringes_per_month_park = syringe_df.groupby(['month', 'location'])['total_syringes'].mean().reset_index()

# Box plot data
tompkinsSquare_park_data = avg_syringes_per_month_park[avg_syringes_per_month_park['location'] == 'Tompkins Square Park']['total_syringes']
coleman_playground_data = avg_syringes_per_month_park[avg_syringes_per_month_park['location'] == 'Coleman Playground']['total_syringes']

# Plot data in box chart
fig, ax = plt.subplots()
ax.set_title('Syringes Collected Per Month')
ax.set_xlabel("Manhattan Parks")
ax.set_ylabel("Average Total Syringes")
ax.boxplot([tompkinsSquare_park_data, coleman_playground_data],
           labels=['Tompkins Square Park', 'Coleman Playground'])

# Show plot
plt.show()
#Explain why the selected visualizations are appropriate for your analysis.
'''
I chose a box plot to showcase w=the median numbers of syringes found in these
two parks in Manhattan. A box plot allowed me to compare the median of syringe
collections in different parks. You can easily see which parks tend to have
higher or lower median numbers of syringes collected.
'''
```

Data Visualization - Box Plot

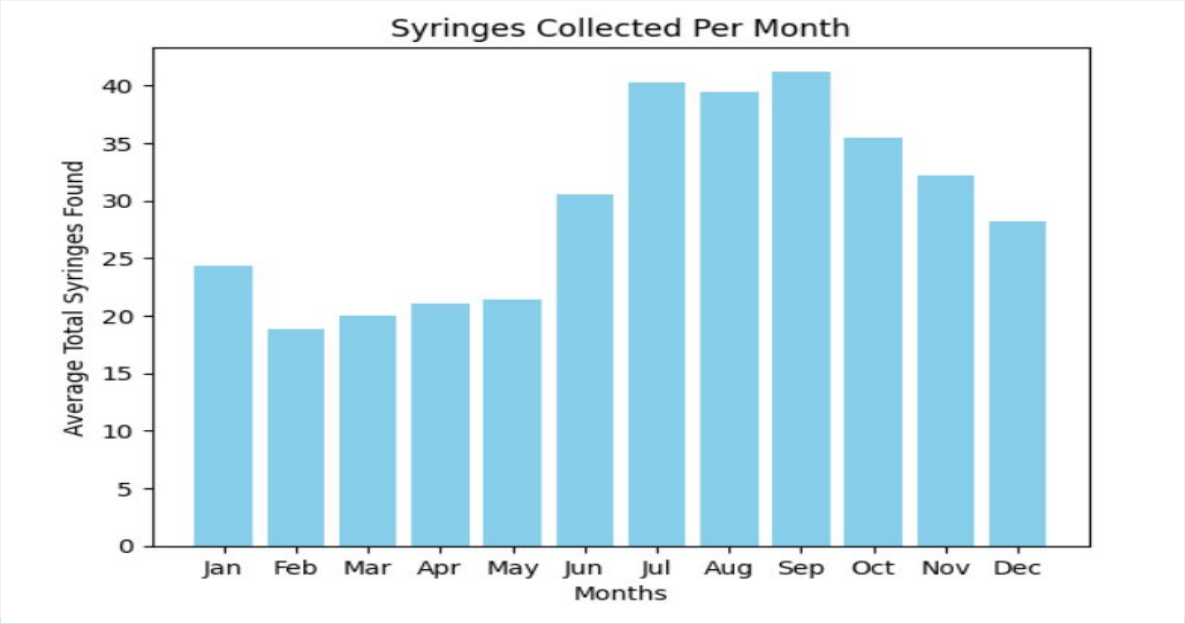


- The median amount of average syringes found in Tompkins Parks is greater than the amount in Coleman Playground

Data Visualization - Bar Chart

```
5]: #-------------------------Data Visualization-------------------------------------#
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#----------Bar Chart-------

# Create a Figure and Axes object
fig, ax = plt.subplots()

#Sort months so it shows correclty in the bar chart
syringe_df_sortedby_month = syringe_df.sort_values(by='month')

#Multiple cells had duplicates of the same months. had to drop them to correctly display the chart
unique_months_df =syringe_df_sortedby_month[['month', 'month_text']].drop_duplicates()

#Find what is the average amount of syringes that are found each month.
avg_syringe_per_month = syringe_df.groupby('month')['total_syringes'].mean()
#print (avg_syringe_per_month)

#Set variables
months = unique_months_df['month_text'].values
values = avg_syringe_per_month.values

#Plot the data in a bar chart
ax.bar(months, values, color='skyblue')
ax.set_title('Syringes Collected Per Month')
ax.set_xlabel("Months")
ax.set_ylabel("Average Total Syringes Found")

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

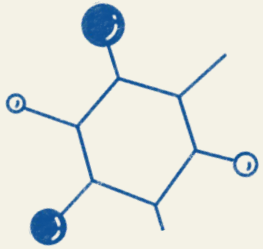
Data Visualization - Box Plot



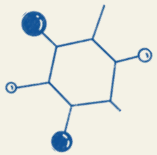
- The average number of syringes found per month increases during the summer months an decreases slightly in the enter the winter months

04

Regression Analysis



Regression Analysis



```
13]: import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn.preprocessing import LabelEncoder

     #Multiple Linear Regression Model with columns 'Source' which held the underlying source of data for this
     #entry and the 'year' number to predict the "Group" value which holds the names of the groups responsible for the
     #collection of syringes

     # Here i will encode categorical variables using one-hot encoding. Changing the values to binary
     df_encoded = pd.get_dummies(syringe_df, columns=['source'], drop_first=True)

     # Encode the 'group' column as well
     label_encoder = LabelEncoder()
     df_encoded['group_encoded'] = label_encoder.fit_transform(syringe_df['group'])

     # Split the data into x and y variables
     X = df_encoded[['year'] + list(df_encoded.filter(regex='source_').columns)]
     y = df_encoded['group_encoded']

     # Split the data
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     # Create and fit the linear regression model!!
     model = LinearRegression()
     model.fit(X_train, y_train)

     # Make predictions with the test sets
     y_pred = model.predict(X_test)
```

- Used pd.get.dummies to convert the categorical information to binary values
- Multiple Linear Regression Model with columns 'Source' and the 'Year' number to predict the "Group" value which holds the names of the groups responsible for the
- collection of syringes

Regression Analysis: Correlation Coefficient (r)

- For multiple linear regression, the correlation coefficient (r) is typically used to describe the relationship between each independent variable and the dependent variable individually.

```
11]: #R and r^2 values
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y_test, y_pred)
r_squared = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r_squared)
```

```
Mean Squared Error: 0.39449215923140313
R-squared: 0.7317946533029376
```

- -1 indicates a perfect negative linear relationship,
- 0 indicates no linear relationship, and
- 1 indicates a perfect positive linear relationship.
- My value was 0.39 or 0.4 which indicates that there was no linear relationship.

Regression Analysis: Coefficient of Determination (R-squared)

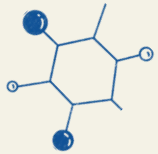
- For multiple linear regression, R-squared is calculated as the proportion of the variance in the dependent variable that is predictable from the independent variables.

```
11]: #R and r^2 values
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y_test, y_pred)
r_squared = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r_squared)
```

```
Mean Squared Error: 0.39449215923140313
R-squared: 0.7317946533029376
```

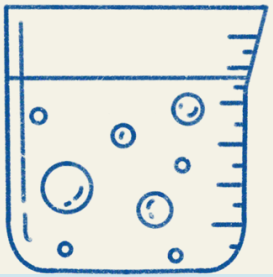
- 0 indicates that the independent variables do not explain any of the variability of the dependent variable, and
- 1 indicates that the independent variables explain all of the variability of the dependent variable.
- My value for R-Squared, it is 0.73

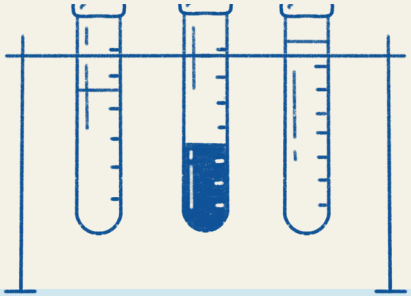
Conclusion



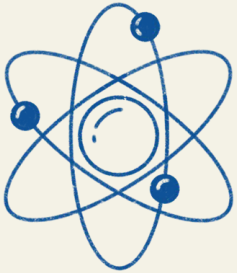
Using Data Science and Regression Analysis is an important skill in understanding, manipulating and presenting data in an organized way that provides insight on various trends. From this analysis I was able to better understand the dataset I chose and find significance in the numbers.

In conclusion, this scientific report provides valuable insights into syringe surveillance in NYC and serves as a foundation for informed decision-making and policy development aimed at addressing syringe litter and ensuring the well-being of NYC residents.





Thanks!



Do you have any questions?

youremail@freepik.com
+34 654 321 432
yourwebsite.com

Resources

NYC Open Data:
https://data.cityofnewyork.us/Public-Safety/Summary-of-Syringe-Data-in-NYC-Parks/t8xi-d5wb/about_data

https://www.nrpa.org/parks-recreation-magazine/2019/february/addressing-public-injection-and-syringe-disposal-in-nyc-parks/