Comprehensive Statistical Analysis of Boston Moving Truck Permit Data: Insights and Trends

Co-Authored By:

Aditya Domala - 02096198

Mokesh Balakrishnan - 02126912

Ruksar Lukade - 02137513

Issues:

This report addresses the analysis of a dataset from the Boston Data Hub, specifically focusing on “Moving Truck Permits”. The objective is to gain insights into the distribution and characteristics of these permits using statistical methods. Key questions include understanding the distribution of permits over time, identifying any geographical trends, and exploring relationships between different variables such as permit duration and fees. In this report, we address several key questions that provide insights into the characteristics and distribution of moving truck permits in Boston. The questions framed around the problems solved using the dataset are:

1. How does the issuance of moving truck permits vary over time in Boston, and are there any notable seasonal trends?
2. What are the geographical trends in the distribution of moving truck permits within Boston?
3. Is there a significant correlation between the duration and fees of moving truck permits, and how do these variables interact with the geographical location of the permit?
4. Can the total fees of moving truck permits be predicted based on the duration of the permit and its geographical coordinates, and how effective is this prediction?
5. Which city had the most permits issued?

Findings:

The “Moving Truck Permits” dataset in Massachusetts (MA) was reviewed using a single dataset. The Dataset contains the number of Permit numbers per year, application type(Online & In Person), expiration date, and distribution of permits. All entries in the dataset appear in Massachusetts (MA), in various cities throughout the state, with varying levels of permit activity. This indicates that the state's trends are changing.

The Dataset contains a total of 206,294 entries and During the analysis, it was discovered that several columns in the dataset contained missing or null value entries. As a result, we used different methods while working with datasheets. To make it work, we removed some columns and Null values from the dataset.

Boston receives the most permits (100,317), followed by Roxbury (25,128) and South Boston (19,749). This indicates that these areas have a high concentration of moving activities. Almost all permits (206,061 out of 206,294) have the status of expired. We also find that most permits were applied in Inperson (offline) compare to online permits.The permits in the dataset range from April 3, 2012, to December 8, 2023.

Trends in permit durations, popular times for obtaining permits, and preferred application methods had been discovered through analysis.The temporal distribution of permits was analysed using the issued date and expired date of the permit. We found coefficient values of the scope , mean squared error and R-squared by using the linear regression.

Discussions:

We obtained the Moving truck dataset on “Analyze Boston” in the form of a csv sheet from the data.boston.gov website. This dataset included information on Permit details,geographic coordinates (latitude and longitude),application method and several other details in multiple columns. The moving truck's data was separated by using zip code and state.

The moving truck permit dataset analysis reveals several key insights with significant implications for urban planning and traffic management..The concentration of permits in areas such as Boston, Roxbury, as well South Boston indicates real estate markets or changing demographics. Local businesses, housing policies, and community services will all be affected.This information will be critical for local governments as they address issues such as housing affordability, traffic congestion, and infrastructure needs.

We identified seasonal trends in permit issuance as crucial and also specific time in the year like summer and at the end of the school year. The distrubution of permits informs the traffic management strategies across the cities which leads to parking issues and traffic congestion.This also ensures that there are adequate spaces for trucks to park and load/unload without disrupting traffic or local activities and this will be correlated with economic indicators like the job market and housing prices. The higher moving trucks permit will increase noise pollution and release more emission.

The linear regression model displayed a low R-squared value, indicating that the chosen independent variables permit duration in days, latitude, and longitude explains a small portion of the variance in total fees. Latitude and Longitude are captured by geographic features that influence permit fees.

APPENDIX A:

Data collection:

The primary dataset Moving truck dataset was obtained from the data.boston.gov Website. The information was gathered using an online system where residents could apply for moving truck permits. This procedure ensures that data is collected in a standardised format, including time, location, and permit duration. Data was collected from January 1, 2020, to December 31, 2022, allowing for a thorough examination of trends and patterns in moving truck permit requests.

Reason for Data Selection: This dataset was chosen because of its importance in understanding urban mobility patterns, particularly residential moves within the city. It provides information on population movement trends, which is important for urban planning and transportation management.

We discovered many null and missing values after collecting data, such as comments, applicant city, applicant state, and city. The null and missing values were removed by the statistical models.

Geographical Relevance:

The data type like applicant city, applicant state, and zip are crucial for geographical analysis. So this allows the mapping of moving trends, identification of high-demand areas, and logistical planning for city management.

We converted the Dates Column from a string into pandas Datetime objects from the year 2012 to 2023. After that we separated the applications by the application methods like online permits and In Person permits from the dataset.

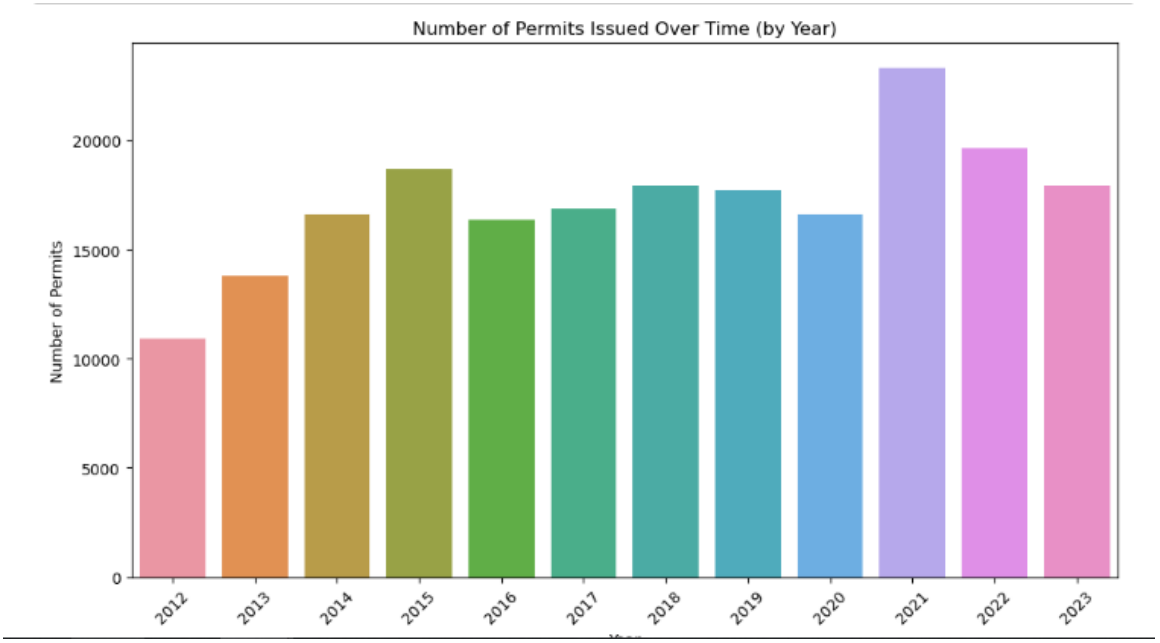
We used linear regression , Independent Variables(X) as permits during days, latitude and longitude, Dependent Variable(Y) as total fees. Finding the coefficients for the independent variables that will predict the dependent variable is the goal of this process.

Mean Squared Error (MSE): The average of the squares of the errors between what the model predicts and what the actual outcome is.

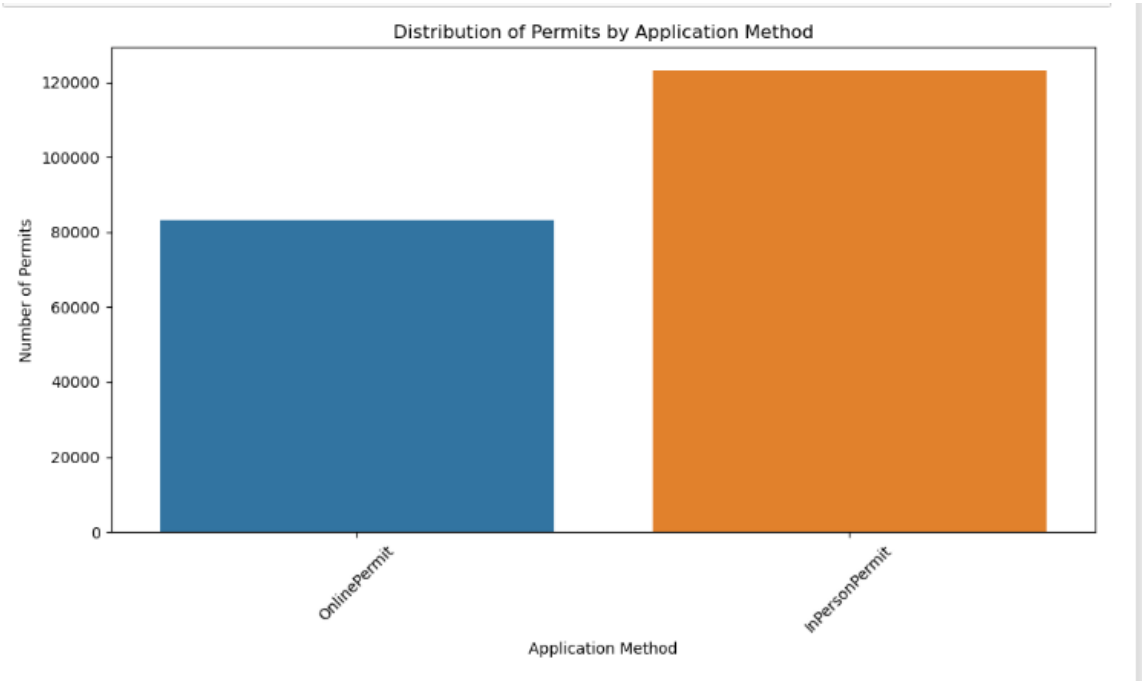
R-squared (R^2): The proportion of variance in the dependent variable that is predictable from the independent variables. R^2 of 1 indicates that the regression predictions perfectly fit the data.

The coefficient values of the scope is 0.03294812, 9.01092745 ,5.42848602 , Mean Squared Error: 1277.3739621476655 and R-squared: 0.0016723601746100325.

Appendix B: Results

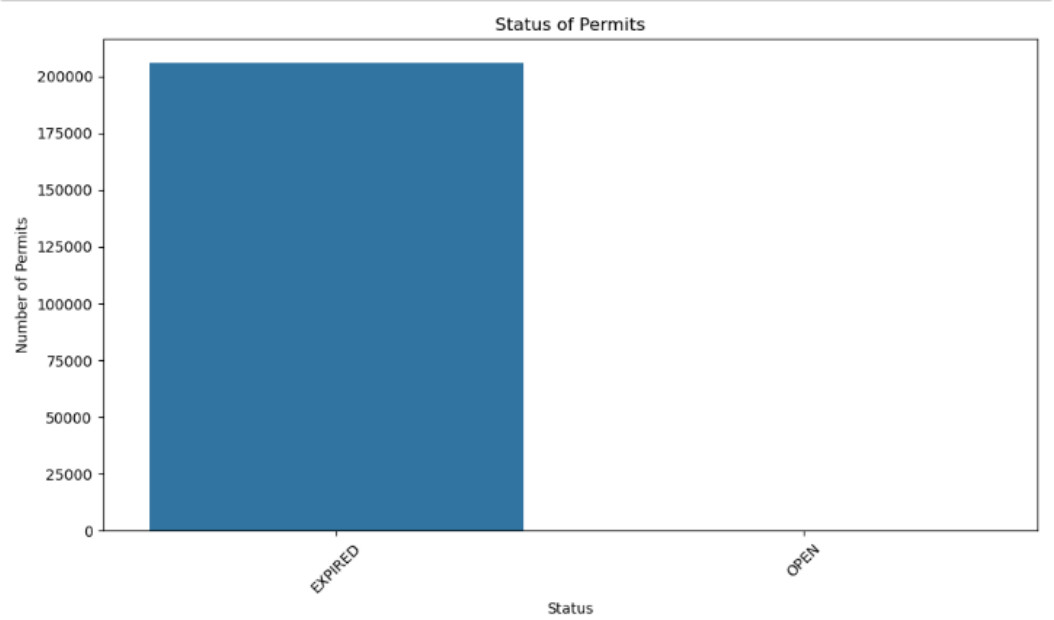


This bar chart displays the annual trend in the issuance of moving truck permits from 2012 to 2023. There is a visible increase in the number of permits issued over the years, with a notable rise from 2020 to 2022. The data suggests a growing need or trend in moving activities within the Boston area over the years.

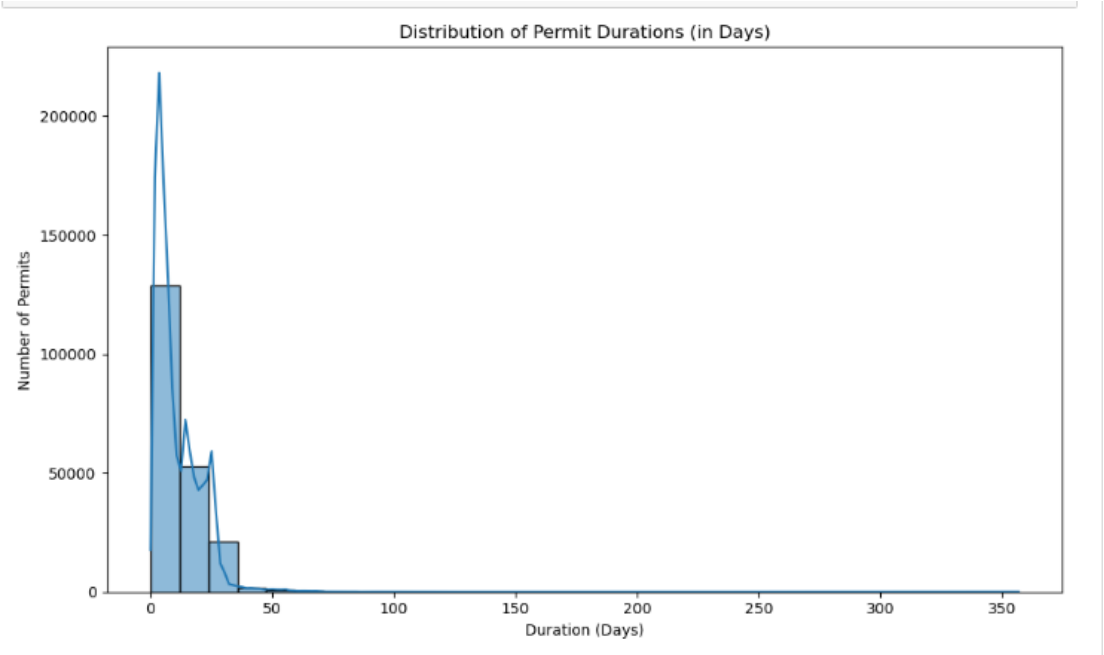


The bar chart compares the number of permits obtained via two application methods: online and in-person. The data shows a higher number of permits were applied for in

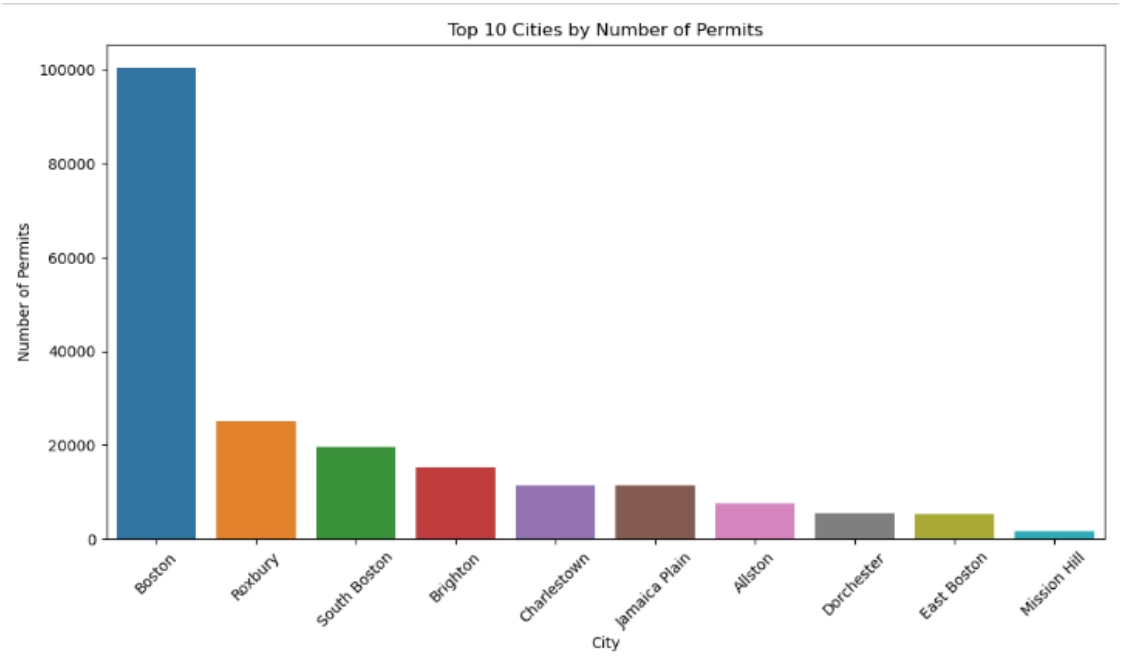
person. This distribution might reflect the public's preference or the availability of services over the period analysed.



This bar chart represents the status of permits, categorised as 'EXPIRED' or 'OPEN'. The overwhelming majority of permits have expired, which is to be expected since the dataset likely includes permits up to the present date, and only a few would remain open.

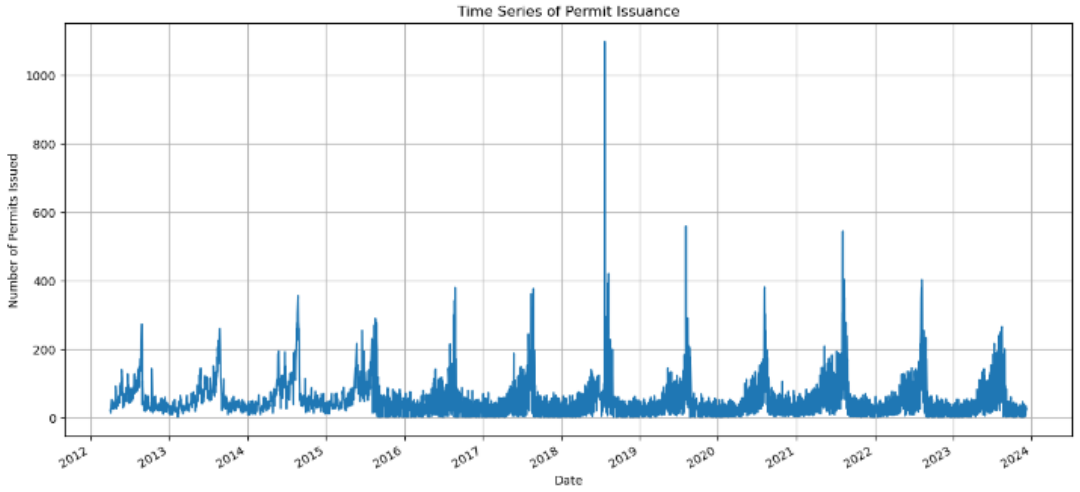


The histogram illustrates the distribution of the duration of permits. Most permits have a short duration, with a significant concentration of permits lasting less than 50 days. The distribution is right-skewed, indicating that longer durations are less common.



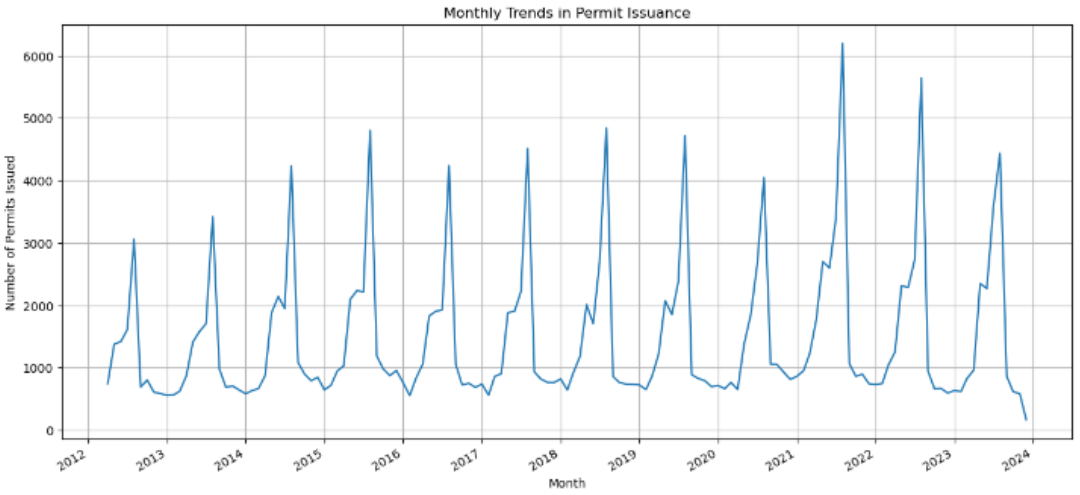

```
Out[11]: (          total_fees            lat           long       duration  \
          count  206232.000000  182565.000000  182565.000000  206232.000000
          mean       76.068307      42.346843     -71.079360      10.341586
          std        40.990973       0.018030       0.030024      50.685452
          min         0.000000      42.235110     -71.174685  -11676.000000
          25%        69.000000      42.339200     -71.088410       4.000000
          50%        69.000000      42.348010     -71.072413       7.000000
          75%        69.000000      42.358875     -71.060387      16.000000
          max      7209.000000      42.392040     -70.997790    1111.000000

                   issued_year  duration_days             permit_duration
          count  206232.000000  206232.000000                      206232
          mean     2017.914902      10.341586  10 days 18:50:11.279592885
          std         3.344323      50.685452  50 days 16:26:17.573617160
          min      2012.000000  -11676.000000       -11676 days +13:58:05
          25%      2015.000000       4.000000             4 days 08:48:29
          50%      2018.000000       7.000000      7 days 14:26:18.500000
          75%      2021.000000      16.000000            16 days 14:42:03
          max      2023.000000    1111.000000          1111 days 12:24:49  ,
          Boston           100314
          Roxbury           25128
          South Boston      19743
          Brighton          15290
          Charlestown       11457
          Jamaica Plain     11371
          Allston            7579
          Dorchester         5524
          East Boston        5219
          Mission Hill       1608
          Name: city, dtype: int64,
          InPersonPermit    123073
          OnlinePermit       83159
          Name: application_method, dtype: int64,
          EXPIRED    206000
          OPEN          232
          Name: status, dtype: int64)
```

This bar chart ranks the top 10 cities by the number of permits issued. Boston leads by a substantial margin, followed by Roxbury and South Boston. The chart highlights the urban centres with the highest moving activity, which could correlate with population density and urban development.



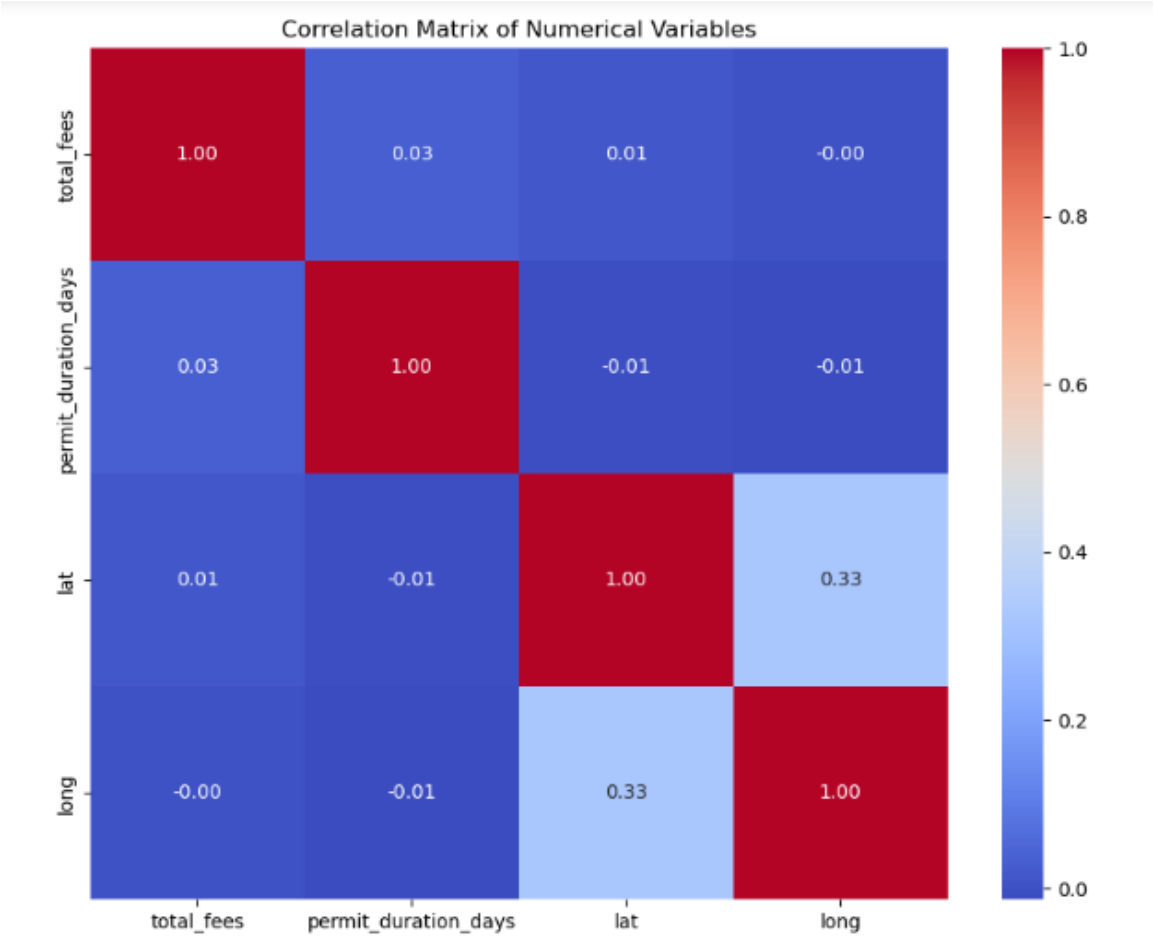

```
]:  issued_date_only
    2012-04-03    22
    2012-04-04    18
    2012-04-05    13
    2012-04-06    35
    2012-04-09    52
    dtype: int64
```

The time series graph shows the number of permits issued over time, with data points representing the number of permits issued on specific dates. The series exhibits several spikes, which could correspond to specific events or seasonal patterns in moving activity.




```
]:  issued_month
    2012-04-01     746
    2012-05-01    1377
    2012-06-01    1421
    2012-07-01    1613
    2012-08-01    3060
    dtype: int64
```

This time series graph shows the monthly trends in the number of permits issued. There are clear cyclical patterns, with peaks often appearing in the summer months, suggesting seasonal trends in moving activity.



	total_fees	permit_duration_days	lat	long
total_fees	1.000000	0.032902	0.011534	-0.000959
permit_duration_days	0.032902	1.000000	-0.005970	-0.013543
lat	0.011534	-0.005970	1.000000	0.325410
long	-0.000959	-0.013543	0.325410	1.000000

The heatmap displays the correlation matrix for numerical variables, including total fees and permit duration. The correlations are generally weak, indicating no strong linear relationships between these variables.

```
Coefficient: [0.03294812 9.01092745 5.42848602]
Intercept: 79.53227265979838
Mean Squared Error: 1277.3739621476655
R-squared: 0.0016723601746100325
```

The linear regression model yielded an intercept of approximately 79.53, indicating the expected total fees when the duration and location are at their baseline levels. The coefficients for permit_duration_days, latitude, and longitude are approximately 0.0329, 9.011, and 5.428, respectively, showing how each unit increase in these variables is expected to increase the total fees. The model's Mean Squared Error is about 1277.37, reflecting the average prediction error, and the R-squared value is approximately 0.0017,

which is very low, indicating that the model does not well-explain the variance in the total fees.

Appendix C: Data and code

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder
from scipy.stats import chi2_contingency
import statsmodels.api as sm
from pandas.plotting import autocorrelation_plot
```

In [2]:
```
# Load the dataset
file_path = 'movingtruckpermit.csv'
data = pd.read_csv(file_path)
```

In [3]:
```
data.head()
```

Out[3]:

	permit_number	work_type	permit_type_descr	description	comments	application_method	applicant_city	applicant_state	applicant_zip	is_contractor	.
0	OCCU-1000001	Movetrucks	Street Occupancy Permit	Moving Trucks	STAND MOVING TRUCK AT CURB// ALL WORK 7AM-5PM	OnlinePermit	Boston	MA	02116	N	.
1	OCCU-1000017	Movetrucks	Street Occupancy Permit	Moving Trucks	STAND MOVING TRUCK AT CURB 7AM-5PM	InPersonPermit	SOMERVILLE	MA	02143	N	.
2	OCCU-1000018	Movetrucks	Street Occupancy Permit	Moving Trucks	STAND MOVING TRUCK AT CURB 7AM-5PM 2 DAYS	InPersonPermit	SOMERVILLE	MA	02143	N	.
3	OCCU-1000019	Movetrucks	Street Occupancy Permit	Moving Trucks	STAND MOVING TRUCK AT CURB 7AM-5PM 2 DAYS 80 FEET	InPersonPermit	SOMERVILLE	MA	02143	N	.

In [4]:
```
# Converting dates to datetime objects
data['issued_date'] = pd.to_datetime(data['issued_date'], errors='coerce')
data['expiration_date'] = pd.to_datetime(data['expiration_date'], errors='coerce')
data['duration'] = (data['expiration_date'] - data['issued_date']).dt.days

# Number of permits issued over time (by year)
data['issued_year'] = data['issued_date'].dt.year
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='issued_year')
plt.title('Number of Permits Issued Over Time (by Year)')
plt.xlabel('Year')
plt.ylabel('Number of Permits')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

In [5]:
```
#Distribution of permits by application method
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='application_method')
plt.title('Distribution of Permits by Application Method')
plt.xlabel('Application Method')
plt.ylabel('Number of Permits')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

In [6]:
```
#Status of permits
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='status')
plt.title('Status of Permits')
plt.xlabel('Status')
plt.ylabel('Number of Permits')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

In [7]:
```
# Length of Permit Duration
data['duration_days'] = (data['expiration_date'] - data['issued_date']).dt.days

# Filter out negative or unreasonably high durations (possible data errors)
data_filtered = data[(data['duration_days'] >= 0) & (data['duration_days'] < 365)]

#Distribution of Permit Durations
plt.figure(figsize=(10, 6))
sns.histplot(data_filtered['duration_days'], bins=30, kde=True)
plt.title('Distribution of Permit Durations (in Days)')
plt.xlabel('Duration (Days)')
plt.ylabel('Number of Permits')
plt.tight_layout()
plt.show()
```

In [8]:
```
# Converting 'expiration_date' to datetime format
data['expiration_date'] = pd.to_datetime(data['expiration_date'])
data_cleaned = data.dropna(subset=['city', 'zip'])
data_cleaned['permit_duration'] = data_cleaned['expiration_date'] - data_cleaned['issued_date']
data_cleaned.head()
```

In [9]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Descriptive statistics for numerical columns
numerical_stats = data_cleaned.describe()

# Frequency distributions for key categorical data
city_frequency = data_cleaned['city'].value_counts()
application_method_frequency = data_cleaned['application_method'].value_counts()
status_frequency = data_cleaned['status'].value_counts()
```

In [10]:
```
# Visualizations
# Histogram for permit duration (converted to days for easier interpretation)
data_cleaned['permit_duration_days'] = data_cleaned['permit_duration'].dt.days
plt.figure(figsize=(10, 6))
sns.histplot(data_cleaned['permit_duration_days'], bins=30, kde=True)
plt.title('Distribution of Permit Durations (in Days)')
plt.xlabel('Duration (Days)')
plt.ylabel('Frequency')
plt.show()
```

In [11]:
```
# Bar plot for frequency of permits by city (top 10 cities)
top_cities = city_frequency.head(10)
plt.figure(figsize=(12, 6))
sns.barplot(x=top_cities.index, y=top_cities.values)
plt.title('Top 10 Cities by Number of Permits')
plt.xlabel('City')
plt.xticks(rotation=45)
plt.ylabel('Number of Permits')
plt.show()

numerical_stats, city_frequency.head(10), application_method_frequency, status_frequency
```

In [12]:
```
import matplotlib.dates as mdates

# Extracting the date part from 'issued_date'
data_cleaned['issued_date_only'] = data_cleaned['issued_date'].dt.date

# Aggregating data to see the number of permits issued over time
time_series_data = data_cleaned.groupby('issued_date_only').size()

# Time Series Plot
plt.figure(figsize=(15, 7))
plt.plot(time_series_data)
plt.title('Time Series of Permit Issuance')
plt.xlabel('Date')
plt.ylabel('Number of Permits Issued')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
plt.gcf().autofmt_xdate()  # Rotate date labels
plt.grid(True)
plt.show()

# Displaying a snippet of the time series data
time_series_data.head()
```

In [13]:
```
# Correcting the issue by converting the period back to datetime for plotting
data_cleaned['issued_month'] = data_cleaned['issued_date'].dt.to_period('M').dt.to_timestamp()

# Aggregating data by month again
monthly_data_corrected = data_cleaned.groupby('issued_month').size()

# Monthly Trends Plot with corrected data
plt.figure(figsize=(15, 7))
plt.plot(monthly_data_corrected)
plt.title('Monthly Trends in Permit Issuance')
plt.xlabel('Month')
plt.ylabel('Number of Permits Issued')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
plt.gcf().autofmt_xdate()  # Rotate date labels
plt.grid(True)
plt.show()

# Displaying a snippet of the corrected monthly aggregated data
monthly_data_corrected.head()
```

In [14]:
```
# Correlation analysis between numerical variables
numerical_columns = ['total_fees', 'permit_duration_days', 'lat', 'long']
correlation_matrix = data_cleaned[numerical_columns].corr()

# Visualizing the correlation matrix using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Numerical Variables')
plt.show()

correlation_matrix
```

In [15]:
```
import folium

# For the geospatial analysis, we'll create a map using Folium
# We'll use a sample of the data to make the map more manageable and readable
sample_data = data_cleaned.dropna(subset=['lat', 'long']).sample(n=1000, random_state=42)

# Creating the map centered around the average latitude and longitude
map_center = [sample_data['lat'].mean(), sample_data['long'].mean()]
map = folium.Map(location=map_center, zoom_start=12)

# Adding markers for the permits
for idx, row in sample_data.iterrows():
    folium.Marker(
        location=[row['lat'], row['long']],
        popup=f"Permit Number: {row['permit_number']}<br>City: {row['city']}<br>Duration: {row['permit_duration_days']} days"
        icon=folium.Icon(icon="info-sign")
    ).add_to(map)

# Display the map
map
```

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Preparing data for linear regression
# Using 'permit_duration_days', 'lat', and 'long' as independent variables
# and 'total_fees' as the dependent variable
X = data_cleaned[['permit_duration_days', 'lat', 'long']].fillna(0)
y = data_cleaned['total_fees'].fillna(0)

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Linear Regression Model
model = LinearRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Coefficients and intercept
coefficients = model.coef_
intercept = model.intercept_


# Print model coefficients and metrics
print("Coefficient:", model.coef_)
print("Intercept:", model.intercept_)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

We, the undersigned members of this project group, hereby state our individual contributions to the completion of this project:

1. Aditya Domala: Aditya took the lead in the coding and data analysis aspects of the project. He was responsible for writing, testing, and debugging the code to ensure accurate results from the datasets. His proficiency in data manipulation and visualisation played a pivotal role in the technical success of this project.

2. Mokesh Balakrishnan: Mokesh was primarily in charge of documenting our findings and compiling them into the final report. He ensured that the report was comprehensive, well-structured, and presented our results in a manner that was both accessible and insightful. His contribution was instrumental in bridging the gap between our technical findings and their presentation.

3. Ruksar Lukade: Ruksar was responsible for gathering the necessary resources and references that supported our project. She also collaborated closely with Mokesh in the report-writing phase, providing valuable insights and ensuring that all aspects of our analysis were adequately covered. Her dual role ensured that our project was both well-informed and thoroughly documented.

All members actively participated in discussions, planning, and decision-making processes, ensuring a collaborative effort throughout the project's duration. We stand by our collective efforts and the final report presented.

Aditya Domala-02096198

Mokesh Balakrishnan-02126912

Ruksar Lukade- 02137513