PROJECT DOCUMENTATION

EXPLORATORY DATA ANALYSIS USING PYTHON

TITLE	**DOHMH New York City Restaurant EDA Project**
NAME	**Pavithra M**
COURSE	**DADS - Offline**
BATCH	**July - 2025**

1.INTRODUCTION

The NYC dataset contains detailed information about food establishments, their inspections, and geographic attributes. It includes variables such as inspection **SCORE**, **GRADE**, establishment identifiers (CAMIS, BIN, BBL), and location details (**ZIPCODE, Latitude, Longitude, Street, Community Board, Council District, Census Tract, NTA**). This dataset enables analysis of food safety compliance across different neighborhoods, identification of geographic patterns, and exploration of governance-related impacts on inspection outcomes.

Project Overview:

- Analyze food inspection scores and grades.
- Study the impact of location factors on results.
- Find geographic and trend patterns.
- Clean data for accurate analysis.
- Provide insights to support better decisions.

2. AIM OF THE PROJECT

- To study NYC food inspection scores and grades.
- To understand the effect of location factors on outcomes.
- To clean and transform data to enhance quality and accuracy for analysis.
- To apply statistical hypothesis testing to validate observed patterns and relationships.
- To identify geographic and trend patterns through EDA.
- To handle data issues for reliable analysis.
- To provide insights for better food safety decisions.

3. PROBLEM STATEMENT

Food safety is a critical public health concern in New York City, where thousands of establishments are inspected regularly. However, inspection outcomes vary across locations, and patterns influencing scores and grades are not always clear. Without proper analysis, it becomes difficult to identify key risk factors, geographic trends, and data quality issues. This project addresses these challenges by analyzing inspection data to uncover insights that can improve monitoring and decision-making.

4. PROJECT WORKFLOW

1. Data Understanding

- Collect and explore the NYC food inspection dataset.

- Identify key variables (SCORE, GRADE, location features, etc.).
- Understand data structure, types, and relevance.

2. Data Cleaning & Preprocessing

- Handle missing values, invalid entries, and duplicates.
- Treat outliers using IQR or statistical methods.
- Standardize formats (ZIP codes, phone numbers, etc.).
- Encode categorical variables if needed.

3. Statistical Tests

- Perform Independent T-Statistic test
- Perform Chi Square Test

4. Exploratory Data Analysis (EDA)

- **Univariate Analysis**: Distribution of SCORE and ZIPCODE.
- **Bivariate Analysis**: Relationship between SCORE and GRADE, SCORE vs. location features.
- **Multivariate Analysis**: Box plots, pair plots and Correlation heatmaps.
- Detect multicollinearity among features.

5. Insights & Interpretation

- Identify geographic trends.
- Highlight weak, moderate, and strong correlations.
- Detect multicollinearity and recommend feature selection.
- Provide data-driven insights for food safety compliance.

6. Visualization

- Create histograms, bar charts, scatter plots, and boxplots.
- Develop correlation heatmaps and geographic visualizations.
- Summarize findings through clear visual storytelling.

7. Conclusion & Recommendations

- Summarize key findings (influence of location, geographic patterns).
- Suggest improvements for inspection monitoring.
- Provide recommendations for policy and decision-making.

5. Data Understanding

This step involves exploring the NYC food inspection dataset to get familiar with its structure, variables, and contents. Key aspects include identifying important features (like SCORE, GRADE, location details), checking data types, and summarizing basic statistics. The goal is to understand what the data represents and assess its quality before further analysis.

Dataset Overview

- **Rows : 290022**
- **Columns : 27**

Key Variables

1. Inspection Outcome Variables:

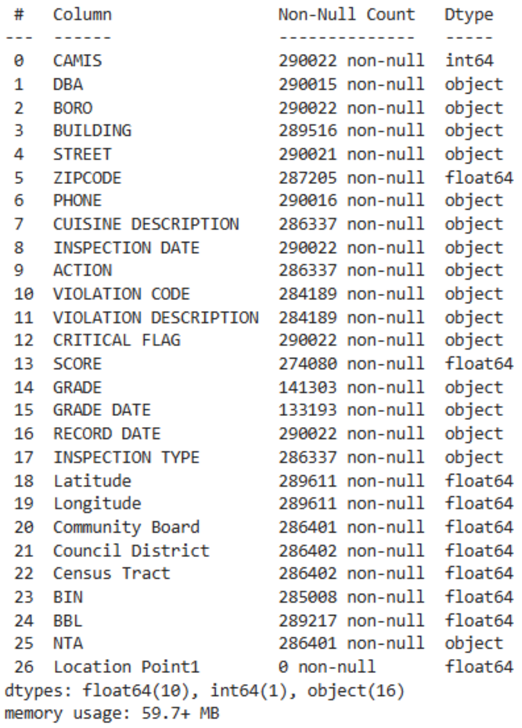
- SCORE
- GRADE

2. Location Variables:

- Community Board
- Council District
- Latitude & Longitude
- ZIPCODE
- Street, Census Tract, BIN, BBL, NTA

3. Establishment Identifiers & Contact:

- CAMIS
- PHONE



```
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   CAMIS                  290022 non-null  int64
 1   DBA                    290015 non-null  object
 2   BORO                   290022 non-null  object
 3   BUILDING               289516 non-null  object
 4   STREET                 290021 non-null  object
 5   ZIPCODE                287205 non-null  float64
 6   PHONE                  290016 non-null  object
 7   CUISINE DESCRIPTION    286337 non-null  object
 8   INSPECTION DATE        290022 non-null  object
 9   ACTION                 286337 non-null  object
 10  VIOLATION CODE         284189 non-null  object
 11  VIOLATION DESCRIPTION  284189 non-null  object
 12  CRITICAL FLAG          290022 non-null  object
 13  SCORE                  274080 non-null  float64
 14  GRADE                  141303 non-null  object
 15  GRADE DATE             133193 non-null  object
 16  RECORD DATE            290022 non-null  object
 17  INSPECTION TYPE        286337 non-null  object
 18  Latitude               289611 non-null  float64
 19  Longitude              289611 non-null  float64
 20  Community Board        286401 non-null  float64
 21  Council District       286402 non-null  float64
 22  Census Tract           286402 non-null  float64
 23  BIN                    285008 non-null  float64
 24  BBL                    289217 non-null  float64
 25  NTA                    286401 non-null  object
 26  Location Point1        0 non-null       float64
dtypes: float64(10), int64(1), object(16)
memory usage: 59.7+ MB
```

6. Data Cleaning

Data cleaning ensures the dataset is accurate, consistent, and ready for analysis.

Key steps include:

Dropping Features:

- Dropped Features that having too many missing values
- Dropped Features that are not useful for analysis

Handling Missing Values:

- Identify missing or null entries in critical columns (e.g., SCORE, GRADE, Latitude/Longitude).
- Fill missing values using appropriate methods: mean for numerical data and “Not Available” for categorical data, or drop rows if necessary.

Removing Duplicates:

- Check for duplicate rows or repeated establishment IDs (CAMIS).
- Remove duplicates to prevent skewed analysis or bias.

Handling Outliers:

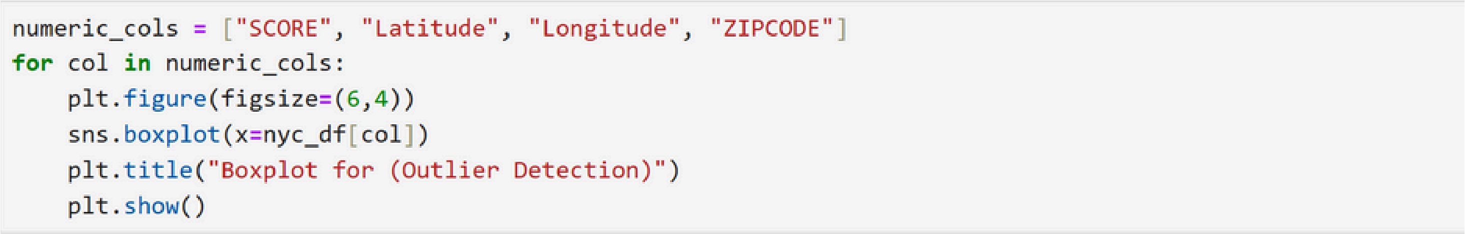
- Detect outliers in numeric columns like SCORE or Latitude/Longitude using IQR Method.

Correcting Invalid Data:

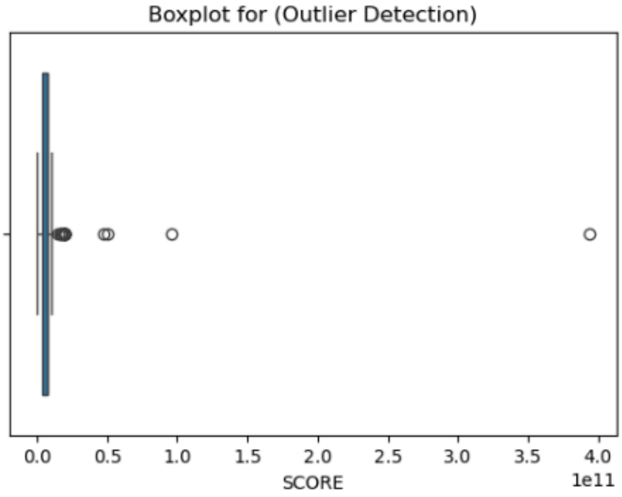
- Standardize formats for categorical variables (GRADE should be A/B/C).
- Correct inconsistencies in ZIP codes, phone numbers, and address fields.

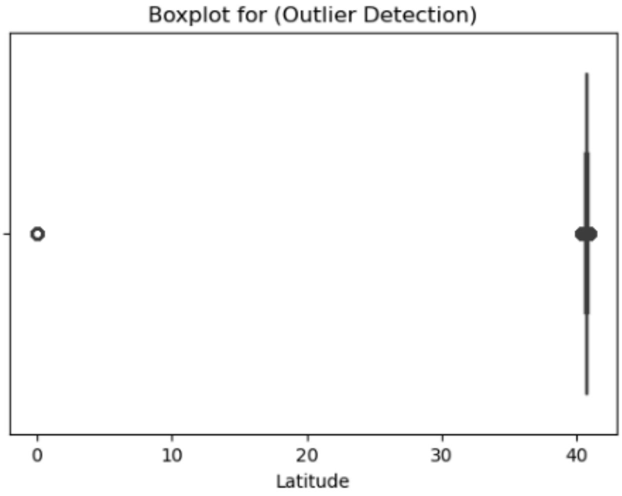
Outlier Treatment:

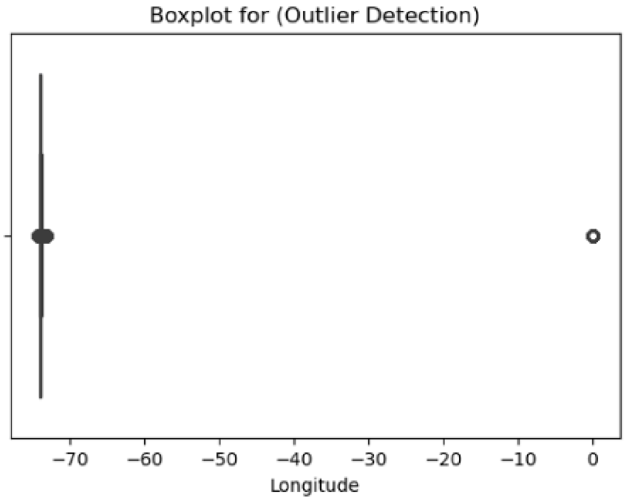
- Community Board, Council District, Census Tract, BIN, BBL are Categorical Codes . So, there is no need to find outliers in these Features.
- Used Boxplot for Identify Outliers in numeric columns.

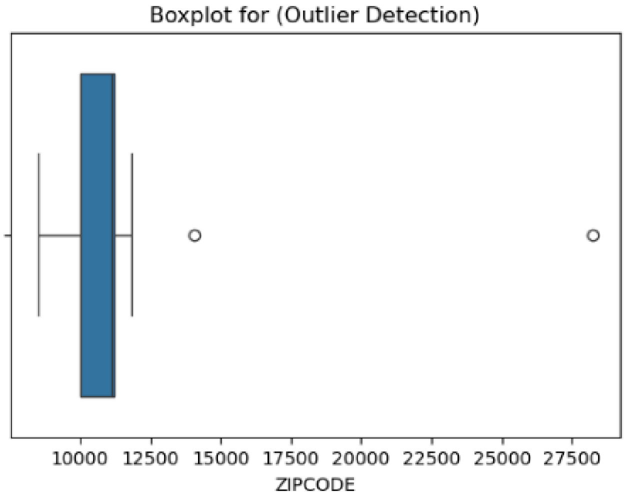


```
numeric_cols = ["SCORE", "Latitude", "Longitude", "ZIPCODE"]
for col in numeric_cols:
    plt.figure(figsize=(6,4))
    sns.boxplot(x=nyc_df[col])
    plt.title("Boxplot for (Outlier Detection)")
    plt.show()
```

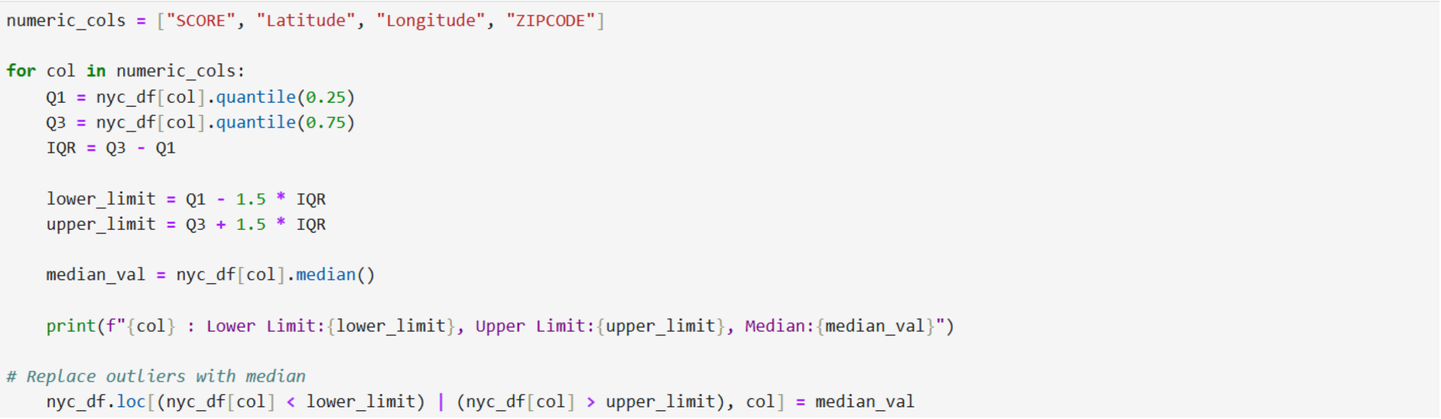






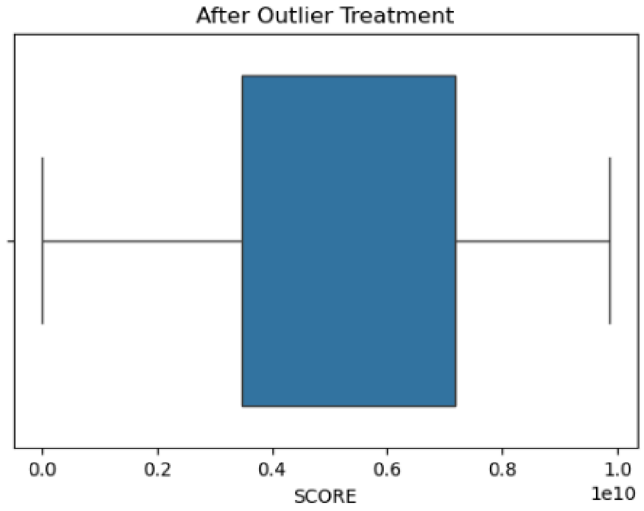


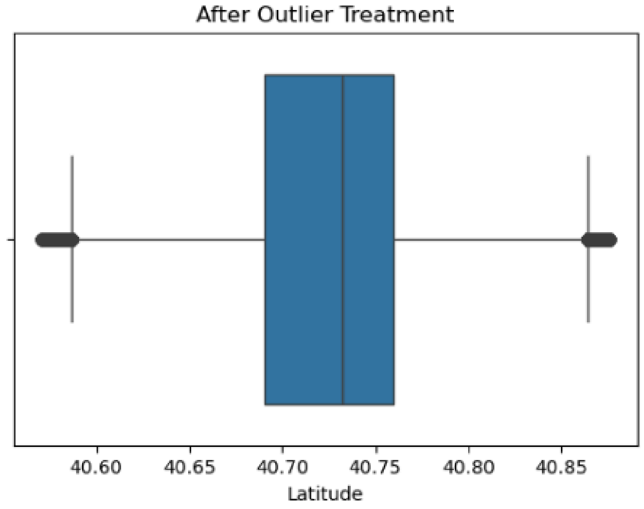
- In my dataset, all the numeric columns have ouliers.
- Treating outliers with IQR method using for loop

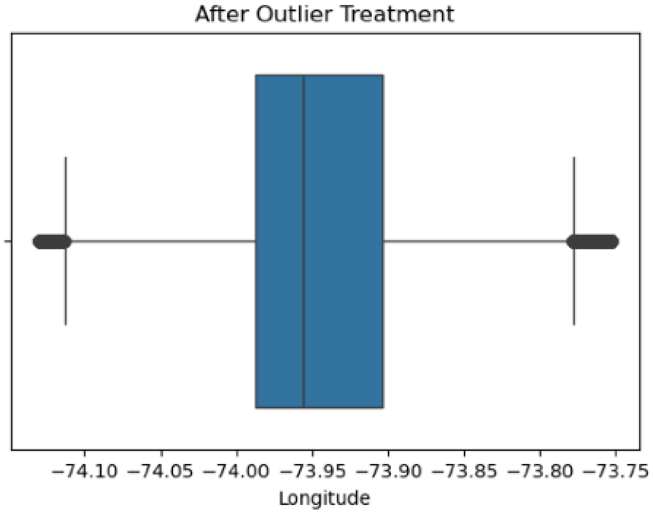


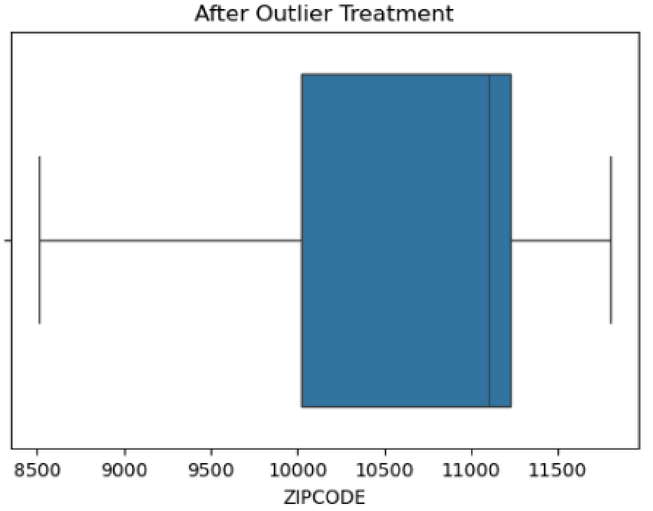
```
numeric_cols = ["SCORE", "Latitude", "Longitude", "ZIPCODE"]

for col in numeric_cols:
    Q1 = nyc_df[col].quantile(0.25)
    Q3 = nyc_df[col].quantile(0.75)
    IQR = Q3 - Q1

    lower_limit = Q1 - 1.5 * IQR
    upper_limit = Q3 + 1.5 * IQR

    median_val = nyc_df[col].median()

    print(f"{col} : Lower Limit:{lower_limit}, Upper Limit:{upper_limit}, Median:{median_val}")

# Replace outliers with median
    nyc_df.loc[(nyc_df[col] < lower_limit) | (nyc_df[col] > upper_limit), col] = median_val
```

After treating outlier treatment:


After Outlier Treatment
0.0
0.2
0.4
0.6
0.8
1.0
1e10
SCORE


After Outlier Treatment
40.60
40.65
40.70
40.75
40.80
40.85
Latitude





After cleaning my dataset :

- **Rows - 290016**
- **Columns - 15**

7. Feature Engineering

Feature engineering enhances the dataset's usefulness by creating meaningful, analysis-ready variables that improve insights, trends detection, and predictive modeling performance.

Reducing Multicollinearity

- Identify highly correlated features.
- Dropped features to prevent redundancy in modeling.

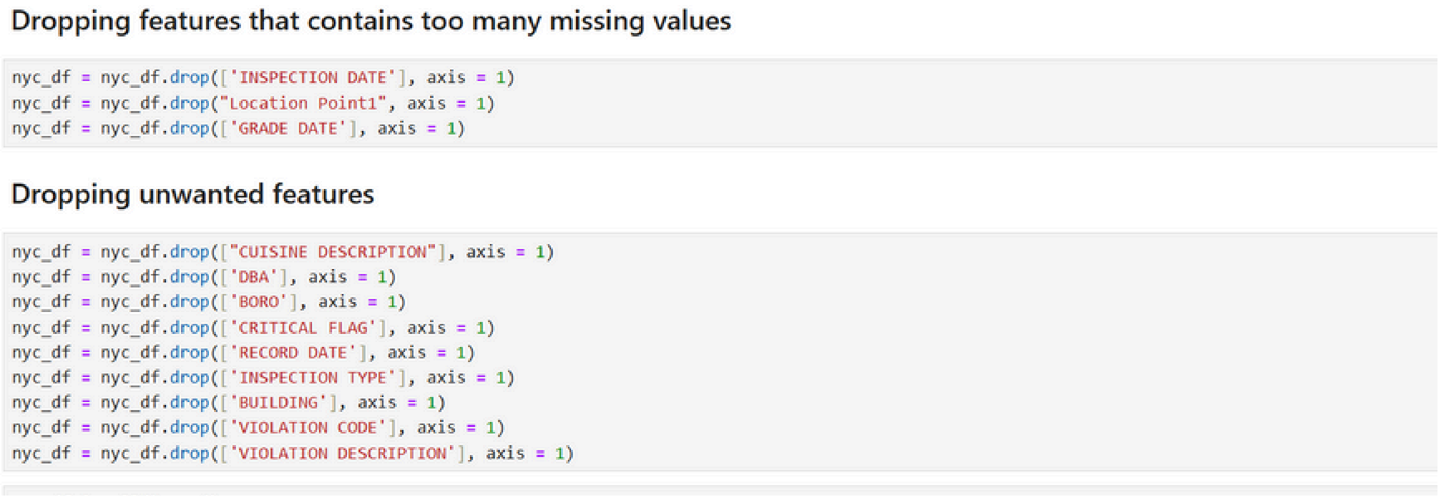
8.Filtering Data For analysis

Filtering is the process of selecting a subset of the dataset that is relevant, clean, and suitable for a specific analysis or visualization. This helps improve accuracy and focus on meaningful insights.

Key Steps:

Remove Unnecessary Columns:

- Drop columns that are not required for analysis (e.g., DBA) to simplify the dataset.



Dropping features that contains too many missing values

```
nyc_df = nyc_df.drop(['INSPECTION DATE'], axis = 1)
nyc_df = nyc_df.drop("Location Point1", axis = 1)
nyc_df = nyc_df.drop(['GRADE DATE'], axis = 1)
```

Dropping unwanted features

```
nyc_df = nyc_df.drop(["CUISINE DESCRIPTION"], axis = 1)
nyc_df = nyc_df.drop(['DBA'], axis = 1)
nyc_df = nyc_df.drop(['BORO'], axis = 1)
nyc_df = nyc_df.drop(['CRITICAL FLAG'], axis = 1)
nyc_df = nyc_df.drop(['RECORD DATE'], axis = 1)
nyc_df = nyc_df.drop(['INSPECTION TYPE'], axis = 1)
nyc_df = nyc_df.drop(['BUILDING'], axis = 1)
nyc_df = nyc_df.drop(['VIOLATION CODE'], axis = 1)
nyc_df = nyc_df.drop(['VIOLATION DESCRIPTION'], axis = 1)
```

Select Relevant Rows:

- Filter rows based on conditions, e.g., valid SCORE values, specific grades (A, B, C), or certain ZIP codes/Community Boards.

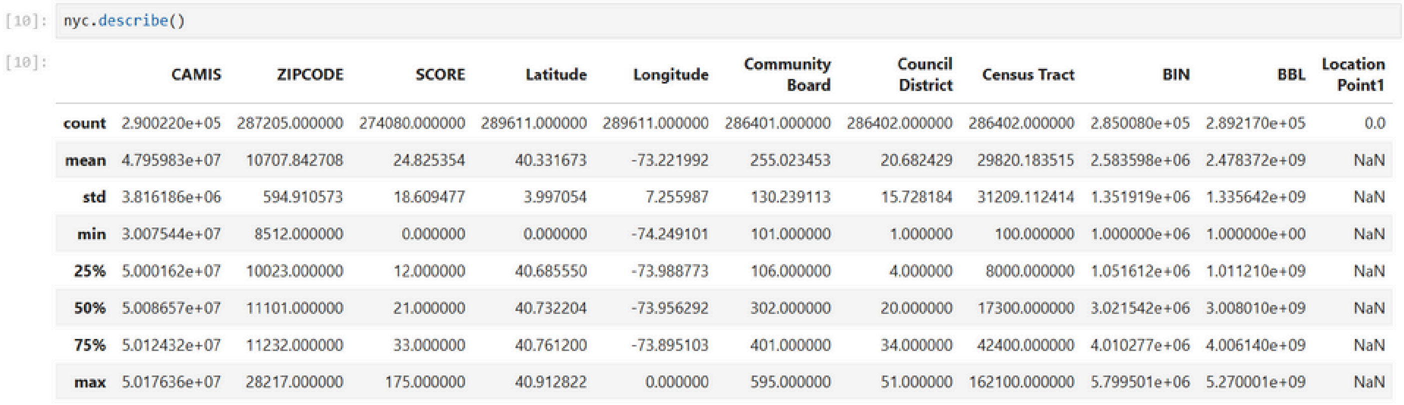
Handle Outliers:

- Exclude extreme SCORE values or incorrect latitude/longitude data to prevent skewed results.

9.Statistical analysis and Testing

Descriptive Statistics:

Descriptive statistics summarize and describe the main features of a dataset. They include metrics like mean, median, standard deviation, minimum, maximum, and quartiles, which give a snapshot of the distribution and spread of values.



```
[10]: nyc.describe()
```

[10]:

	CAMIS	ZIPCODE	SCORE	Latitude	Longitude	Community Board	Council District	Census Tract	BIN	BBL	Location Point1
count	2.900220e+05	287205.000000	274080.000000	289611.000000	289611.000000	286401.000000	286402.000000	286402.000000	2.850080e+05	2.892170e+05	0.0
mean	4.795983e+07	10707.842708	24.825354	40.331673	-73.221992	255.023453	20.682429	29820.183515	2.583598e+06	2.478372e+09	NaN
std	3.816186e+06	594.910573	18.609477	3.997054	7.255987	130.239113	15.728184	31209.112414	1.351919e+06	1.335642e+09	NaN
min	3.007544e+07	8512.000000	0.000000	0.000000	-74.249101	101.000000	1.000000	100.000000	1.000000e+06	1.000000e+00	NaN
25%	5.000162e+07	10023.000000	12.000000	40.685550	-73.988773	106.000000	4.000000	8000.000000	1.051612e+06	1.011210e+09	NaN
50%	5.008657e+07	11101.000000	21.000000	40.732204	-73.956292	302.000000	20.000000	17300.000000	3.021542e+06	3.008010e+09	NaN
75%	5.012432e+07	11232.000000	33.000000	40.761200	-73.895103	401.000000	34.000000	42400.000000	4.010277e+06	4.006140e+09	NaN
max	5.017636e+07	28217.000000	175.000000	40.912822	0.000000	595.000000	51.000000	162100.000000	5.799501e+06	5.270001e+09	NaN

Hypothesis Testing :

Hypothesis testing is a statistical method used to validate whether an observed pattern in the data is due to chance or represents a meaningful difference/relationship. It involves setting up a Null Hypothesis (Ho), which assumes no effect or difference, and an Alternative Hypothesis (H1), which assumes there is a significant effect or difference. The decision to reject or accept Ho is based on a p-value compared to a significance level (alpha = 0.05 in this case).

Tests Performed

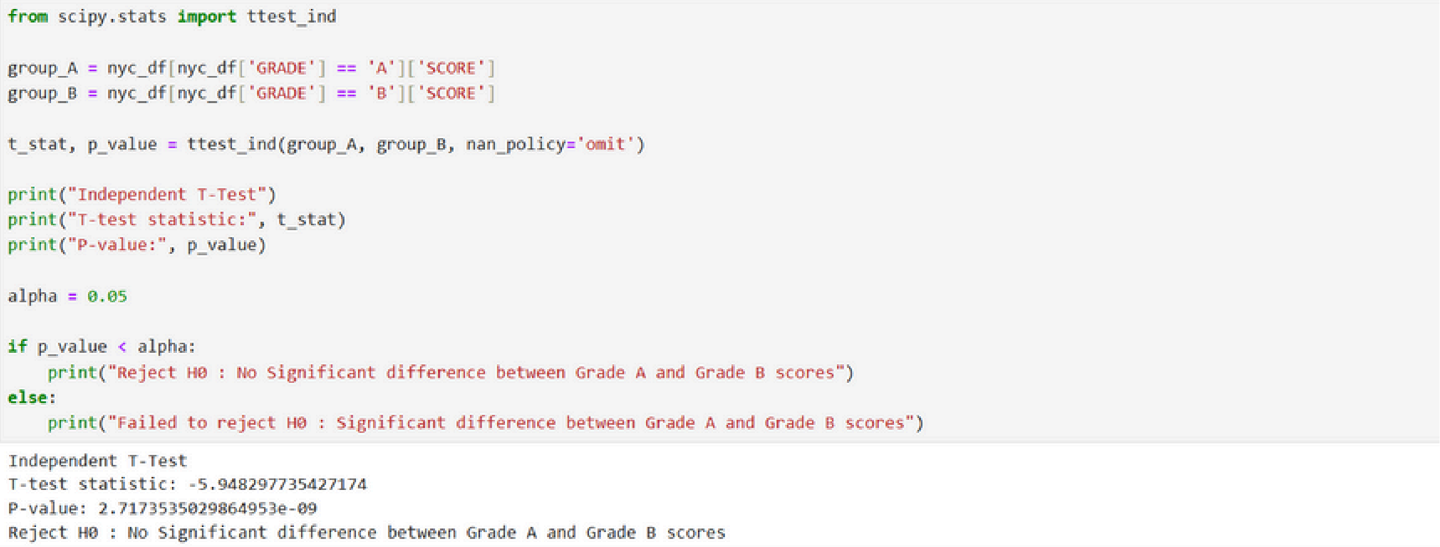
1.Independent Sample t-test

Comparing SCORE between GRADE A and GRADE B Restaurants

Null Hypothesis Ho : No Significant difference between GRADE A and GRADE B SCORES

Alternate Hypothesis Ha : Significant difference between GRADE A and GRADE B SCORES

Results : The p_value is less than 0.05.Hence,the SCORES of GRADE A and GRADE B have significant differences.



```
from scipy.stats import ttest_ind

group_A = nyc_df[nyc_df['GRADE'] == 'A']['SCORE']
group_B = nyc_df[nyc_df['GRADE'] == 'B']['SCORE']

t_stat, p_value = ttest_ind(group_A, group_B, nan_policy='omit')

print("Independent T-Test")
print("T-test statistic:", t_stat)
print("P-value:", p_value)

alpha = 0.05

if p_value < alpha:
    print("Reject H0 : No Significant difference between Grade A and Grade B scores")
else:
    print("Failed to reject H0 : Significant difference between Grade A and Grade B scores")
```

```
Independent T-Test
T-test statistic: -5.948297735427174
P-value: 2.7173535029864953e-09
Reject H0 : No Significant difference between Grade A and Grade B scores
```

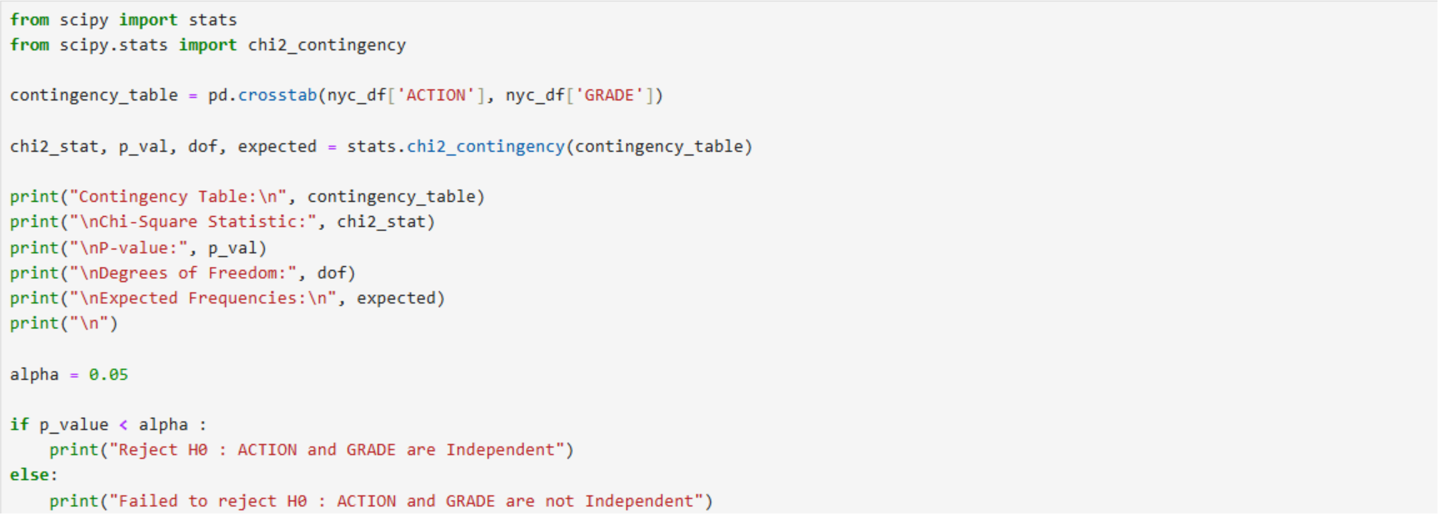
2.Chi Square Test

Taking two categorical Features ACTION and GRADE to check they are independent or not

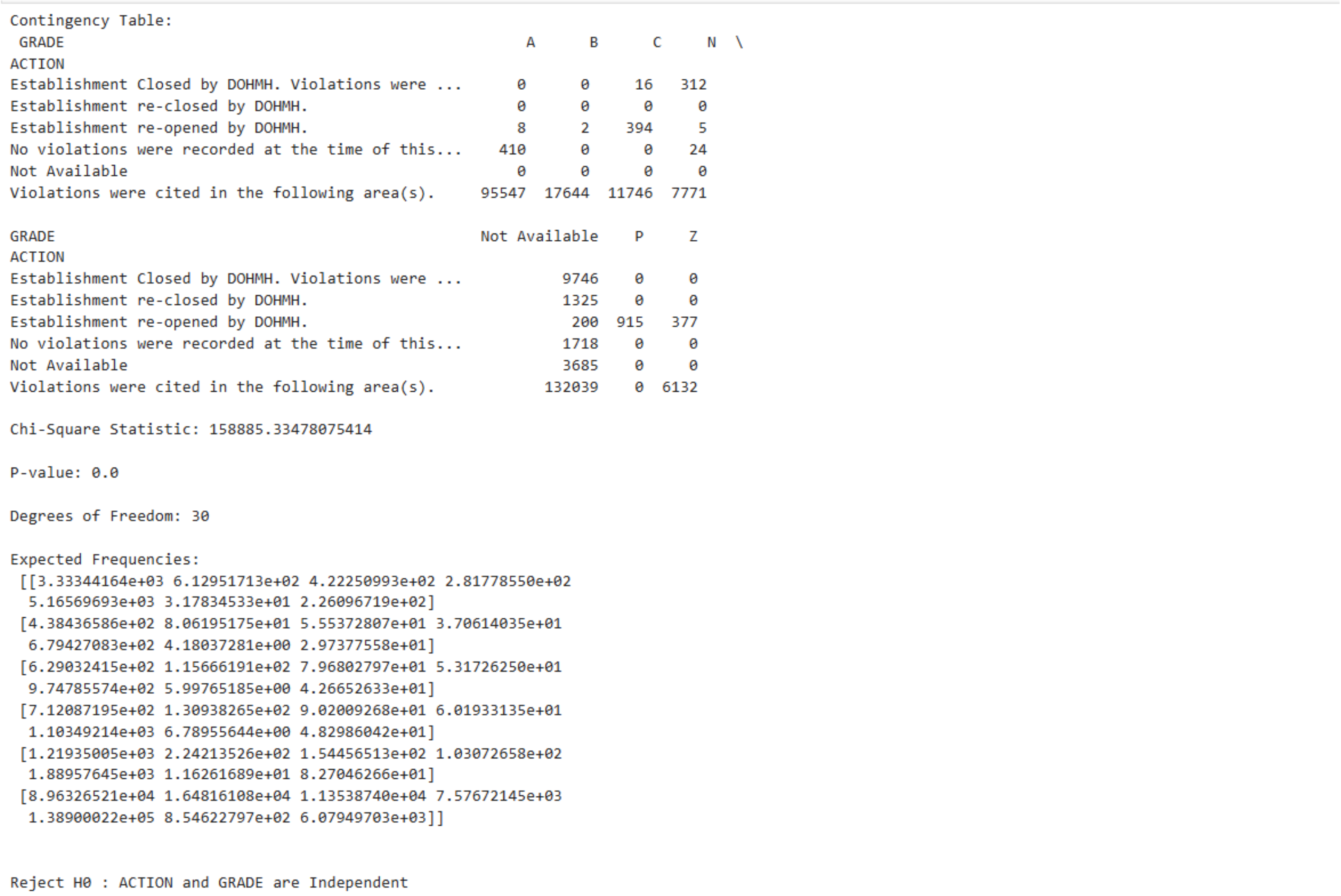
Null Hypothesis Ho : ACTION and GRADE are independent

Alternate Hypothesis Ha : ACTION and GRADE are dependent

Results : The p_value is 0.0 which is less than 0.05.HenceACTION and GRADE are dependent.



```
from scipy import stats
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(nyc_df['ACTION'], nyc_df['GRADE'])

chi2_stat, p_val, dof, expected = stats.chi2_contingency(contingency_table)

print("Contingency Table:\n", contingency_table)
print("\nChi-Square Statistic:", chi2_stat)
print("\nP-value:", p_val)
print("\nDegrees of Freedom:", dof)
print("\nExpected Frequencies:\n", expected)
print("\n")

alpha = 0.05

if p_value < alpha :
    print("Reject H0 : ACTION and GRADE are Independent")
else:
    print("Failed to reject H0 : ACTION and GRADE are not Independent")
```

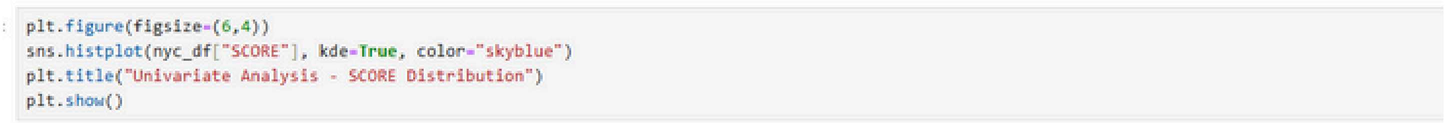


```
Contingency Table:
 GRADE                                              A      B      C     N  \
ACTION
Establishment Closed by DOHMH. Violations were ...      0      0     16   312
Establishment re-closed by DOHMH.                       0      0      0     0
Establishment re-opened by DOHMH.                       8      2    394     5
No violations were recorded at the time of this...    410      0      0    24
Not Available                                           0      0      0     0
Violations were cited in the following area(s).     95547  17644  11746  7771

GRADE                                              Not Available     P     Z
ACTION
Establishment Closed by DOHMH. Violations were ...          9746     0     0
Establishment re-closed by DOHMH.                           1325     0     0
Establishment re-opened by DOHMH.                            200   915   377
No violations were recorded at the time of this...          1718     0     0
Not Available                                               3685     0     0
Violations were cited in the following area(s).           132039     0  6132

Chi-Square Statistic: 158885.33478075414

P-value: 0.0

Degrees of Freedom: 30

Expected Frequencies:
 [[3.33344164e+03 6.12951713e+02 4.22250993e+02 2.81778550e+02
  5.16569693e+03 3.17834533e+01 2.26096719e+02]
 [4.38436586e+02 8.06195175e+01 5.55372807e+01 3.70614035e+01
  6.79427083e+02 4.18037281e+00 2.97377558e+01]
 [6.29032415e+02 1.15666191e+02 7.96802797e+01 5.31726250e+01
  9.74785574e+02 5.99765185e+00 4.26652633e+01]
 [7.12087195e+02 1.30938265e+02 9.02009268e+01 6.01933135e+01
  1.10349214e+03 6.78955644e+00 4.82986042e+01]
 [1.21935005e+03 2.24213526e+02 1.54456513e+02 1.03072658e+02
  1.88957645e+03 1.16261689e+01 8.27046266e+01]
 [8.96326521e+04 1.64816108e+04 1.13538740e+04 7.57672145e+03
  1.38900022e+05 8.54622797e+02 6.07949703e+03]]


Reject H0 : ACTION and GRADE are Independent
```

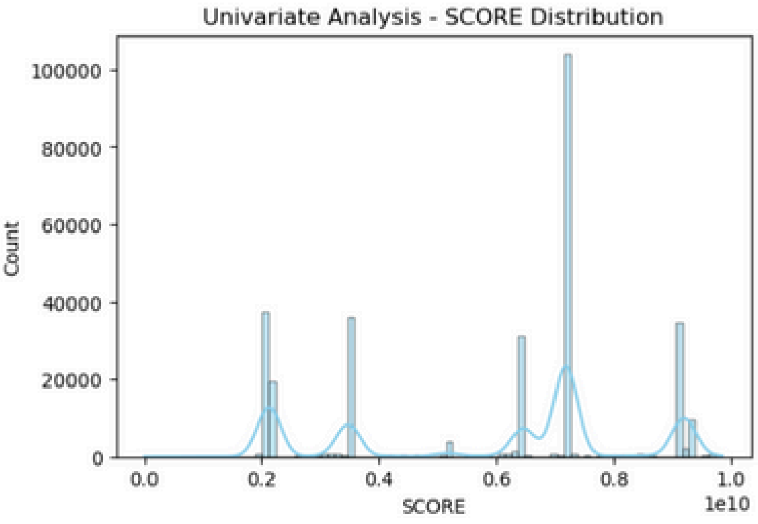
10.Exploratory Data Analysis- Univariate Analysis

1.Numeric Univariate Analysis - SCORE

- **Used Histplot for visualization**



```
plt.figure(figsize=(6,4))
sns.histplot(nyc_df["SCORE"], kde=True, color="skyblue")
plt.title("Univariate Analysis - SCORE Distribution")
plt.show()
```

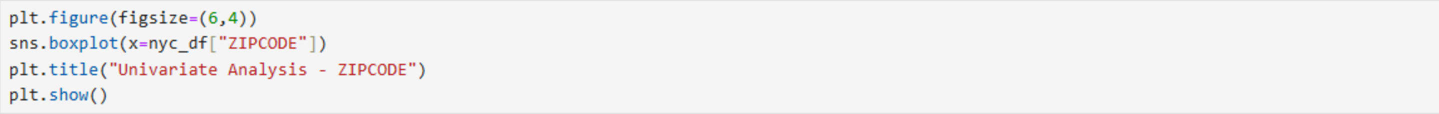


Insights :

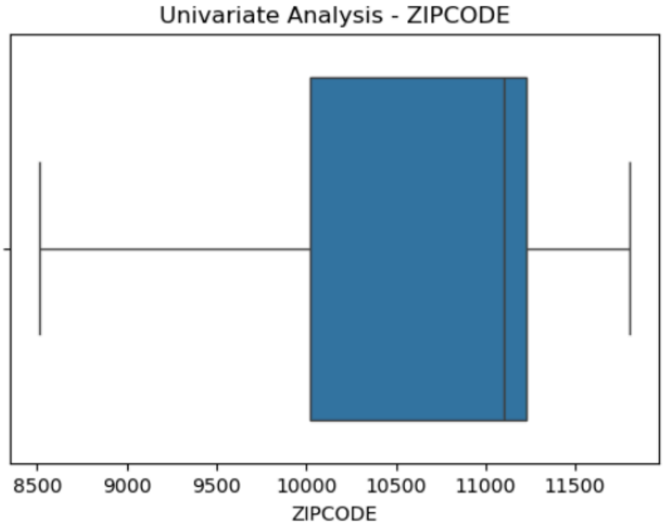
- The SCORE values appear to be highly skewed with extreme peaks at certain points, suggesting possible data entry errors or outliers.
- Instead of a smooth distribution, the plot shows spikes, which might mean scores are not continuous but clustered around specific values.

2.Categorical Analysis - ZIPCODE

- **Used Boxplot for visualization**



```
plt.figure(figsize=(6,4))
sns.boxplot(x=nyc_df["ZIPCODE"])
plt.title("Univariate Analysis - ZIPCODE")
plt.show()
```



Insights :

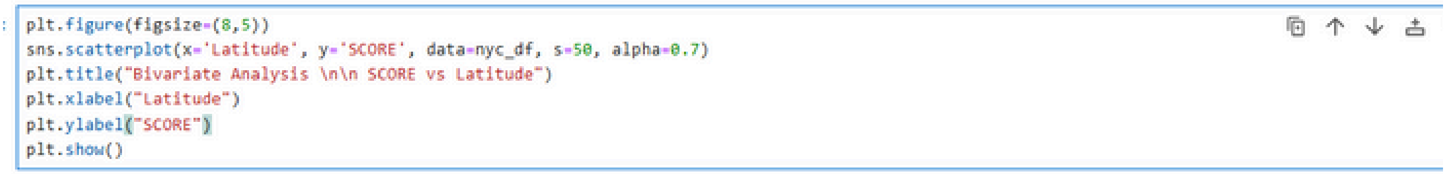
- The Median is around 11200
- The Interquartile Range IQR is between 10000 and 11400

11.Bivariate Analysis

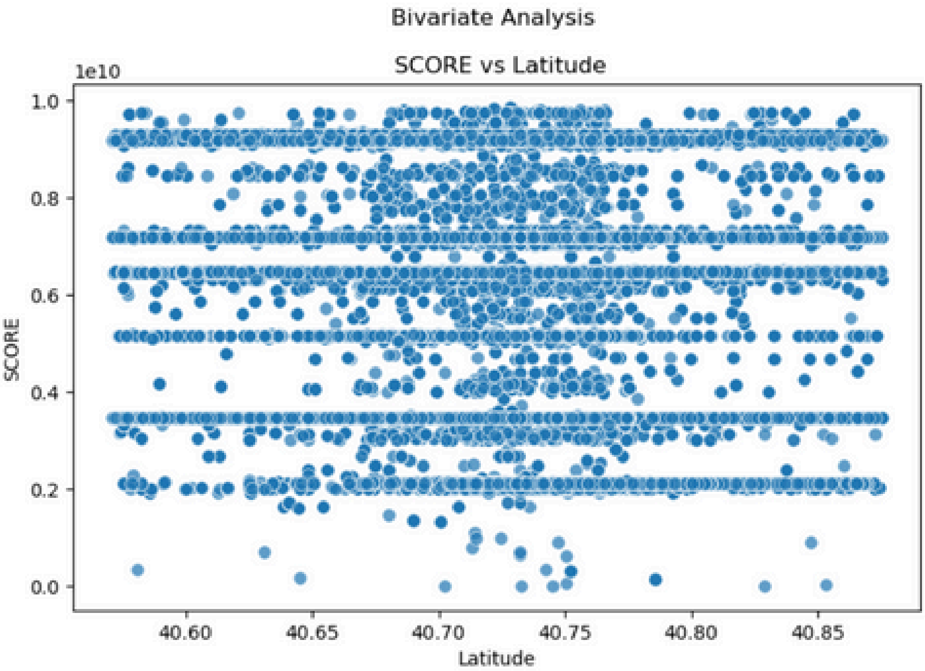
1.Numerical vs Numerical - Latitude and SCORE

- **Used Scatterplot for visualization**

Latitude and SCORE



```
plt.figure(figsize=(8,5))
sns.scatterplot(x='Latitude', y='SCORE', data=nyc_df, s=50, alpha=0.7)
plt.title("Bivariate Analysis \n\n SCORE vs Latitude")
plt.xlabel("Latitude")
plt.ylabel("SCORE")
plt.show()
```

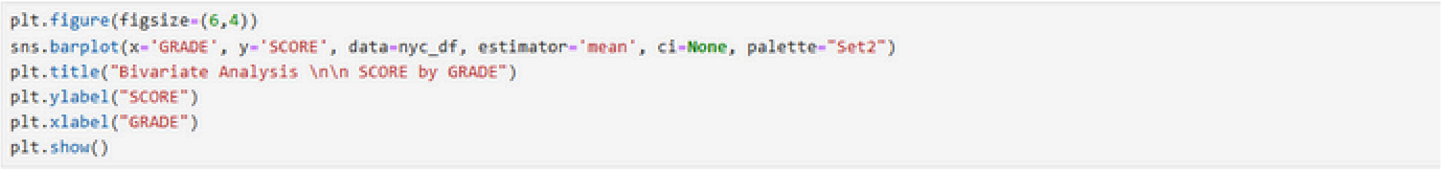


Insights :

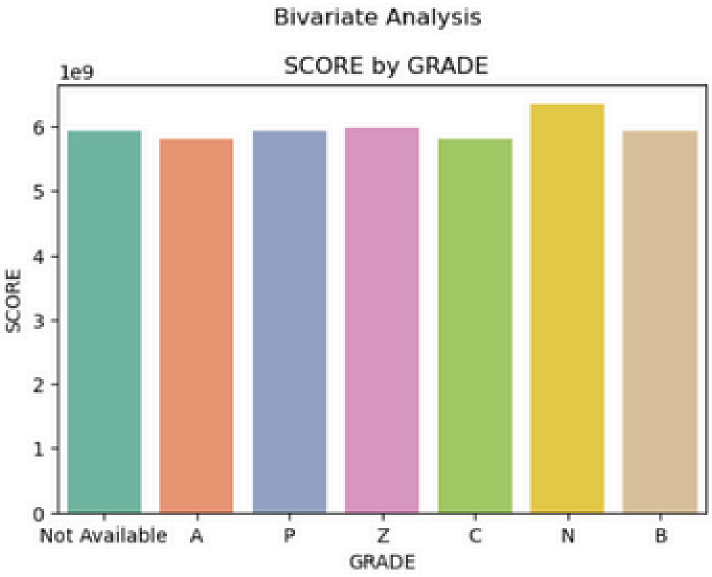
- SCORE is scattered in bands and does not vary strongly with Latitude, showing weak or no clear relationship.

2.Categorical vs Numerical - GRADE and SCORE

- **Used Barplot for visualization**



```
plt.figure(figsize=(6,4))
sns.barplot(x='GRADE', y='SCORE', data=nyc_df, estimator='mean', ci=None, palette="Set2")
plt.title("Bivariate Analysis \n\n SCORE by GRADE")
plt.ylabel("SCORE")
plt.xlabel("GRADE")
plt.show()
```



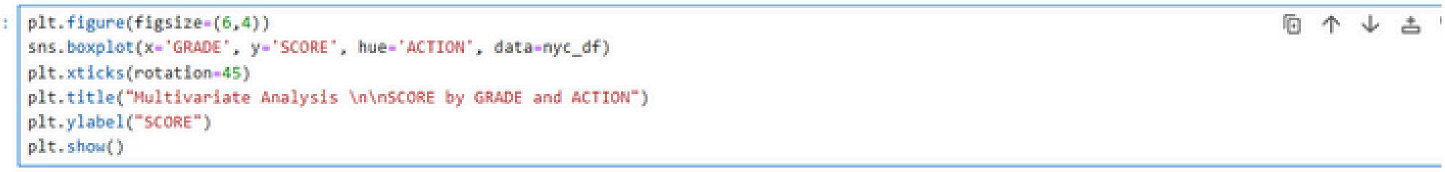
Insights :

- The SCORE increases with higher GRADE, showing that the grading system aligns well with inspection performance
- The SCORE is Minimum in GRADE C and the Maximum in GRADE N

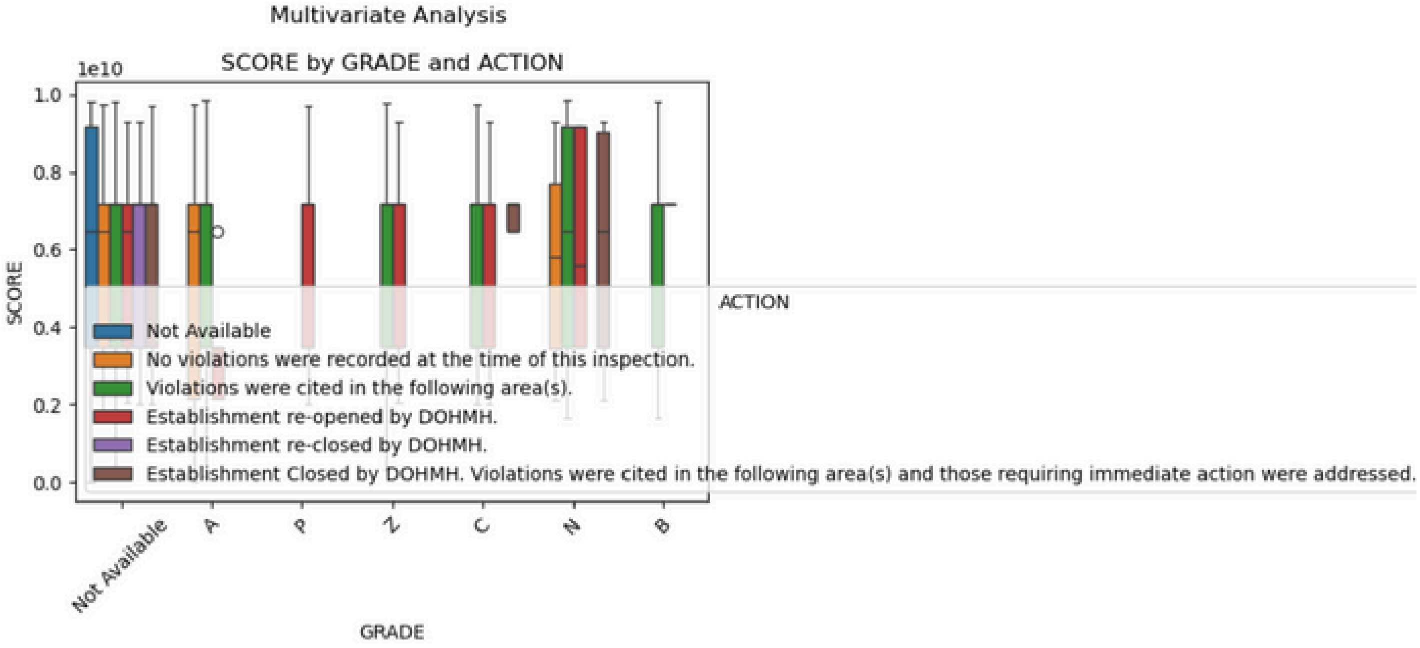
12.Multivariate Analysis

1.Numerical vs Two categorical

- **SCORE - Numerical , GRADE and ACTION - Categorical**
- **Used Boxplot for visualization**



```
plt.figure(figsize=(6,4))
sns.boxplot(x='GRADE', y='SCORE', hue='ACTION', data=nyc_df)
plt.xticks(rotation=45)
plt.title("Multivariate Analysis \n\nSCORE by GRADE and ACTION")
plt.ylabel("SCORE")
plt.show()
```

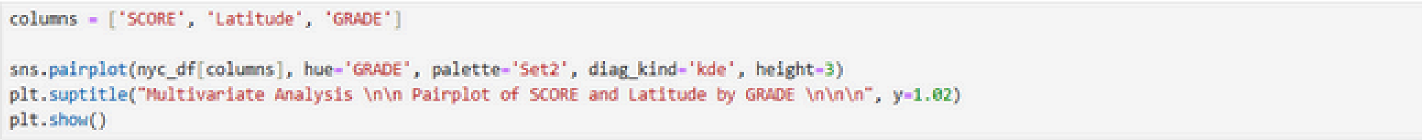


Insights :

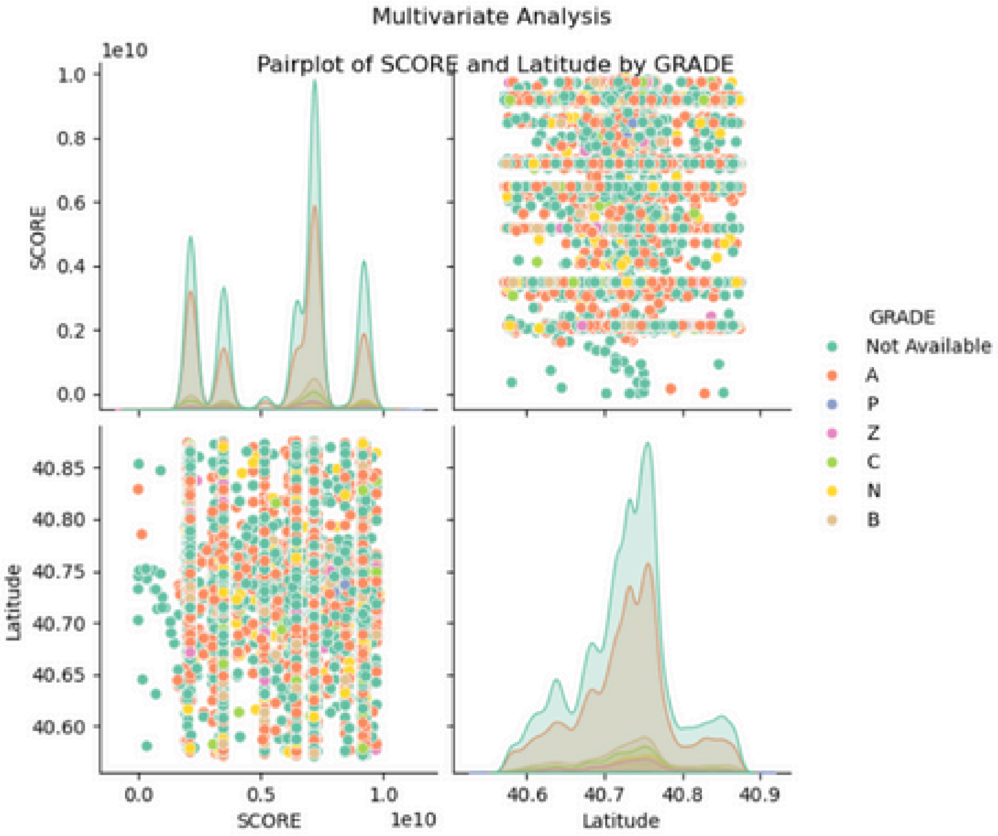
- Grade A has the lowest scores, indicating better compliance
- Grades B and C show higher scores and more severe actions
- Actions like “Closed” or “Re-closed” are linked to high violation scores, regardless of grade
- “Not Available” grade has wide score variation — possibly inconsistent or missing data

2.Two Numerical and one Categorical

- **SCORE and Latitude - Numerical , GRADE - Categorical**
- **Used Pairplot for visualization**



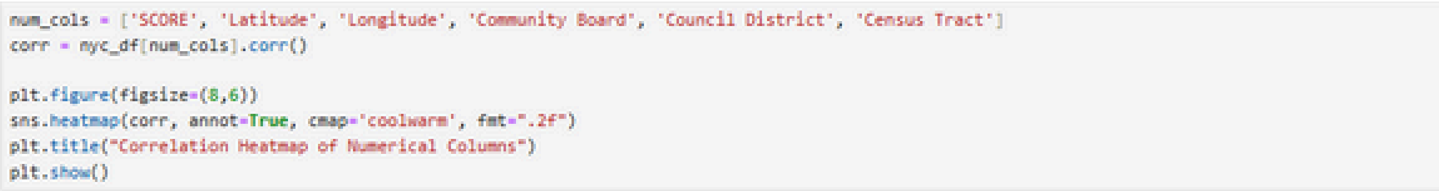
```
columns = ['SCORE', 'Latitude', 'GRADE']

sns.pairplot(nyc_df[columns], hue='GRADE', palette='Set2', diag_kind='kde', height=3)
plt.suptitle("Multivariate Analysis \n\n Pairplot of SCORE and Latitude by GRADE \n\n\n", y=1.02)
plt.show()
```



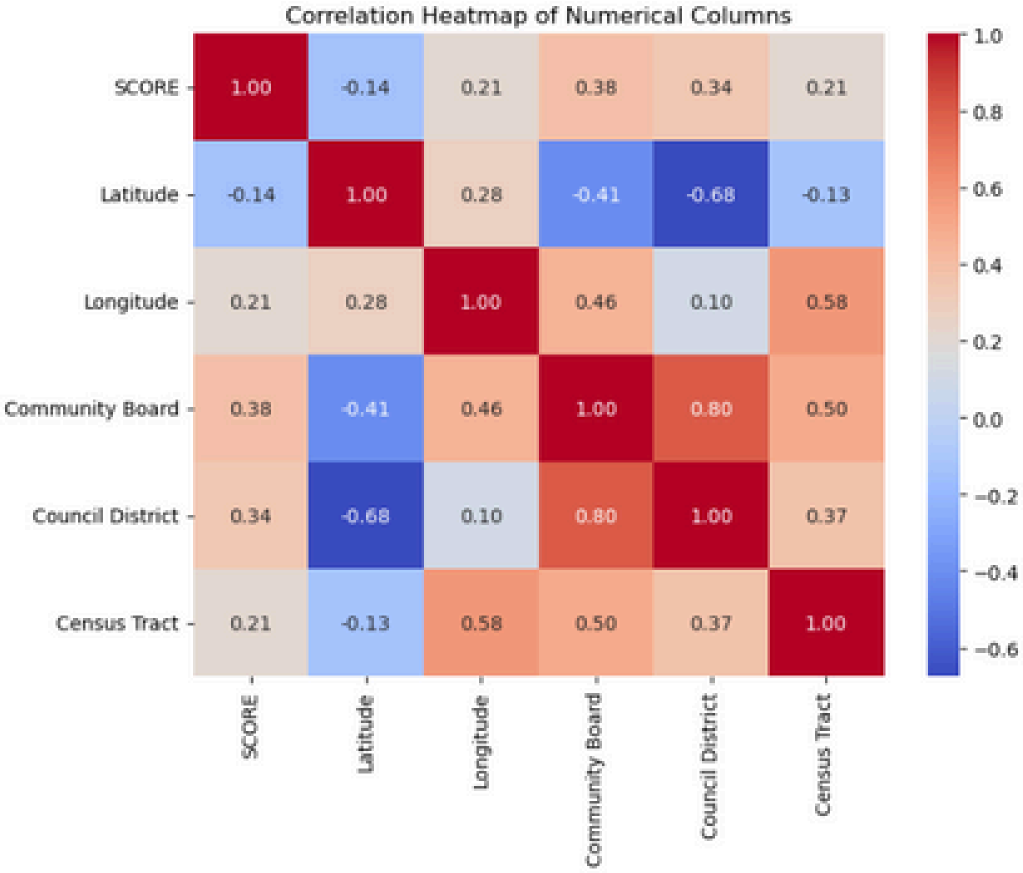
Insights :

- SCORE values are spread across different Grades without a clear pattern by Latitude
- Distribution shows multiple peaks in SCORE, suggesting clustering or categorical effects
- Grade categories are spread evenly, meaning GRADE may not strongly explain SCORE vs Latitude relationship

3.Correlation Heatmap



```
num_cols = ['SCORE', 'Latitude', 'Longitude', 'Community Board', 'Council District', 'Census Tract']
corr = nyc_df[num_cols].corr()

plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap of Numerical Columns")
plt.show()
```



Insights :

- SCORE shows weak to moderate positive correlation with Community Board (0.38) and Council District (0.34).
- Latitude has a strong negative correlation with Council District (-0.68).
- Community Board and Council District are highly correlated (0.80), which may indicate multicollinearity.

13. Overall Insights from analysis

- The distribution of SCORE is not normal and appears irregular.
- Values are clustered at specific points, creating sharp peaks in the histogram.
- Extremely large values close to 1e101e101e10 are present in the data.
- These large values are unrealistic for scores and likely represent invalid entries.
- Outliers dominate the distribution and affect the overall shape.
- Because of the skewness, the median is a better measure of central tendency than the mean.
- The spikes indicate possible data entry errors or inconsistent scaling.
- Cleaning the data is essential before performing further statistical tests.
- Summary statistics (min, max, percentiles) should be checked to confirm anomalies.
- Replotting after outlier removal will reveal the true underlying score distribution.

14. Conclusion

The exploratory data analysis revealed important insights into restaurant inspections and food safety compliance.

The SCORE distribution is highly skewed with extreme outliers and unrealistic values. The presence of sharp peaks and very large numbers suggests possible data entry errors or scaling issues. Using the median instead of the mean provides a more reliable central tendency, but meaningful insights cannot be derived until the data is properly cleaned.