Module Code & Module Title

CC5067NI Smart Data Discovery

60% Individual Coursework

Submission: Final Submission

Academic Semester: Spring Semester 2025

Credit: 15 credit semester long module

Student Name: Ishpa Maharjan

London Met ID: 23047616

College ID: NP01CP4A230045

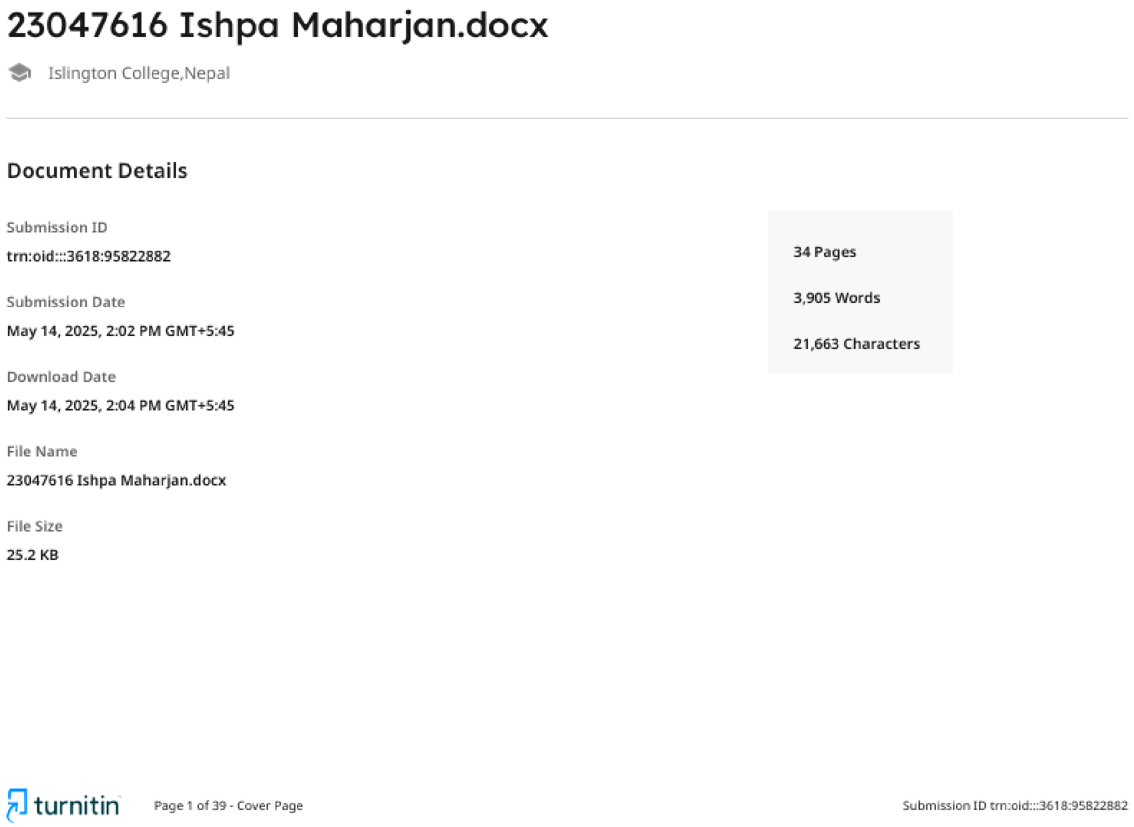
Assignment Due Date: Thursday, May 8, 2025

Assignment Submission Date: Thursday, May 15, 2025

Submitted To: Dipeshor Silwal

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Similarity Report



23047616 Ishpa Maharjan.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:95822882

Submission Date

May 14, 2025, 2:02 PM GMT+5:45

Download Date

May 14, 2025, 2:04 PM GMT+5:45

File Name

23047616 Ishpa Maharjan.docx

File Size

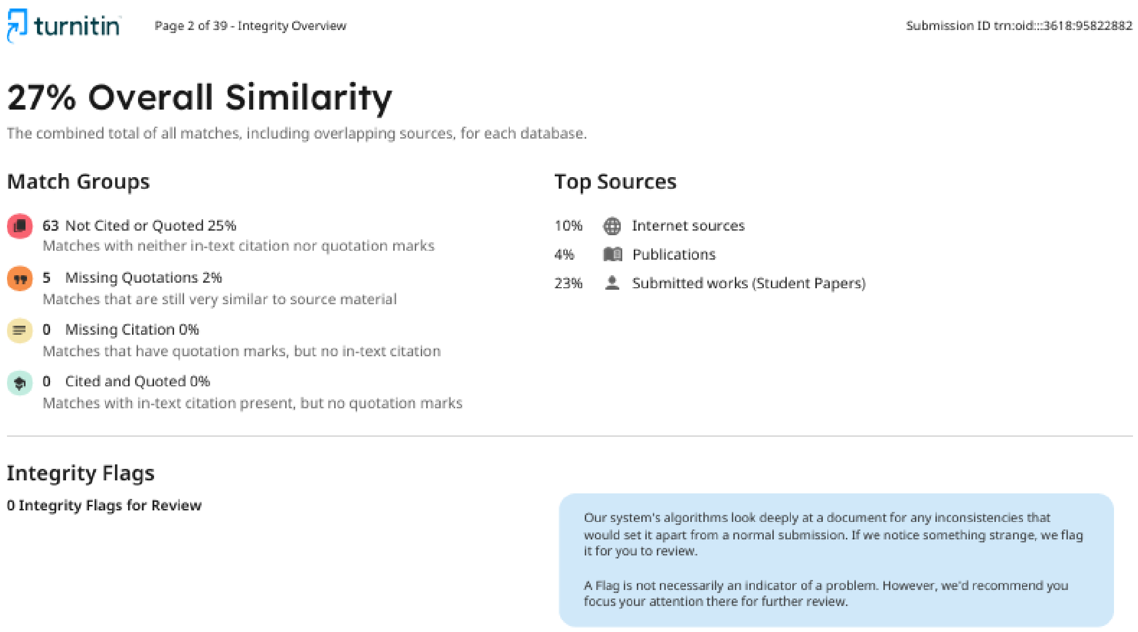
25.2 KB

34 Pages

3,905 Words

21,663 Characters

turnitin Page 1 of 39 - Cover Page Submission ID trn:oid:::3618:95822882



turnitin Page 2 of 39 - Integrity Overview Submission ID trn:oid:::3618:95822882

27% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- 63 Not Cited or Quoted 25%
  Matches with neither in-text citation nor quotation marks
- 5 Missing Quotations 2%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

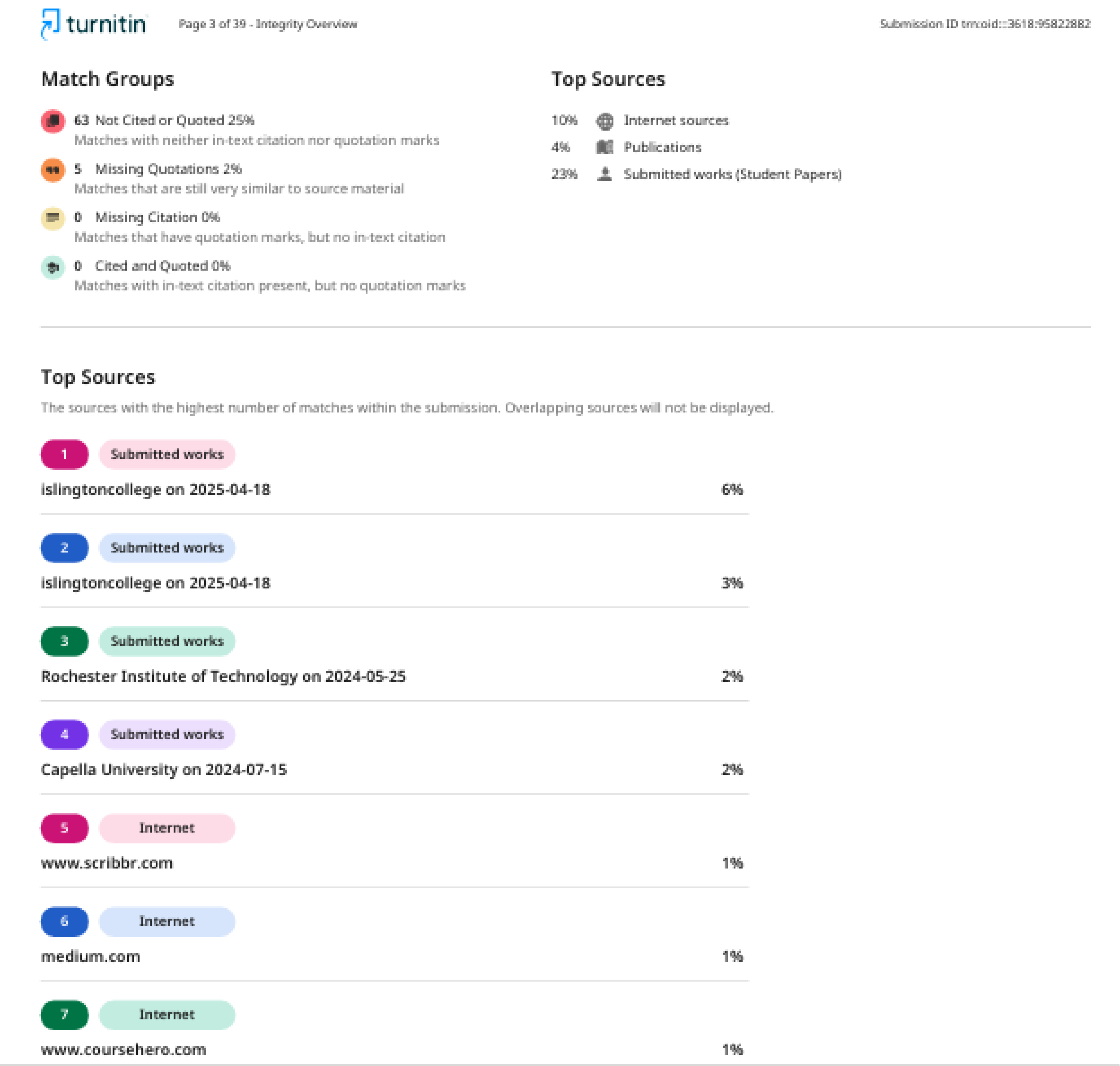
10%	Internet sources
4%	Publications
23%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.



turnitin Page 3 of 39 - Integrity Overview Submission ID trn:oid:::3618:95822882

Match Groups

- 63 Not Cited or Quoted 25%
  Matches with neither in-text citation nor quotation marks
- 5 Missing Quotations 2%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

10%	Internet sources
4%	Publications
23%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	islingtoncollege on 2025-04-18	6%
2	Submitted works	islingtoncollege on 2025-04-18	3%
3	Submitted works	Rochester Institute of Technology on 2024-05-25	2%
4	Submitted works	Capella University on 2024-07-15	2%
5	Internet	www.scribbr.com	1%
6	Internet	medium.com	1%
7	Internet	www.coursehero.com	1%

The similarity is shown in the key words such as column names and questions.

Table of Contents

1. Data Understanding 1

2. Data Preparation and Library Used 5

2.1 Data Preparation 5

2.2 Library Used 5

2.2.1 Jupyter Lab 5

2.2.2 Ms-Word 6

2.2.3 Anaconda Prompt: 7

2.3 Question 1: 8

2.4 Question 2: 10

2.5 Question 3: 11

2.6 Question 4: 12

3. Data Analysis 13

3.1 Question 1: 13

3.2 Question 2: 15

4. Data Exploration 16

4.1 Four major insights through visualization 17

4.2 Arrange the complaint types 25

5. Statistical Testing 27

5.1 Test 1: 27

5.2 Test 2: 30

6. Conclusion 32

7. References 33

Table of Figures

Figure 1: JupyterLab 6
Figure 2: Ms-Word..... 6
Figure 3: Anaconda..... 7
Figure 4: Importing dataset 8
Figure 5:Converting the columns "Created Date" and "Closed Date" to datetime datatype and create a new column "Request_Closing_Time" as the time elapsed between request creation and request closing. 9
Figure 6: Drop Columns. 10
Figure 7: Drop columns with null values..... 11
Figure 8: Unique values from the columns. 12
Figure 9: show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame. 13
Figure 10: Show correlation of all variables..... 15
Figure 11: Top 10 common complaint types in NYC. 17
Figure 12: Report of top 10 common complaint types in NYC..... 18
Figure 13: Distribution of complaints by borough. 19
Figure 14: Report of complaint distribution by borough. 20
Figure 15: Complaint trends over time. 21
Figure 16: Report of complaints trend over time (monthly)..... 22
Figure 17: Average request closing time by complaint type. 23
Figure 18: Report of Average request closing time by complaint type. 24
Figure 19: Arrange the complaint types..... 25
Figure 20: Report of arranging the complaint types. 26
Figure 21: Test 1 28
Figure 22: Test 1 Output 28
Figure 23: Test 2 31
Figure 24: Test 2 Output 31

Table of tables

Table 1: Columns Description and it's data types... 1

1. Data Understanding

Data Understanding refers to the summarization of the data by identifying key characteristics such as features, description, data types, values, etc. It is the phase of exploring the data, identifying any problems, and making the necessary adjustments to the data to ensure its quality and reliability. By identifying the issues early in the process, the data analyst can take the necessary steps to address them and ensure that the data is of the highest quality. In this coursework data understanding is a complete analysis of the dataset “Customer Service Request” of 311 customers in New York City. This dataset contains various records and has 53 columns, each representing a distinct service request submitted by a resident. These requests include various types of complaints such as Noise Issues, Blocked Driveway, Illegal Parking, Derelict Vehicle, Animal Abuse, Homeless Encampment, Urinating in Public, Drinking, Posting Advertisements, Vending, Traffic, Panhandling or Heating Problems. The attributes are provided in complaint type, location details, and agency information to timestamps such as request creation and closure dates. The table is generated below for the detailed information of all 53 columns with its description and data types (Yennhi95zz, 2023).

Table 1: Columns Description and it's data types

S.N	Column Name	Description	Data Type
1	Unique Key	It is a unique identifier assigned to 311 service requests.	Integer
2	Created Date	Date and time when the service request was created.	Datetime
3	Closed Date	Date and time when the service request was closed.	Datetime
4	Agency	Agency that handles the request.	String (Object)
5	Agency Name	Name of the agency.	String (Object)

6	Complaint Type	Type of the complaints such as noise, Blocked Driveway, Illegal Parking, etc.	String (Object)
7	Descriptor	Description of the complaint.	String (Object)
8	Location Type	Type of place where the incident occurred.	String (Object)
9	Incident Zip	ZIP code where the incident occurred.	Float
10	Incident Address	Address where the incident occurred.	String (Object)
11	Street Name	Name of the street involved in the incident.	String (Object)
12	Cross Street 1	Name of the nearest cross street to the incident location.	String (Object)
13	Cross Street 2	Name of the second nearest cross street to the incident location.	String (Object)
14	Intersection Street 1	First intersecting street near the incident location.	String (Object)
15	Intersection Street 2	Second intersecting street near the incident location.	String (Object)
16	Address Type	Type of the address.	String (Object)
17	City	Name of the city.	String (Object)
18	Landmark	Name of the landmark of the incident.	String (Object)
19	Facility Type	Type of facility related to the request.	String (Object)
20	Status	Current status of the service request.	String (Object)

21	Due Date	The last date by which issue should be resolved.	Datetime
22	Resolution Description	Description of the resolution of the issue.	String (Object)
23	Resolution Action Updated Date	Updated date if the resolved issue.	Datetime
24	Community Board	Community board district where the incident was reported.	String (Object)
25	Borough	The borough where the incident took place.	String (Object)
26	X Coordinate (State Plane)	X coordinate in NYC's state plane coordinate system.	Float
27	Y Coordinate (State Plane)	Y coordinate in NYC's state plane coordinate system.	Float
28	Park Facility Name	Name of the park facility.	String (Object)
29	Park Borough	Borough where the park is located.	String (Object)
30	School Name	Name of the school.	String (Object)
31	School Number	Number assigned to the school.	Integer
32	School Region	Region code for school.	String (Object)
33	School Code	Code of the school.	String (Object)
34	School Phone Number	School contact number.	Integer
35	School Address	Address of the school.	String (Object)
36	School City	City where the school is located.	String (Object)
37	School State	State of the school.	String (Object)
38	School Zip	ZIP Code of the school.	Integer

39	School Not Found	Indicates whether the school was found in the database.	String (Object)
40	School or Citywide Complaint	Specifies whether the complaint is about a school or a citywide issue.	String (Object)
41	Vehicle Type	Type of vehicle	String (Object)
42	Taxi Company Borough	The borough of the taxi company.	String (Object)
43	Taxi Pick Up Location	Pickup location of the taxi.	String (Object)
44	Bridge Highway Name	Name of the bridge or highway.	String (Object)
45	Bridge Highway Direction	Direction of the bridge or highway.	String (Object)
46	Road Ramp	Specific road ramp involved in the incident.	String (Object)
47	Bridge Highway Segment	Segment of the bridge or highway.	String (Object)
48	Garage Lot Name	Name of the garage or parking lot.	String (Object)
49	Ferry Direction	Direction of the ferry.	String (Object)
50	Ferry Terminal Name	Name of Ferry terminal.	String (Object)
51	Latitude	Latitude coordinate of the complaint location.	Float
52	Longitude	Longitude coordinate of the complaint location	Float
53	Location	Combined latitude/longitude coordinate of the incident.	String (Object)

2. Data Preparation and Library Used

2.1 Data Preparation

Data Preparation is the process of preparing the raw data by analyzing, processing, exploring and visualizing the data. A data is prepared by cleaning, validating, labeling, gathering and visualizing the data. It comprises importing the dataset, converting the "Created Date" and "Closed Date" columns to datetime format and inserting a new column "Request_Closing_Time" as the difference in time between request closing and request creation. It is made to find the time spent in addressing each request. This data has more than one record and contains 53 columns, each of which contains a distinct service request made by a citizen. These requests cover various types of complaints such as Noise Issues, Blocked Driveway, Illegal Parking, Derelict Vehicle, Animal Abuse, Homeless Encampment, Urinating in Public, Drinking, Posting Advertisements, Vending, Traffic, Panhandling or Heating Problems. According to the question we drop some columns such as address, school-related columns, and transport. Identify and handle the missing values by dropping rows containing null values from the data set. In another question unique or distinct values of all columns are examined to understand the distribution of data (AWS, 2025).

2.2 Library Used

2.2.1 Jupyter Lab

Project Jupyter is a large umbrella project that contains a lot of separate software offerings and tools. That includes Jupyter Notebook and JupyterLab, both extremely popular notebook-editor programs. The Jupyter project, and spin-off projects, are all focused on providing tools and specifications for interactive computing with computational notebooks. If you aren't familiar with some of those, you can peruse a guided tour below explaining each one from start. The term "Jupyter" is typically abbreviated as a reference to one of those products or ideas (jupyter.org, 2015).

Figure 1: JupyterLab

(JupyterLab, 2023)

2.2.2 Ms-Word

Microsoft Word is one component of Microsoft Office that helps in writing, revising, and designing documents. It has spell and grammar check tools and text formatting. You can further insert pictures, tables, and charts into your documents. When doing tasks like formal letters, creating Signature in Microsoft Word can be a fast and secure way of signing off your documents. It's employed for letters, reports, and CVs. Word enables many people to work on a document simultaneously, monitoring changes. It provides templates to make document preparation easier. Word is used by users, organisations, and schools because it is easy to use and has many useful features. For visual storytelling, Microsoft Sway can be employed as a complement to Word for creating rich, multimedia presentations (Roberts, 2025).

Figure 2: Ms-Word

(techmodena.com, 2022)

2.2.3 Anaconda Prompt:

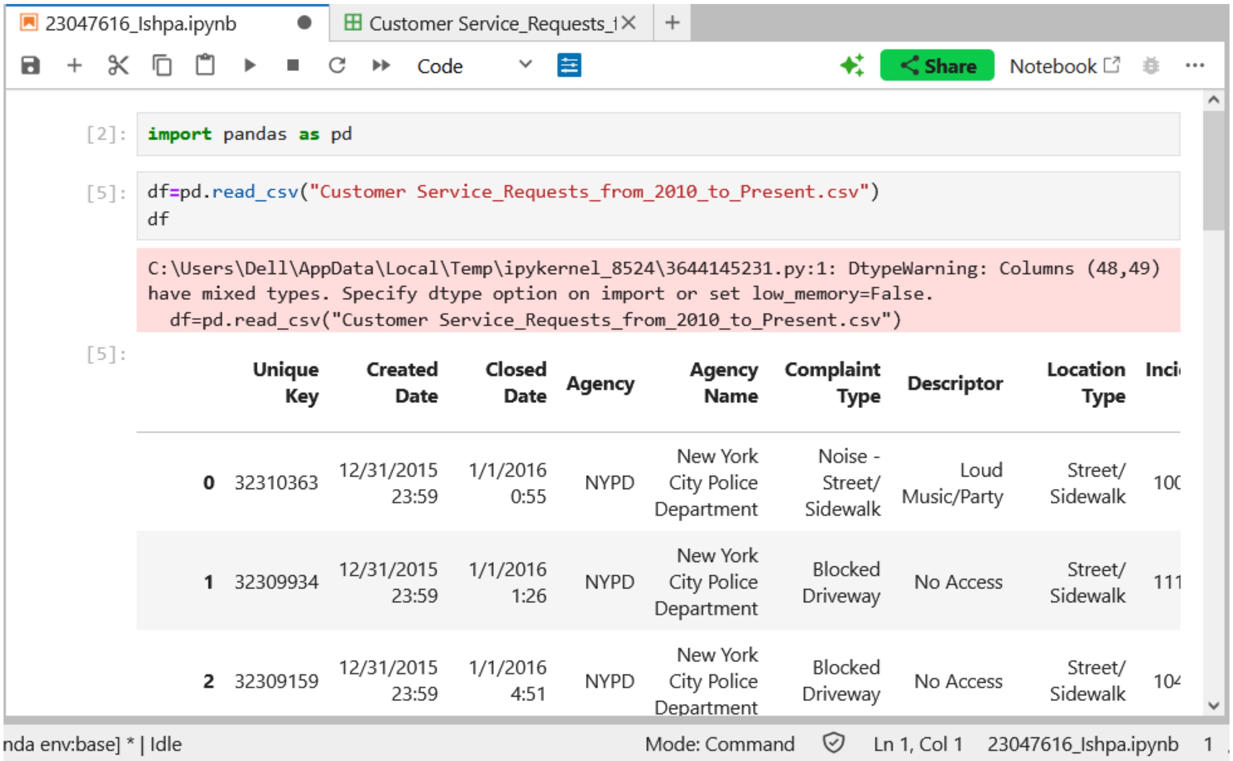
Anaconda is an open-source R and Python programming tools distribution for data science. It simplifies package deployment and management by providing a wide range of software and libraries that make users' work easier in scientific computing, machine learning, and data science projects (Dey, 2023).

Figure 3: Anaconda

(Dey, 2023)

2.3 Question 1:

Convert the columns "**Created Date**" and "**Closed Date**" to datetime datatype and create a new column "**Request_Closing_Time**" as the time elapsed between request creation and request closing.



```
[2]: import pandas as pd
```

```
[5]: df=pd.read_csv("Customer Service_Requests_from_2010_to_Present.csv")
     df
```

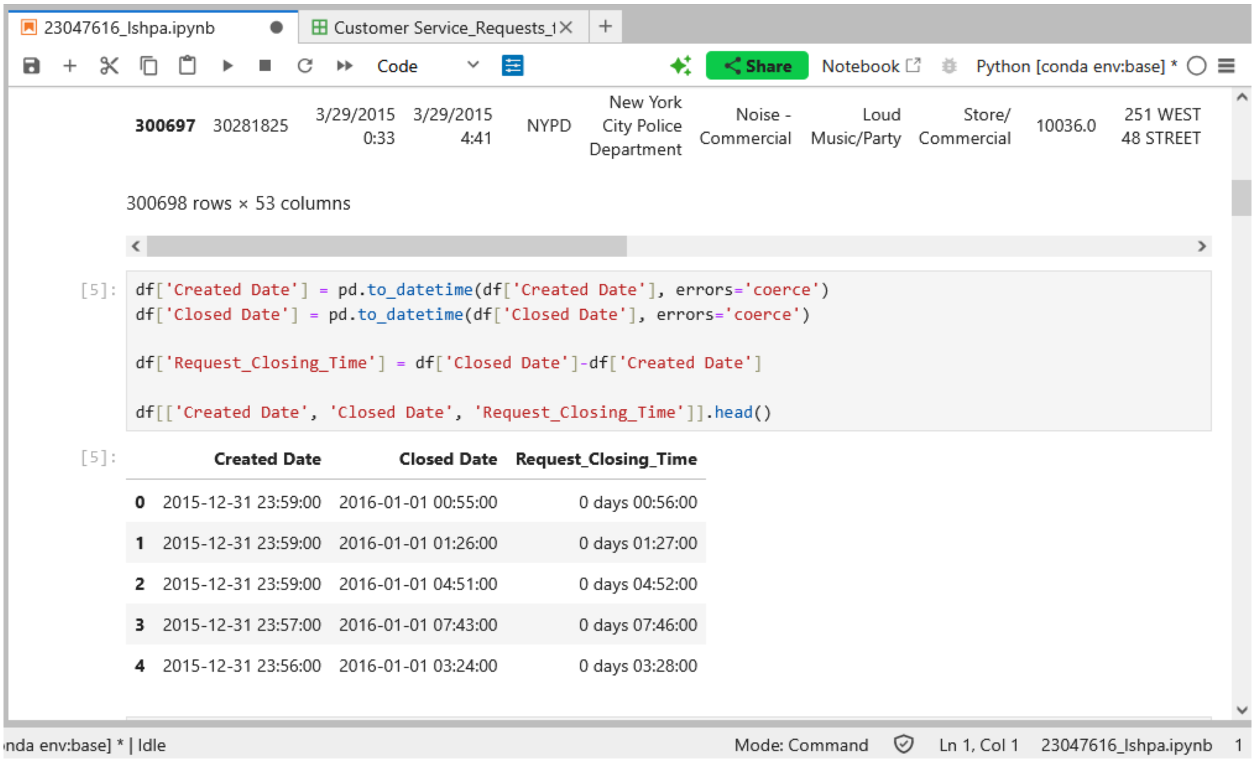
```
C:\Users\Dell\AppData\Local\Temp\ipykernel_8524\3644145231.py:1: DtypeWarning: Columns (48,49)
have mixed types. Specify dtype option on import or set low_memory=False.
  df=pd.read_csv("Customer Service_Requests_from_2010_to_Present.csv")
```

[5]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Inci
0	32310363	12/31/2015 23:59	1/1/2016 0:55	NYPD	New York City Police Department	Noise - Street/ Sidewalk	Loud Music/Party	Street/ Sidewalk	10
1	32309934	12/31/2015 23:59	1/1/2016 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/ Sidewalk	111
2	32309159	12/31/2015 23:59	1/1/2016 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/ Sidewalk	10

Figure 4: Importing dataset

In figure 1, the Customer_Service_Request.csv dataset is imported.



300697	30281825	3/29/2015 0:33	3/29/2015 4:41	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Store/ Commercial	10036.0	251 WEST 48 STREET

300698 rows × 53 columns

```
[5]: df['Created Date'] = pd.to_datetime(df['Created Date'], errors='coerce')
     df['Closed Date'] = pd.to_datetime(df['Closed Date'], errors='coerce')

     df['Request_Closing_Time'] = df['Closed Date']-df['Created Date']

     df[['Created Date', 'Closed Date', 'Request_Closing_Time']].head()
```

[5]:

	Created Date	Closed Date	Request_Closing_Time
0	2015-12-31 23:59:00	2016-01-01 00:55:00	0 days 00:56:00
1	2015-12-31 23:59:00	2016-01-01 01:26:00	0 days 01:27:00
2	2015-12-31 23:59:00	2016-01-01 04:51:00	0 days 04:52:00
3	2015-12-31 23:57:00	2016-01-01 07:43:00	0 days 07:46:00
4	2015-12-31 23:56:00	2016-01-01 03:24:00	0 days 03:28:00

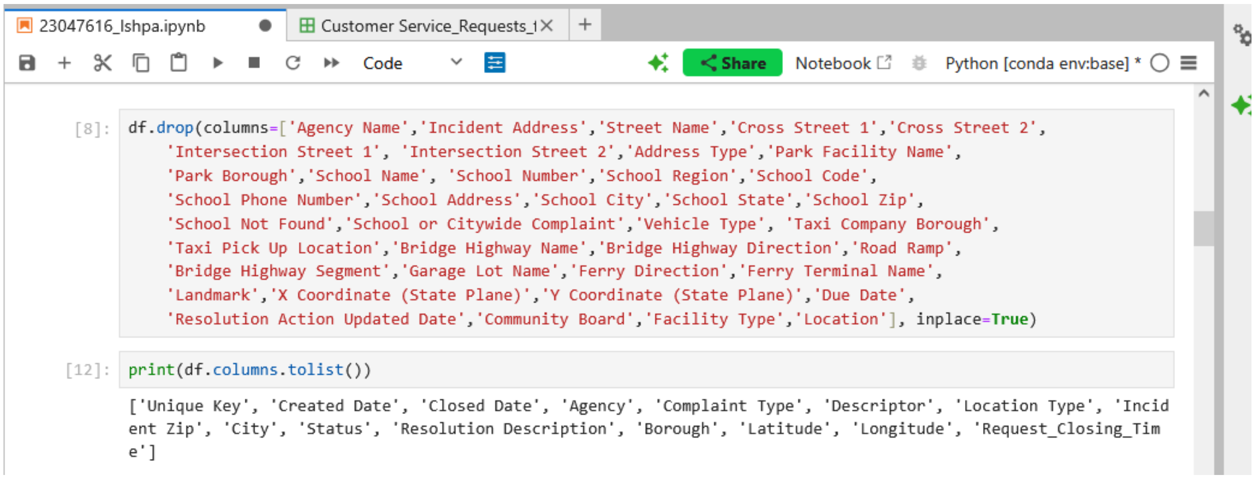
Figure 5:Converting the columns "Created Date" and "Closed Date" to datetime datatype and create a new column "Request_Closing_Time" as the time elapsed between request creation and request closing.

In figure 2, the created date and closed date is converted into datetime datatype where to_datetime() is a convenient method that directly handles the integer dates by specifying the format of the integer. It is easier since no string conversion is needed. The code uses pandas to_datetime() function, sending the integer as a string to call the function. It uses the format argument of %y%m%d, to_datetime() which correctly interprets the integer as a date. It returns a pandas Timestamp object, which native Python datetime objects are compatible with (Collins, 2024).

2.4 Question 2:

Write a python program to drop irrelevant Columns which are listed below.

['Agency Name','Incident Address','Street Name','Cross Street 1','Cross Street 2','Intersection Street 1', 'Intersection Street 2','Address Type','Park Facility Name','Park Borough','School Name', 'School Number','School Region','School Code','School Phone Number','School Address','School City', 'School State','School Zip','School Not Found','School or Citywide Complaint','Vehicle Type', 'Taxi Company Borough','Taxi Pick Up Location','Bridge Highway Name','Bridge Highway Direction', 'Road Ramp','Bridge Highway Segment','Garage Lot Name','Ferry Direction','Ferry Terminal Name','Landmark', 'X Coordinate (State Plane)','Y Coordinate (State Plane)','Due Date','Resolution Action Updated Date','Community Board','Facility Type', 'Location']



```
[8]: df.drop(columns=['Agency Name','Incident Address','Street Name','Cross Street 1','Cross Street 2',
         'Intersection Street 1', 'Intersection Street 2','Address Type','Park Facility Name',
         'Park Borough','School Name', 'School Number','School Region','School Code',
         'School Phone Number','School Address','School City','School State','School Zip',
         'School Not Found','School or Citywide Complaint','Vehicle Type', 'Taxi Company Borough',
         'Taxi Pick Up Location','Bridge Highway Name','Bridge Highway Direction','Road Ramp',
         'Bridge Highway Segment','Garage Lot Name','Ferry Direction','Ferry Terminal Name',
         'Landmark','X Coordinate (State Plane)','Y Coordinate (State Plane)','Due Date',
         'Resolution Action Updated Date','Community Board','Facility Type','Location'], inplace=True)
```

```
[12]: print(df.columns.tolist())
```

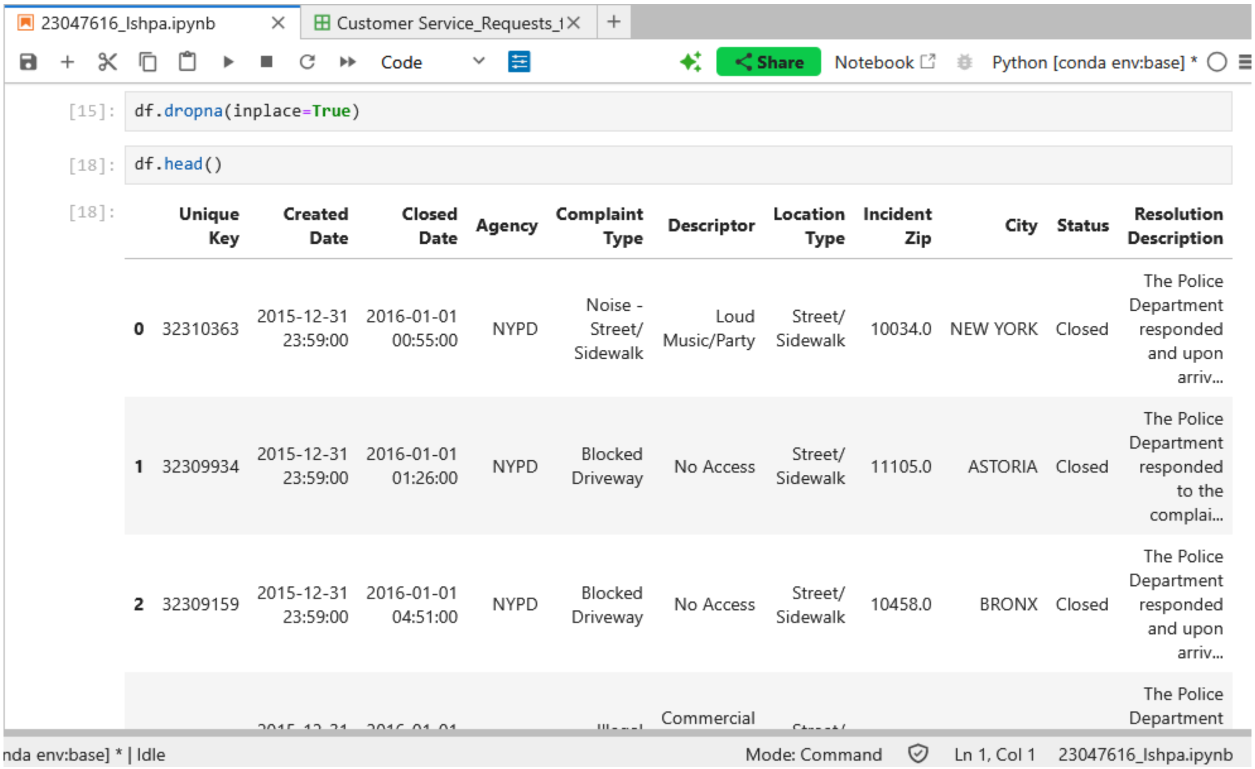
```
['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Complaint Type', 'Descriptor', 'Location Type', 'Incid
ent Zip', 'City', 'Status', 'Resolution Description', 'Borough', 'Latitude', 'Longitude', 'Request_Closing_Tim
e']
```

Figure 6: Drop Columns.

In figure 3, the irrelevant columns are dropped. By removing the irrelevant columns the redundant and unnecessary data are removed which reduces the size if the dataset and the complexity of computation for the machine learning algorithms. It helps to improve the accuracy, performance to enhance the machine learning models (ajaymehta, 2023).

2.5 Question 3:

Write a python program to remove the NaN missing values from updated dataframe.



```
[15]: df.dropna(inplace=True)
```

```
[18]: df.head()
```

[18]:

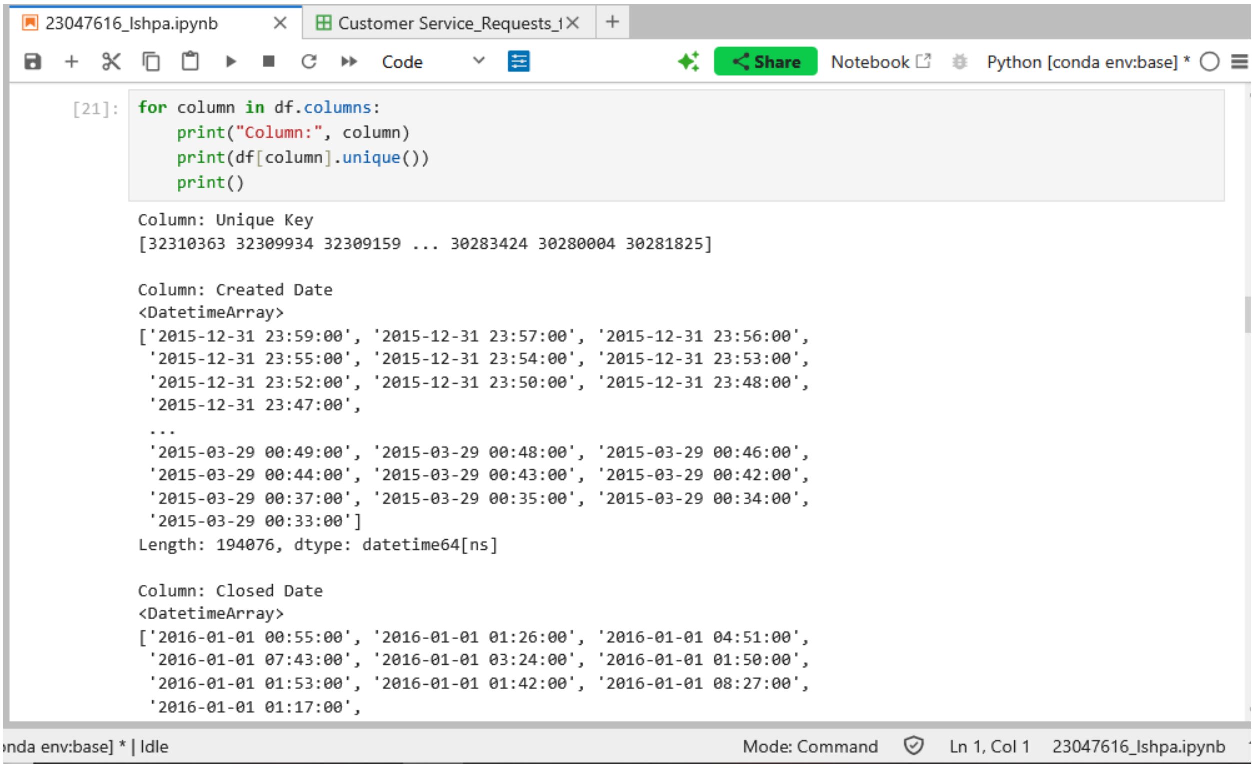
	Unique Key	Created Date	Closed Date	Agency	Complaint Type	Descriptor	Location Type	Incident Zip	City	Status	Resolution Description
0	32310363	2015-12-31 23:59:00	2016-01-01 00:55:00	NYPD	Noise - Street/ Sidewalk	Loud Music/Party	Street/ Sidewalk	10034.0	NEW YORK	Closed	The Police Department responded and upon arriv...
1	32309934	2015-12-31 23:59:00	2016-01-01 01:26:00	NYPD	Blocked Driveway	No Access	Street/ Sidewalk	11105.0	ASTORIA	Closed	The Police Department responded to the complai...
2	32309159	2015-12-31 23:59:00	2016-01-01 04:51:00	NYPD	Blocked Driveway	No Access	Street/ Sidewalk	10458.0	BRONX	Closed	The Police Department responded and upon arriv...
		2015-12-31	2016-01-01		Illegal	Commercial	Street/				The Police Department

Figure 7: Drop columns with null values

In figure 4, the rows with missing values are removed in order to avoid the errors. If a row contains NaN in any column, the whole row must be removed. Here, dropna() function is used to remove the missing values from a dataframe. It can remove rows and columns containing NaN values based on some conditions (geeksforgeeks, 2025).

2.6 Question 4:

Write a python program to see the unique values from all the columns in the dataframe.



```
[21]: for column in df.columns:
          print("Column:", column)
          print(df[column].unique())
          print()
```

```
Column: Unique Key
[32310363 32309934 32309159 ... 30283424 30280004 30281825]

Column: Created Date
<DatetimeArray>
['2015-12-31 23:59:00', '2015-12-31 23:57:00', '2015-12-31 23:56:00',
 '2015-12-31 23:55:00', '2015-12-31 23:54:00', '2015-12-31 23:53:00',
 '2015-12-31 23:52:00', '2015-12-31 23:50:00', '2015-12-31 23:48:00',
 '2015-12-31 23:47:00',
 ...
 '2015-03-29 00:49:00', '2015-03-29 00:48:00', '2015-03-29 00:46:00',
 '2015-03-29 00:44:00', '2015-03-29 00:43:00', '2015-03-29 00:42:00',
 '2015-03-29 00:37:00', '2015-03-29 00:35:00', '2015-03-29 00:34:00',
 '2015-03-29 00:33:00']
Length: 194076, dtype: datetime64[ns]

Column: Closed Date
<DatetimeArray>
['2016-01-01 00:55:00', '2016-01-01 01:26:00', '2016-01-01 04:51:00',
 '2016-01-01 07:43:00', '2016-01-01 03:24:00', '2016-01-01 01:50:00',
 '2016-01-01 01:53:00', '2016-01-01 01:42:00', '2016-01-01 08:27:00',
 '2016-01-01 01:17:00',
```

Figure 8: Unique values from the columns.

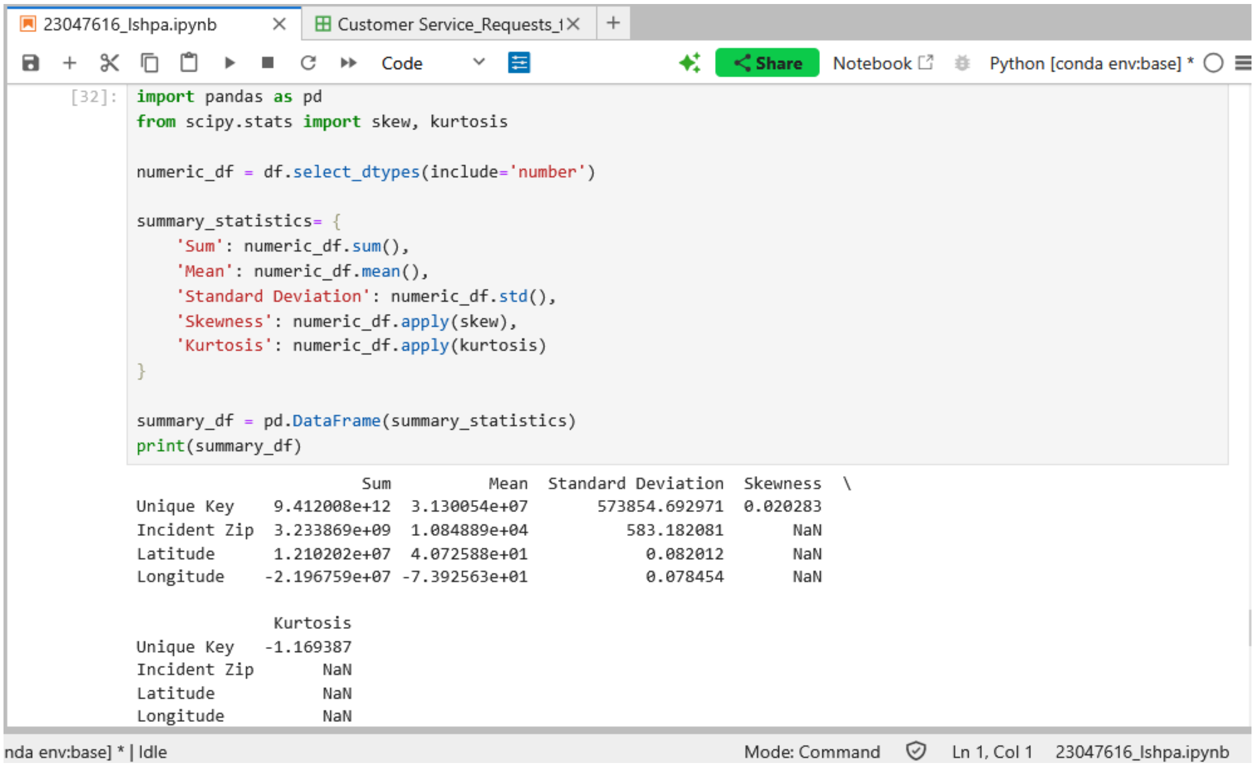
In figure 5, the unique values are shown by using .unique(0 method to get the unique values in the pandas dataframe column. The unique() method in pandas itself does not take any arguments. It is called on a specific column of a dataframe which returns an array of unique values in the called column. By default, he pandas .unique method returns a NumPy array of unique values (datagy, 2023).

3. Data Analysis

Data Analysis refers to the activity of gathering, analyzing and organizing the data for making decisions. The step that includes the statistical analyses to present the summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame. This step includes calculation and present the correlation of all the variables. In this step correlation is being calculated to identify the different numerical variables relate to one another (Coursera, 2025).

3.1 Question 1:

Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame.



```
[32]: import pandas as pd
      from scipy.stats import skew, kurtosis

      numeric_df = df.select_dtypes(include='number')

      summary_statistics= {
          'Sum': numeric_df.sum(),
          'Mean': numeric_df.mean(),
          'Standard Deviation': numeric_df.std(),
          'Skewness': numeric_df.apply(skew),
          'Kurtosis': numeric_df.apply(kurtosis)
      }

      summary_df = pd.DataFrame(summary_statistics)
      print(summary_df)
```

```
                      Sum          Mean  Standard Deviation  Skewness  \
Unique Key   9.412008e+12  3.130054e+07       573854.692971  0.020283   
Incident Zip 3.233869e+09  1.084889e+04          583.182081       NaN   
Latitude     1.210202e+07  4.072588e+01            0.082012       NaN   
Longitude   -2.196759e+07 -7.392563e+01            0.078454       NaN   

              Kurtosis  
Unique Key   -1.169387  
Incident Zip       NaN  
Latitude           NaN  
Longitude          NaN  
```

Figure 9: show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame.

In figure 6, summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame is shown. The summary statistics provide a summary and information about your sample data. It tells you something about the values of your data set. This includes where the mean is at and whether your data is skewed (J.Hand, 2013). Summary statistics fall into three broad categories:

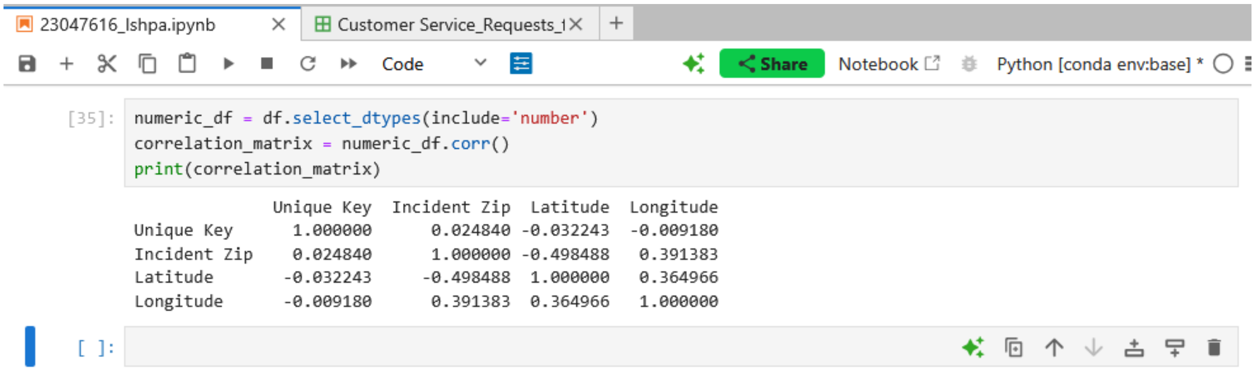
- Measures of location (also known as central tendency).
- Measures of spread.
- Charts/graphs.

Skewness: Skewness is a way to measure the asymmetry of a distribution. A distribution is asymmetrical if it is not a mirror image of its left and right side. A distribution can be right (or positive), left (or negative), or have zero skewness. A right-skewed distribution has more length on the right side of its peak, and a left-skewed distribution has more length on the left side of its peak (Turney, 2022).

Kurtosis: Kurtosis is a measure of the tailedness of a distribution in statistics. It is the frequency of outliers. Excess kurtosis is the tailedness of a distribution relative to the normal distribution. The tails are the narrowing portions on both sides of a distribution. They represent the probability or frequency of values that are extremely high or low compared to the mean. In other words, tails represent the frequency of outliers (Turney, 2022).

3.2 Question 2:

Write a Python program to calculate and show correlation of all variables.



```
[35]: numeric_df = df.select_dtypes(include='number')
      correlation_matrix = numeric_df.corr()
      print(correlation_matrix)
```

```
              Unique Key  Incident Zip  Latitude  Longitude
Unique Key      1.000000      0.024840 -0.032243  -0.009180
Incident Zip    0.024840      1.000000 -0.498488   0.391383
Latitude       -0.032243     -0.498488  1.000000   0.364966
Longitude      -0.009180      0.391383  0.364966   1.000000
```

```
[ ]:
```

Figure 10: Show correlation of all variables.

A statistical measure helping to analyze the correlation between two variables is known as Correlation. It also helps in analyzing the economic behaviour of the variables. The level of correlation between two or more variables can be determined with the help of correlation (geeksforgeeks, 2024). Here, correlation is calculated between numerical variables to identify relationships.

4. Data Exploration

Data exploration is the statistical process through which you can see the distribution of your data, and observe what statistical tests would be most beneficial. Effective data exploration can enable you to become familiar with the properties of your data when you're deciding what type of analysis or interpretation will be needed to generate meaningful, useful results (coursera, 2024).

There are various types of data exploration:

- **Descriptive analysis:**

  Descriptive analysis is the use and interpretation of historical data to identify its overall trends, patterns, and characteristics. descriptive analysis gives you an overview of the data and key features. You could think of frequency counts within ranges of grades, the average grade, outliers, range of grades, and the number of students you have information on (coursera, 2024).

- **Visual analysis:**

  Visual analysis helps in visualizing the trends, distribution pattern, outliers, and tendencies of your data. It benefits from it if you have large and complex data sets that are difficult to interpret using numbers alone. You can see the bigger picture of your data utilizing visual analysis through creating visual descriptions like graphs, charts, and plots. It is also helpful while communicating information regarding your data to a large group of people because you can provide a general overview (coursera, 2024).

- **Statistical analysis:**

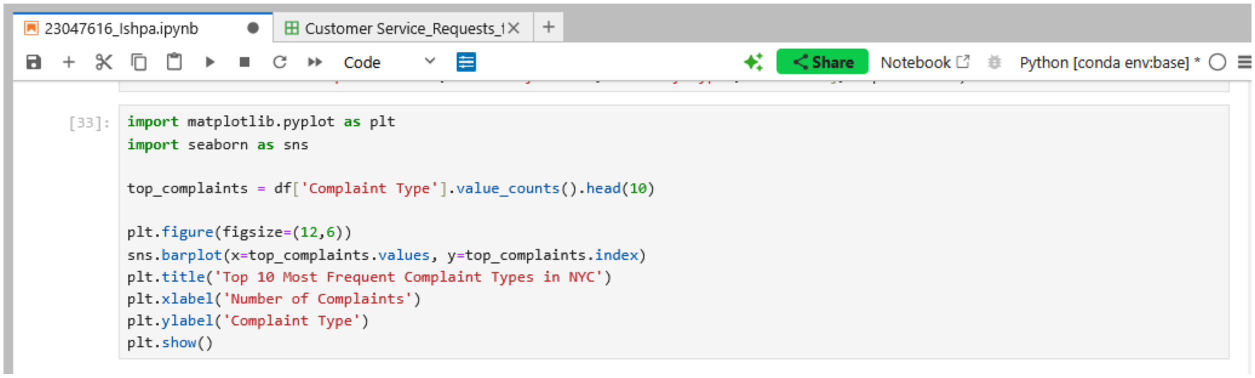
  Statistical analysis helps in providing a better understanding of the data using mathematics which can be used to carry out mean, median, correlations, test hypotheses, collinearity, independence of variables, and understand the probability of particular outcomes (coursera, 2024).

4.1 Four major insights through visualization

The four major insights through visualization that you come up after data mining are:

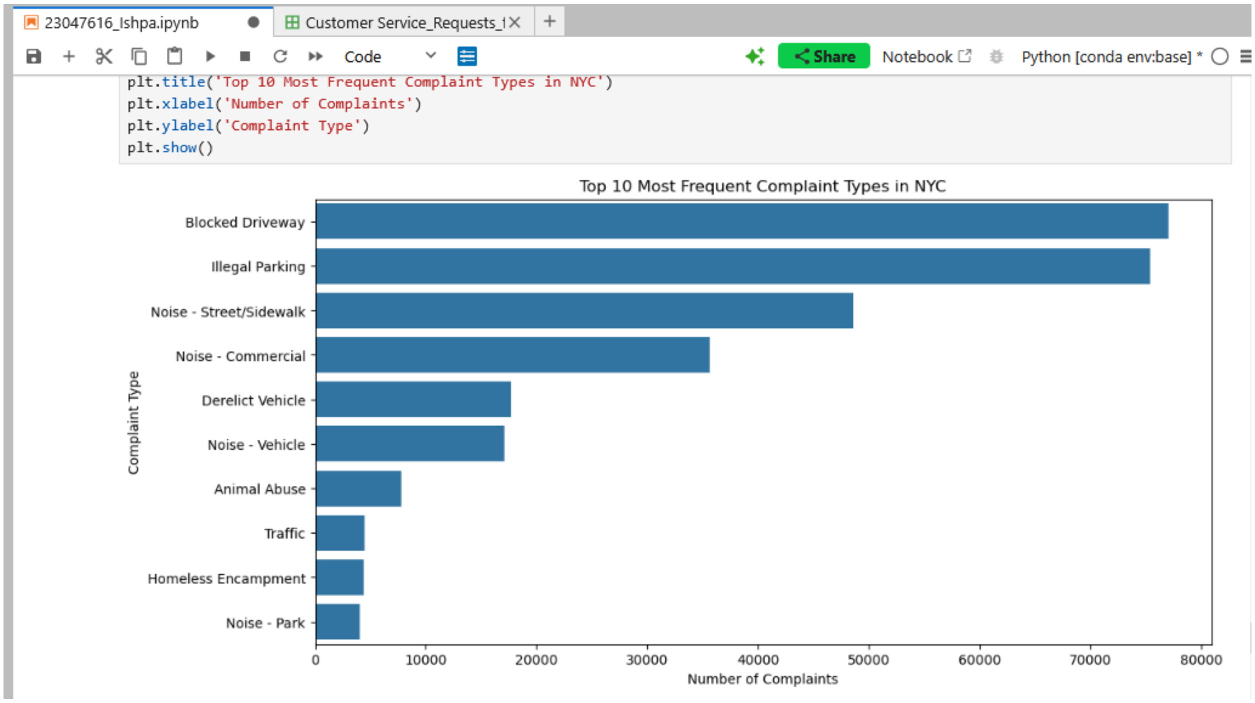
1. Top 10 most common complaint types:

The top 10 most complaint types is one of the major insights through visualization which list the top 10 number of complaints in a bar plot. Blocked driveway, illegal parking, noise complaints, animal abuse, traffic and homeless encampment are the most frequently submitted concerns. This indicates where city services are under pressure or where residents are most affected on a day-to-day basis. This form of data helps budget and resource planning prioritization.



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

      top_complaints = df['Complaint Type'].value_counts().head(10)

      plt.figure(figsize=(12,6))
      sns.barplot(x=top_complaints.values, y=top_complaints.index)
      plt.title('Top 10 Most Frequent Complaint Types in NYC')
      plt.xlabel('Number of Complaints')
      plt.ylabel('Complaint Type')
      plt.show()
```

Figure 11: Top 10 common complaint types in NYC.



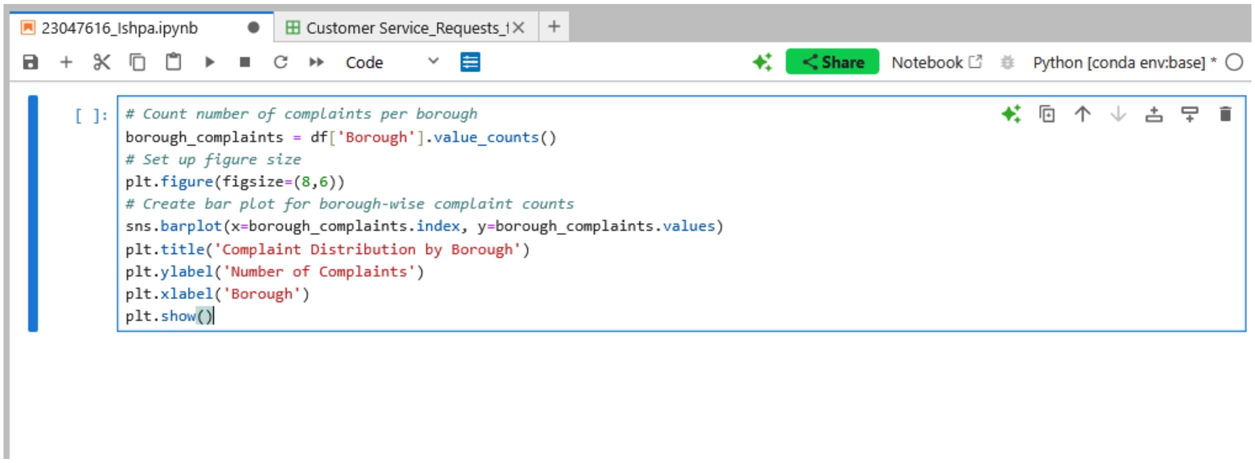
```
plt.title('Top 10 Most Frequent Complaint Types in NYC')
plt.xlabel('Number of Complaints')
plt.ylabel('Complaint Type')
plt.show()
```

Figure 12: Report of top 10 common complaint types in NYC.

Therefore, these are high-priority issues to citizens and city services. Determining the frequency of complaints can help city officials distribute resources to address the most critical and frequent issues of public interest. Hence, this report shows the top 10 frequent complaint categories.

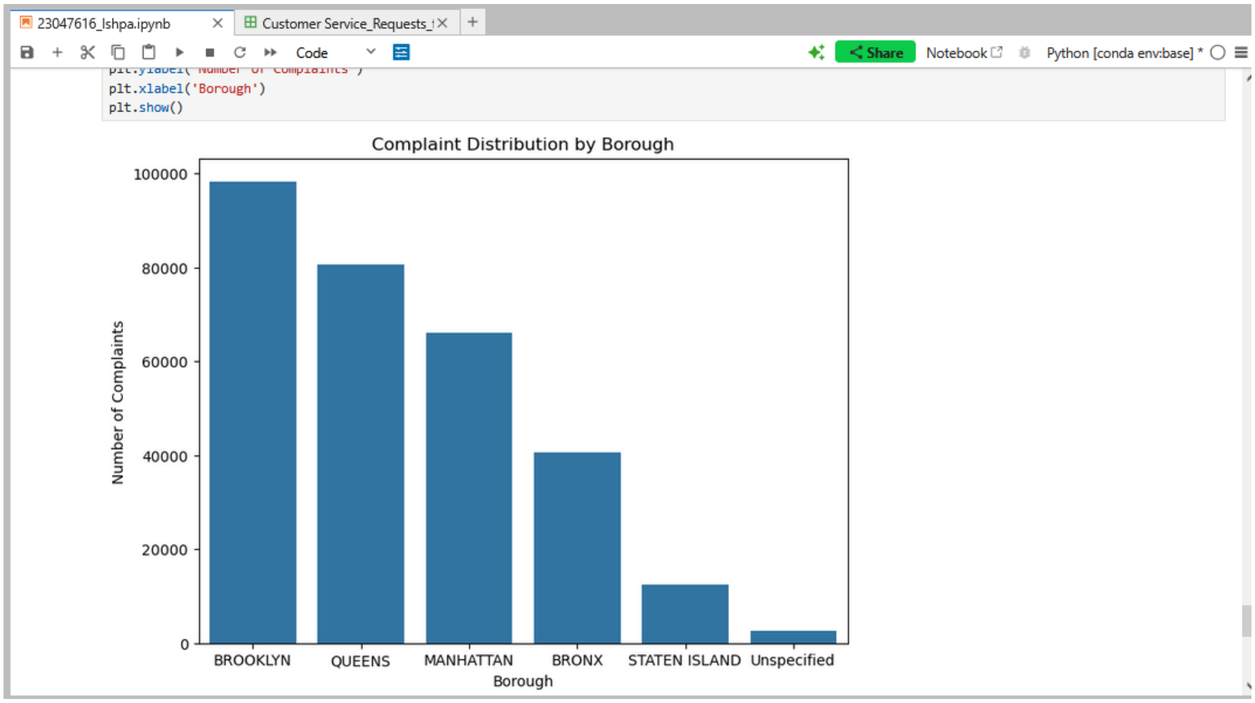
2. Distribution of complaints by borough:

The distribution of complaints by borough is one of the major insights through visualization as the bar chart shows the variation of complaint volumes among various boroughs such as Brooklyn, Queens, Manhattan, Bronx, Staten Island and unspecified. The increased complaints within a borough can be an indication of increased densities of population or greater public unrest with services. This can be helpful to city planners in terms of deploying response teams and studying root causes within areas that are most impacted.



```
# Count number of complaints per borough
borough_complaints = df['Borough'].value_counts()
# Set up figure size
plt.figure(figsize=(8,6))
# Create bar plot for borough-wise complaint counts
sns.barplot(x=borough_complaints.index, y=borough_complaints.values)
plt.title('Complaint Distribution by Borough')
plt.ylabel('Number of Complaints')
plt.xlabel('Borough')
plt.show()
```

Figure 13: Distribution of complaints by borough.



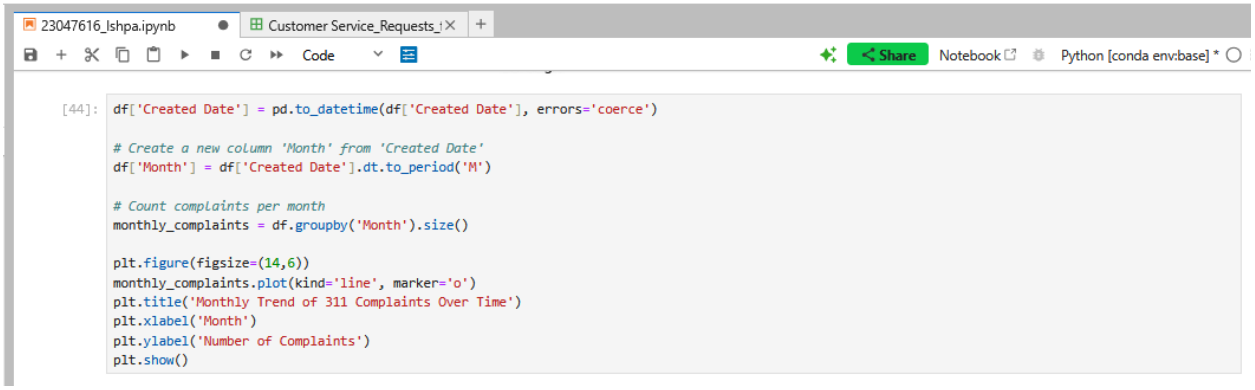
```
plt.ylabel('Number of Complaints')
plt.xlabel('Borough')
plt.show()
```

Figure 14: Report of complaint distribution by borough.

Therefore, this is a chart that shows the bar chart which analyze how the complaints are distributed geographically which reports complaint distribution by borough that has highest complaint rates where Brooklyn, Queens, Manhattan, Bronx, Staten Island, etc, has highest density or huge population who are not satisfied with the services.

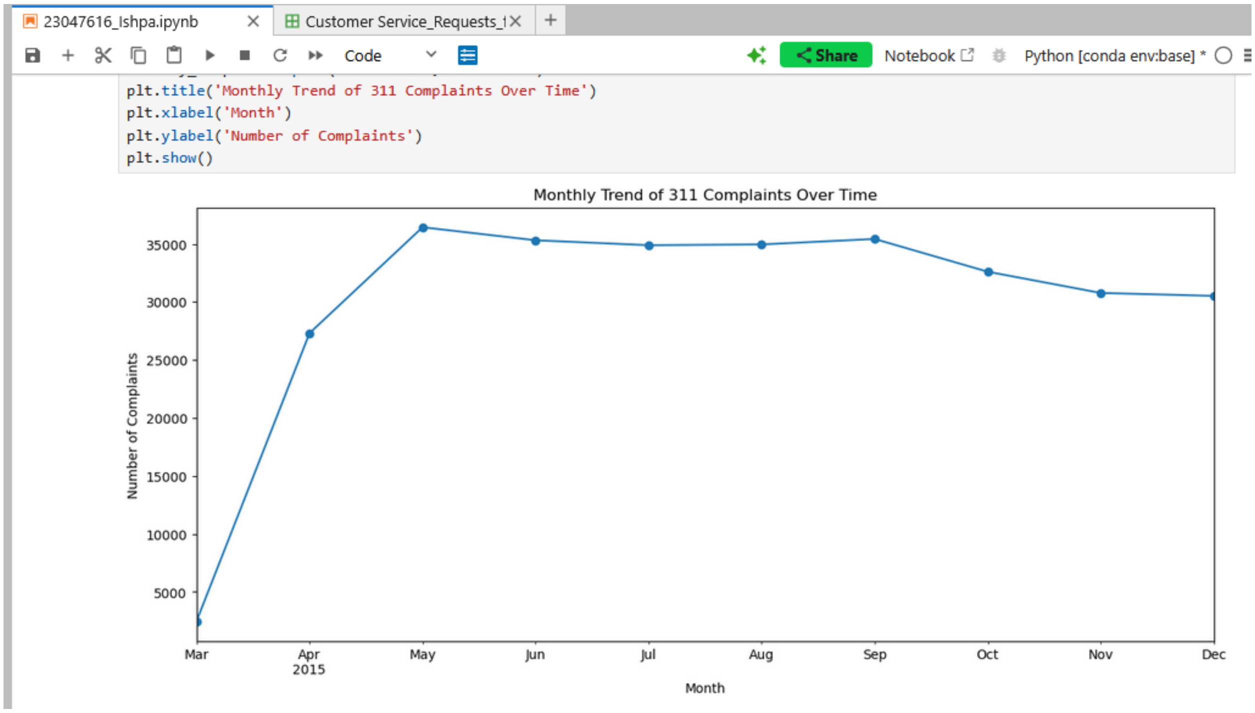
3. Complaint trends over time (Monthly):

The complaints trend over time (monthly) is the other insights via visualization which analyze the seasonal or temporal patterns in complaint volumes that helps the agencies to plan for the seasonal demand or to measure the policy effect. Here, the monthly trend of 311 complaints over time is reported through the line plots that shows the number of complaints according to the month. This time-series plot shows how complaints vary from month to month. As an example, you might notice a rise in heating complaints in winter or noise complaints in summer, implying seasonal patterns. According to the report below the highest complaint is shown in the month of May.



```
[44]: df['Created Date'] = pd.to_datetime(df['Created Date'], errors='coerce')

      # Create a new column 'Month' from 'Created Date'
      df['Month'] = df['Created Date'].dt.to_period('M')

      # Count complaints per month
      monthly_complaints = df.groupby('Month').size()

      plt.figure(figsize=(14,6))
      monthly_complaints.plot(kind='line', marker='o')
      plt.title('Monthly Trend of 311 Complaints Over Time')
      plt.xlabel('Month')
      plt.ylabel('Number of Complaints')
      plt.show()
```

Figure 15: Complaint trends over time.



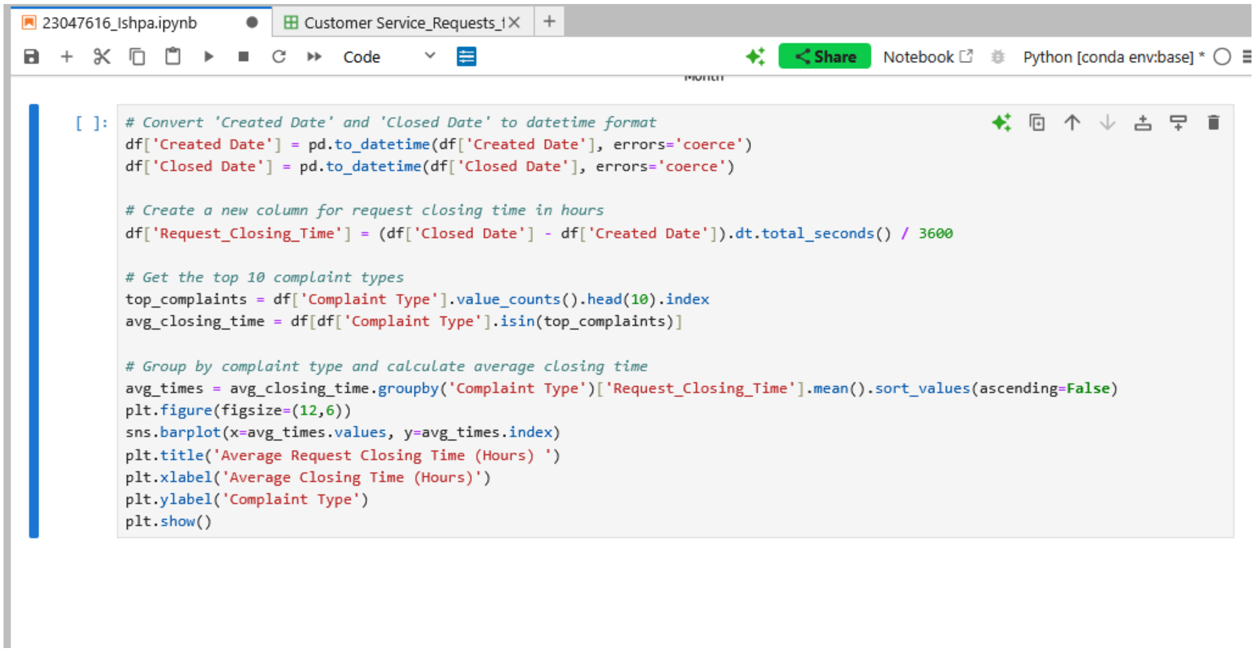
```
plt.title('Monthly Trend of 311 Complaints Over Time')
plt.xlabel('Month')
plt.ylabel('Number of Complaints')
plt.show()
```

Figure 16: Report of complaints trend over time (monthly).

Therefore, this chart shows the seasonal or long-term shifts in complaint volumes with the number of complaints in monthly trend of 311 complaints over time.

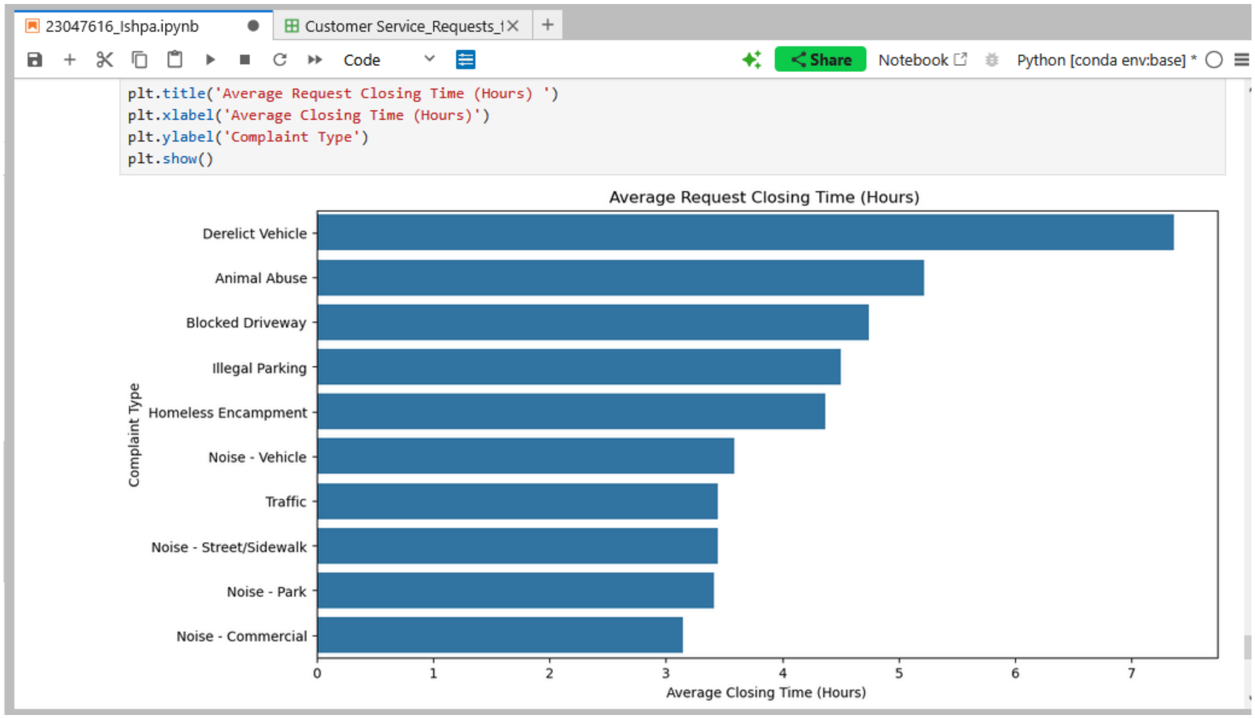
4. Average request closing time by complaint type:

In the average request closing time by complaint type insights, the top 10 most frequent types of complaints in the NYC 311 dataset are examined and calculated the average amount of time it took to close them. These times were displayed in a bar chart. The chart highlights which types of complaints take longer to close. This data helps decide what services might be best optimized through enhanced resource management, faster response channels, or improved complaint handling flows.



```
# Convert 'Created Date' and 'Closed Date' to datetime format
df['Created Date'] = pd.to_datetime(df['Created Date'], errors='coerce')
df['Closed Date'] = pd.to_datetime(df['Closed Date'], errors='coerce')

# Create a new column for request closing time in hours
df['Request_Closing_Time'] = (df['Closed Date'] - df['Created Date']).dt.total_seconds() / 3600

# Get the top 10 complaint types
top_complaints = df['Complaint Type'].value_counts().head(10).index
avg_closing_time = df[df['Complaint Type'].isin(top_complaints)]

# Group by complaint type and calculate average closing time
avg_times = avg_closing_time.groupby('Complaint Type')['Request_Closing_Time'].mean().sort_values(ascending=False)
plt.figure(figsize=(12,6))
sns.barplot(x=avg_times.values, y=avg_times.index)
plt.title('Average Request Closing Time (Hours) ')
plt.xlabel('Average Closing Time (Hours)')
plt.ylabel('Complaint Type')
plt.show()
```

Figure 17: Average request closing time by complaint type.



```
plt.title('Average Request Closing Time (Hours) ')
plt.xlabel('Average Closing Time (Hours)')
plt.ylabel('Complaint Type')
plt.show()
```

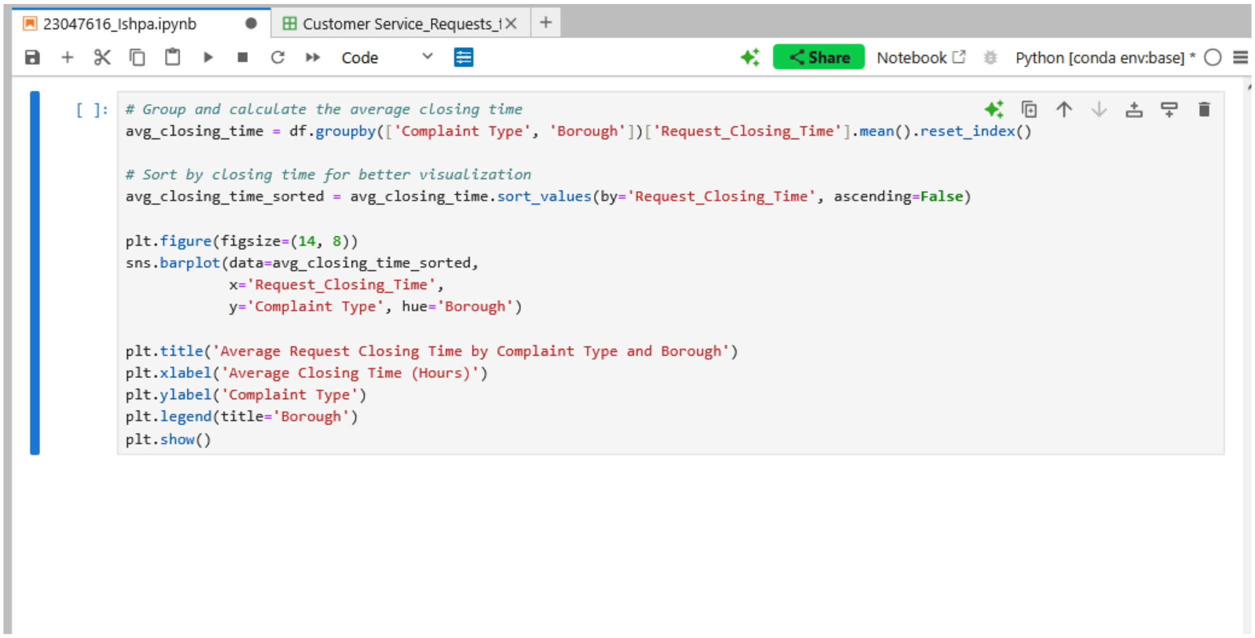
Figure 18: Report of Average request closing time by complaint type.

The x-axis indicates the average closing time in hours where the types of complaints, along the y-axis. It can be seen from the visualization that 'Derelict Vehicle' complaints have the longest resolution time of more than 7 hours. 'Blocked Driveway' and 'Animal Abuse' also have longer closure times, implying these cases can be more time-consuming investigations or lack of resources. Complaints for noise cases take less time, usually within 3 to 4 hours. This would indicate that noise complaints are easier to assess and answer, with quick inspections or dispatches. Overall, this comparison illustrates how much the nature and complexity of a category of complaint have an overwhelming effect on response time and can be employed to guide resource planning and operation effectiveness improvements.

4.2 Arrange the complaint types

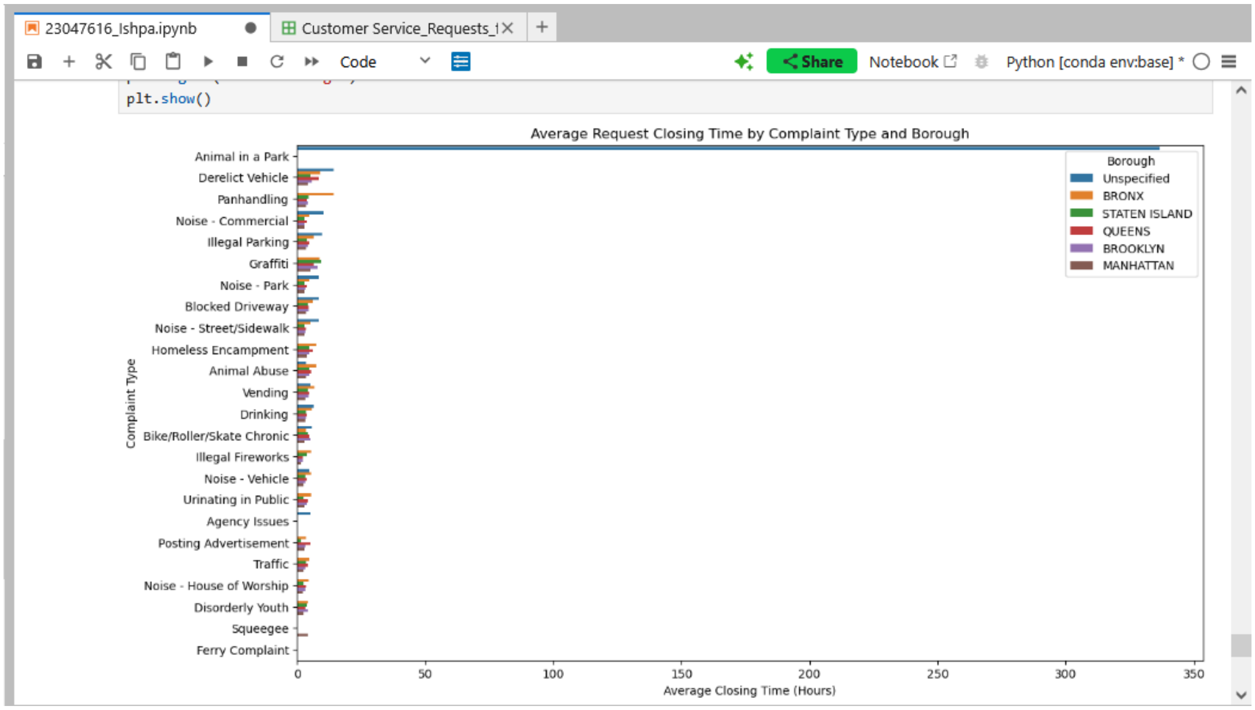
Question: Arrange the complaint types according to their average 'Request_Closing_Time', categorized by various locations. Illustrate it through graph as well.

The question to arrange the complaint types ask how the average time to close service requests (Request_Closing_Time) varies not only by complaint type but also by location, in this case, NYC boroughs. By grouping the data by both Complaint Type and Borough, we calculated the mean resolution time per pair. A bar plot can quickly reveal what types of complaints are slower to close where.



```
# Group and calculate the average closing time
avg_closing_time = df.groupby(['Complaint Type', 'Borough'])['Request_Closing_Time'].mean().reset_index()

# Sort by closing time for better visualization
avg_closing_time_sorted = avg_closing_time.sort_values(by='Request_Closing_Time', ascending=False)

plt.figure(figsize=(14, 8))
sns.barplot(data=avg_closing_time_sorted,
            x='Request_Closing_Time',
            y='Complaint Type', hue='Borough')

plt.title('Average Request Closing Time by Complaint Type and Borough')
plt.xlabel('Average Closing Time (Hours)')
plt.ylabel('Complaint Type')
plt.legend(title='Borough')
plt.show()
```

Figure 19: Arrange the complaint types.



```
plt.show()
```

Figure 20: Report of arranging the complaint types.

From the above bar plot, we see:

Derelict Vehicle complaints close more slowly consistently in the majority of boroughs, particularly Brooklyn and Queens, which may be due to a higher concentration of vehicles or greater volume of derelict vehicles where Animal Abuse complaints close more slowly, on average, in Bronx and Staten Island, which indicates either a slower response or a lower level of resource allocation for animal cases in these boroughs. Complaints such as 'Noise - Commercial' and 'Noise - Street/Sidewalk' are closed more quickly in Manhattan, likely a result of the borough's more concentrated enforcement and quicker dispatch services.

Therefore, this is a valuable finding for planning resources and policy-making, as it can help inform decision-makers to target neighbourhoods and complaint types where the speed of service is slower.

5. Statistical Testing

5.1 Test 1:

Whether the average response time across complaint types is similar or not.

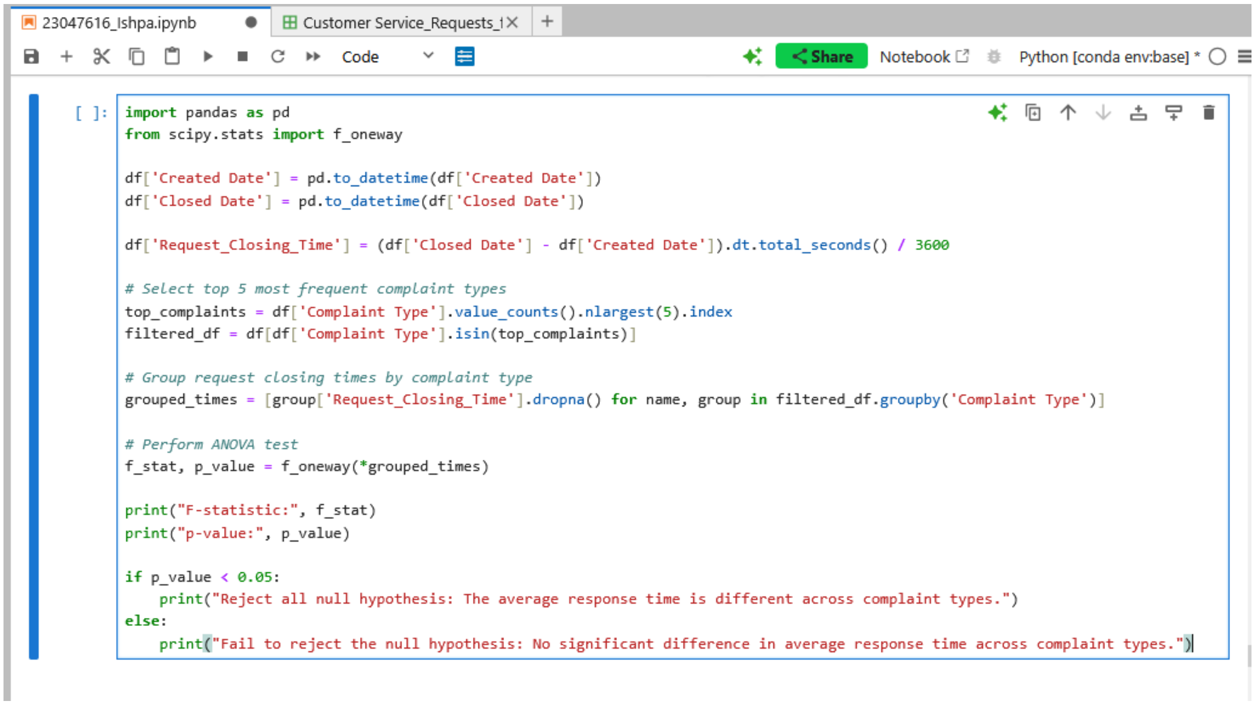
- State the Null Hypothesis (H0) and Alternate Hypothesis (H1).
- Perform the statistical test and provide the p-value.
- Interpret the results to accept or reject the Null Hypothesis.

The main objective of **Test 1** is to ascertain whether distinct complaint categories significantly differ in average response times (Request_Closing_Time). The outcome of this test compares the mean or average through ANOVA method.

Null Hypothesis (H0): The null hypothesis is the assertion that there is no effect in the population. Null hypotheses usually include such words as "no effect," "no difference," or "no relationship." According to our test, the average Request_Closing_Time is the same for all complaint types (Turney, 2022).

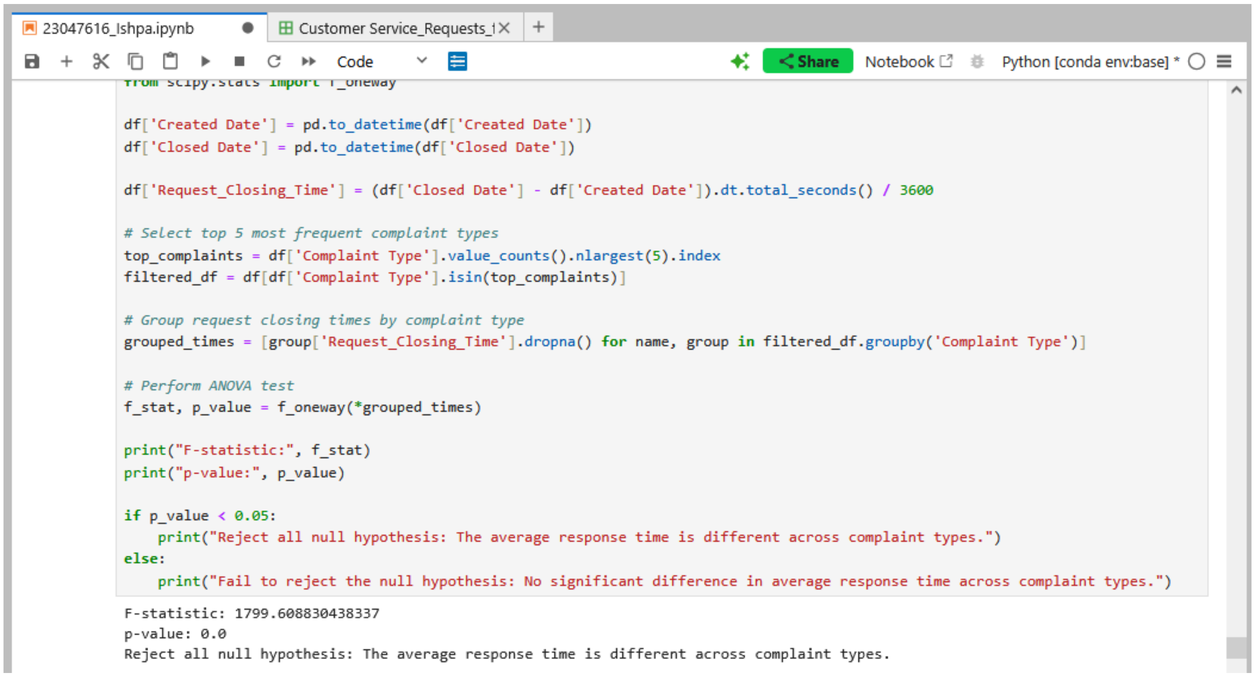
Alternative Hypothesis (H1): The alternative hypothesis (Ha) or (H1) states that there is an effect. The alternative hypothesis is the reverse of the null hypothesis. Alternative hypotheses typically include the words "an effect," "a difference," or "a relationship." According to our test, at least one type of complaint has a different average Request_Closing_Time (Turney, 2022).

Statistical Test Used: One-way **ANOVA (Analysis of Variance)** is a statistical method that is used to compare three or more group means in an effort to determine if there are any statistically significant differences among them. It compares the variability within and between groups to help researchers know if observed differences are due to chance or if they represent real effects. ANOVA is a statistical test used to examine differences in the means of three or more groups. Unlike a t-test, which can only compare two groups, ANOVA can compare more than two groups in a single test, making it an important tool when there are more than two categories in an experiment (Hassan, 2024).



```
import pandas as pd
from scipy.stats import f_oneway

df['Created Date'] = pd.to_datetime(df['Created Date'])
df['Closed Date'] = pd.to_datetime(df['Closed Date'])

df['Request_Closing_Time'] = (df['Closed Date'] - df['Created Date']).dt.total_seconds() / 3600

# Select top 5 most frequent complaint types
top_complaints = df['Complaint Type'].value_counts().nlargest(5).index
filtered_df = df[df['Complaint Type'].isin(top_complaints)]

# Group request closing times by complaint type
grouped_times = [group['Request_Closing_Time'].dropna() for name, group in filtered_df.groupby('Complaint Type')]

# Perform ANOVA test
f_stat, p_value = f_oneway(*grouped_times)

print("F-statistic:", f_stat)
print("p-value:", p_value)

if p_value < 0.05:
    print("Reject all null hypothesis: The average response time is different across complaint types.")
else:
    print("Fail to reject the null hypothesis: No significant difference in average response time across complaint types.")
```

Figure 21: Test 1



```
from scipy.stats import f_oneway

df['Created Date'] = pd.to_datetime(df['Created Date'])
df['Closed Date'] = pd.to_datetime(df['Closed Date'])

df['Request_Closing_Time'] = (df['Closed Date'] - df['Created Date']).dt.total_seconds() / 3600

# Select top 5 most frequent complaint types
top_complaints = df['Complaint Type'].value_counts().nlargest(5).index
filtered_df = df[df['Complaint Type'].isin(top_complaints)]

# Group request closing times by complaint type
grouped_times = [group['Request_Closing_Time'].dropna() for name, group in filtered_df.groupby('Complaint Type')]

# Perform ANOVA test
f_stat, p_value = f_oneway(*grouped_times)

print("F-statistic:", f_stat)
print("p-value:", p_value)

if p_value < 0.05:
    print("Reject all null hypothesis: The average response time is different across complaint types.")
else:
    print("Fail to reject the null hypothesis: No significant difference in average response time across complaint types.")
```

```
F-statistic: 1799.608830438337
p-value: 0.0
Reject all null hypothesis: The average response time is different across complaint types.
```

Figure 22: Test 1 Output

Interpretation:

The F-statistics is 1799.6088 whereas p-value is 0.0.

If the p-value < 0.05, then this shows reject all null hypothesis. The average response time is different across complaint types.

If the p-value is > 0.05, then this shows fail to reject the null hypothesis. No significant difference in average response time across complaint types.

5.2 Test 2:

Whether the type of complaint or service requested and location are related.

- State the Null Hypothesis (H0) and Alternate Hypothesis (H1).
- Perform the statistical test and provide the p-value.
- Interpret the results to accept or reject the Null Hypothesis.

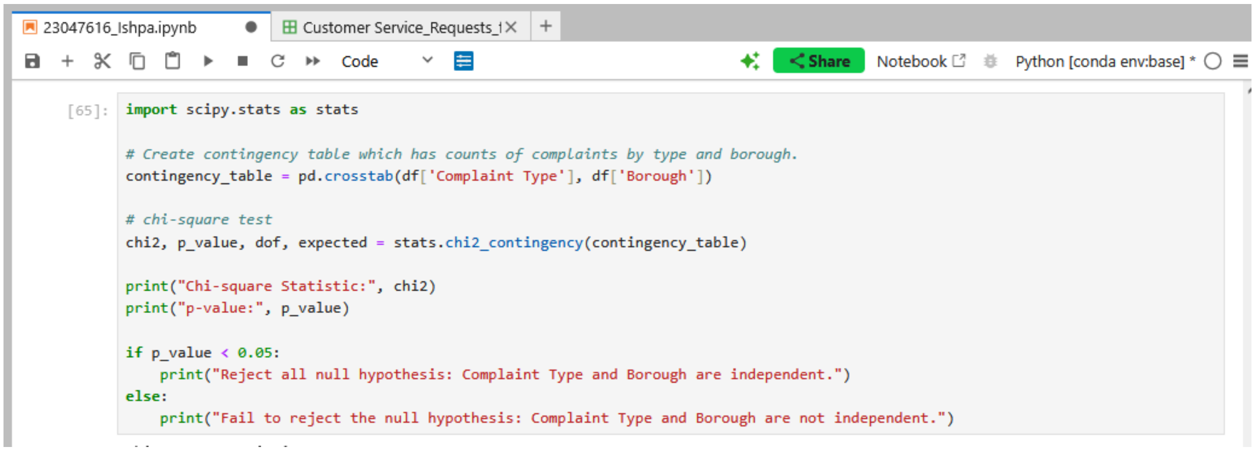
The objective of **Test 2** is to determine whether the type of complaint or service requested is associated with the borough in which it occurs. The outcome of this test compares the frequency through Chi-square method.

Null Hypothesis (H0): The null hypothesis is the assertion that there is no effect in the population. Null hypotheses usually include such words as "no effect," "no difference," or "no relationship." According to our test, the average Request_Closing_Time is the same for all complaint types (Turney, 2022).

Alternative Hypothesis (H1): The alternative hypothesis (Ha) or (H1) states that there is an effect. The alternative hypothesis is the reverse of the null hypothesis. Alternative hypotheses typically include the words "an effect," "a difference," or "a relationship." According to our test, at least one type of complaint has a different average Request_Closing_Time (Turney, 2022)

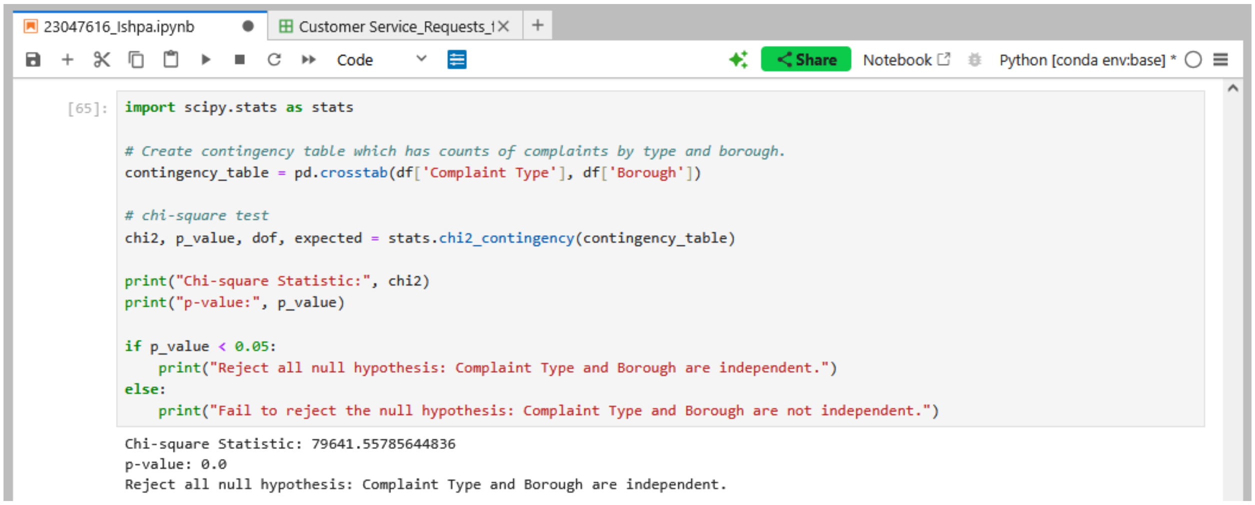
Statistical test Used: Chi-square test is a statistical test for categorical data. It is one method of inquiring whether or not your data significantly differ from what you would expect (Turney, 2022). There are two types of chi-square tests:

- The chi-square goodness of fit test checks whether the frequency distribution for a categorical variable varies from expectation (Turney, 2022).
- The chi-square test of independence is used to see whether two categorical variables are related to each other (Turney, 2022).



```
import scipy.stats as stats

# Create contingency table which has counts of complaints by type and borough.
contingency_table = pd.crosstab(df['Complaint Type'], df['Borough'])

# chi-square test
chi2, p_value, dof, expected = stats.chi2_contingency(contingency_table)

print("Chi-square Statistic:", chi2)
print("p-value:", p_value)

if p_value < 0.05:
    print("Reject all null hypothesis: Complaint Type and Borough are independent.")
else:
    print("Fail to reject the null hypothesis: Complaint Type and Borough are not independent.")
```

Figure 23: Test 2



```
import scipy.stats as stats

# Create contingency table which has counts of complaints by type and borough.
contingency_table = pd.crosstab(df['Complaint Type'], df['Borough'])

# chi-square test
chi2, p_value, dof, expected = stats.chi2_contingency(contingency_table)

print("Chi-square Statistic:", chi2)
print("p-value:", p_value)

if p_value < 0.05:
    print("Reject all null hypothesis: Complaint Type and Borough are independent.")
else:
    print("Fail to reject the null hypothesis: Complaint Type and Borough are not independent.")
```

```
Chi-square Statistic: 79641.55785644836
p-value: 0.0
Reject all null hypothesis: Complaint Type and Borough are independent.
```

Figure 24: Test 2 Output

Interpretation:

The Chi-Square Statistics is 79641.5578 whereas p-value is 0.0.

If the p-value < 0.05, then this shows reject all null hypothesis. Complaint Type and Borough are independent.

If the p-value is > 0.05, then this shows fail to reject the null hypothesis. Complaint Type and Borough are not independent.

6. Conclusion

In the conclusion, we analyzed the NYC 311 Customer service request dataset where in the section of data preparation the data were transformed into usable form by converting various data field into datetime format, computing the request closing time in hours, and removing unnecessary columns and null values.

While analyzing the data we calculated summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame and the correlation of all variables. The most frequently occurring complaint categories were determined, and it was shown that particular categories such as noise or water system complaints appear significantly more dominantly under data exploration. As we find out the major insights through visualization the visual inspection of the patterns of the monthly instances illustrated seasonality in the counts of complaints, identifying the times of increased service demand. By grouping data by complaint types, we discovered that the closing time for averages per request can be very disparate; some problems close rapidly, while others, like complaints involving involved complex infrastructure or safety issues, require long times to close. Additional research based on borough showed geographical differences in response efficiency for service.

The statistical testing was performed using ANOVA and Chi-Square test. Hence this coursework helped to gain knowledge in the various hypothesis such as null hypothesis and alternate hypothesis after calculating the p-value. These findings help actionable insight to policymakers and service agencies. This coursework not only establishes technical competence in data cleaning, statistical testing and visualization but also offers an efficient foundation for data-informed decision-making within urban services.

7. References

ajaymehta, 2023. *Filtering Out Irrelevant Features.* [Online]
Available at: https://medium.com/@dancerworld60/filtering-out-irrelevant-features-a-comprehensive-survey-of-filter-based-techniques-for-feature-9275d86996ab
[Accessed 29 April 2025].

AWS, 2025. *What is data preparation?.* [Online]
Available at: https://aws.amazon.com/what-is/data-preparation/
[Accessed 11 April 2025].

Collins, E. R., 2024. *5 Best ways to convert integer date to datetime in python.* [Online]
Available at: https://blog.finxter.com/5-best-ways-to-convert-integer-date-to-datetime-in-python/
[Accessed 29 April 2025].

coursera, 2024. *What is data exploration?.* [Online]
Available at: https://www.coursera.org/articles/data-exploration
[Accessed 2 May 2025].

Coursera, 2025. *Data Analytics.* [Online]
Available at: https://www.coursera.org/articles/data-analytics?msockid=2718693a355868b23e2b7b8b34596929
[Accessed 11 April 2025].

datagy, 2023. *Pandas unique().* [Online]
Available at: https://datagy.io/pandas-unique/
[Accessed 30 April 2025].

Dey, R., 2023. *Getting started with Anaconda: For Absolute Beginners.* [Online]
Available at: https://medium.com/@roshmitadey/getting-started-with-anaconda-for-absolute-beginners-14acfc44fab9
[Accessed 13 May 2025].

Dey, R., 2023. *Getting started with Anaconda: For Absolute Beginners.* [Online]
Available at: https://medium.com/@roshmitadey/getting-started-with-anaconda-for-

absolute-beginners-14acfc44fab9
[Accessed 13 May 2025].

geeksforgeeks, 2024. *Correlation.* [Online]
Available at: https://www.geeksforgeeks.org/correlation-meaning-significance-types-and-degree-of-correlation/
[Accessed 30 April 2025].

geeksforgeeks, 2025. *Drop row from pandas dataframe with missing values or NaN in columns.* [Online]
Available at: https://www.geeksforgeeks.org/drop-rows-from-pandas-dataframe-with-missing-values-or-nan-in-columns/
[Accessed 29 April 2025].

Hassan, M., 2024. *ANOVA (Analysis of variance).* [Online]
Available at: https://researchmethod.net/anova/
[Accessed 3 May 2025].

J.Hand, D., 2013. *Statistics: A very Short Introduction.* s.l.:Oxford University Press.

jupyter.org, 2015. *What is Jupyter?.* [Online]
Available at: https://docs.jupyter.org/en/stable/what_is_jupyter.html
[Accessed 13 May 2025].

JupyterLab, 2023. [Online]
Available at: https://nightingalehq.ai/knowledgebase/glossary/what-is-jupyter/jupyter.jpg
[Accessed 13 May 2025].

Roberts, S., 2025. *What is Microsoft Word?.* [Online]
Available at: https://www.theknowledgeacademy.com/blog/what-is-microsoft-word/
[Accessed 13 May 2025].

techmodena.com, 2022. *Ms-Word.* [Online]
Available at: https://techmodena.com/wp-content/uploads/2021/09/ms-word.png
[Accessed 13 May 2025].

Turney, S., 2022. *Chi-Square tests.* [Online]
Available at: https://www.scribbr.com/statistics/chi-square-tests/
[Accessed 3 May 2025].

Turney, S., 2022. *Kurtosis.* [Online]
Available at: https://www.scribbr.com/statistics/skewness/
[Accessed 30 April 2025].

Turney, S., 2022. *Null and Alternative Hypothesis.* [Online]
Available at: https://www.scribbr.com/statistics/null-and-alternative-hypotheses/
[Accessed 3 May 2025].

Yennhi95zz, 2023. *Data Understanding.* [Online]
Available at: https://medium.com/@yennhi95zz/2-data-understanding-a-key-element-of-the-crisp-dm-methodology-for-data-mining-1bbd7f580cda
[Accessed 11 April 2025].