Analysis of Public Toilet Problems in NYC

Prashanth & Jeet

2025-04-28

Contents

1 Introduction

This project analyzes public restroom availability and sanitation-related complaints in New York City.

2 Data Preparation

Load Required Libraries

```
library(tidyverse)
library(data.table)
library(lubridate)
library(ggplot2)
library(caret)
library(rpart)
library(rpart.plot)
library(ggmap)
library(sf)
library(ggpubr)
library(multcomp)
library(arrow)
```

3 Load Datasets

```
# Load public restrooms dataset
public_restrooms <- read_csv("Public_Restrooms_20250427.csv")

# Load 311 complaints dataset
complaints <- read_csv("311_Service_Requests_from_2021_to_Present_20250427.csv")
```

4 Exploratory Data Analysis (EDA)

4.1 Explore Dataset Columns

```
# View column names of Public Restrooms
colnames(public_restrooms)
```

```
##  [1] "Facility Name"      "Location Type"      "Operator"
##  [4] "Status"             "Open"               "Hours of Operation"
##  [7] "Accessibility"      "Restroom Type"      "Changing Stations"
## [10] "Additional Notes"   "Website"            "Latitude"
## [13] "Longitude"          "Location"
```

```
# View column names of 311 Complaints
colnames(complaints)
```

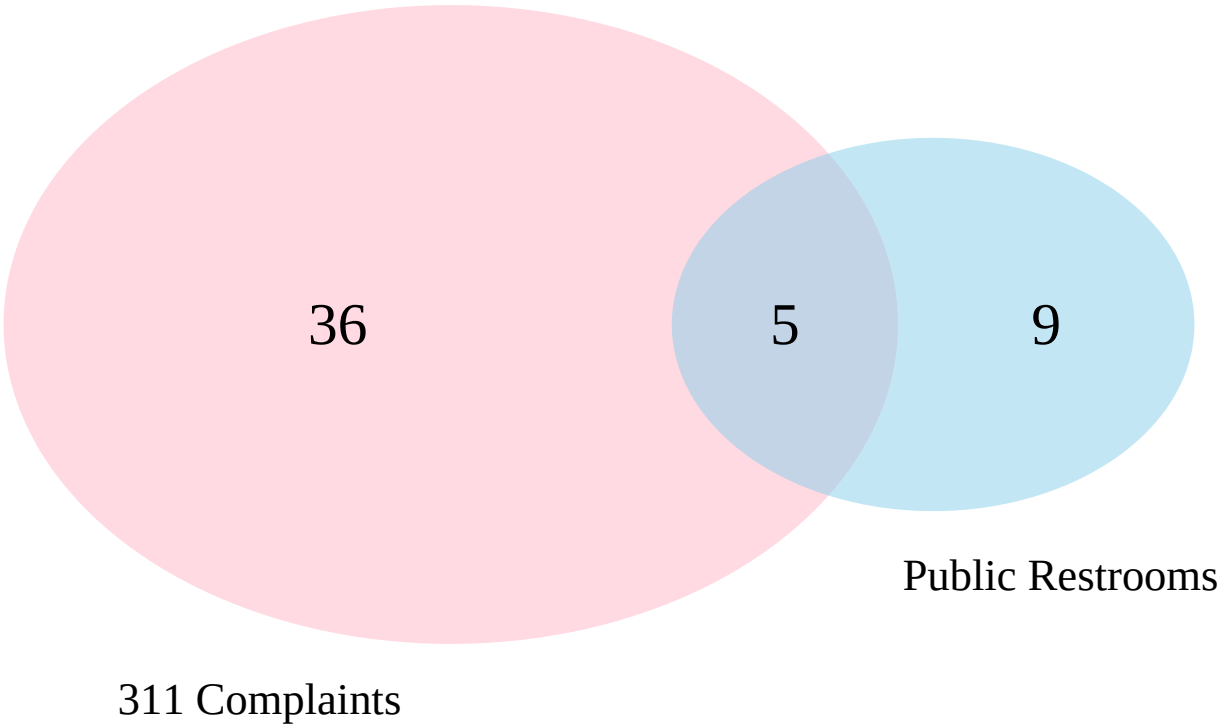
```
##  [1] "Unique Key"                     "Created Date"
##  [3] "Closed Date"                    "Agency"
##  [5] "Agency Name"                    "Complaint Type"
##  [7] "Descriptor"                     "Location Type"
##  [9] "Incident Zip"                   "Incident Address"
## [11] "Street Name"                    "Cross Street 1"
## [13] "Cross Street 2"                 "Intersection Street 1"
## [15] "Intersection Street 2"          "Address Type"
## [17] "City"                           "Landmark"
## [19] "Facility Type"                  "Status"
## [21] "Due Date"                       "Resolution Description"
## [23] "Resolution Action Updated Date" "Community Board"
## [25] "BBL"                            "Borough"
## [27] "X Coordinate (State Plane)"     "Y Coordinate (State Plane)"
## [29] "Open Data Channel Type"         "Park Facility Name"
## [31] "Park Borough"                   "Vehicle Type"
## [33] "Taxi Company Borough"           "Taxi Pick Up Location"
## [35] "Bridge Highway Name"            "Bridge Highway Direction"
## [37] "Road Ramp"                      "Bridge Highway Segment"
## [39] "Latitude"                       "Longitude"
## [41] "Location"
```

5 Identify and Visualize Common Columns between Datasets

```
library(VennDiagram)

# Get the column names
restroom_cols <- colnames(public_restrooms)
complaint_cols <- colnames(complaints)

# Find common columns
common_columns <- intersect(restroom_cols, complaint_cols)
common_columns
```

```
## [1] "Location Type" "Status"        "Latitude"      "Longitude"
## [5] "Location"
```

```r
# Create Venn Diagram
grid.newpage()  # <- This is necessary
invisible(
draw.pairwise.venn(
  area1 = length(restroom_cols),
  area2 = length(complaint_cols),
  cross.area = length(common_columns),
  category = c("Public Restrooms", "311 Complaints"),
  fill = c("skyblue", "pink1"),
  lty = "blank",
  cex = 2,
  cat.cex = 1.5,
  cat.pos = c(-20, 20),
  cat.dist = 0.05
)
)
```



5.0.1 As part of EDA, we identified 5 common columns between the Public Restrooms and 311 Complaints datasets, namely Location Type, Status, Latitude, Longitude, and Location, visualized using a Venn Diagram.

6 Detailed EDA on Complaints Dataset

```
# View first 10 rows of complaints data
head(complaints, 10)
```

```
## # A tibble: 10 x 41
##    `Unique Key` `Created Date`         `Closed Date`        Agency `Agency Name`
##           <dbl> <chr>                  <chr>                <chr>  <chr>
##  1     64217483 02/28/2025 11:52:40 PM 03/03/2025 01:52:05~ DSNY   Department o~
##  2     64217484 02/28/2025 11:38:58 PM 03/01/2025 01:52:04~ DSNY   Department o~
##  3     64219743 02/28/2025 11:35:04 PM 03/05/2025 03:07:56~ DSNY   Department o~
##  4     64219776 02/28/2025 11:32:18 PM 03/03/2025 01:26:46~ DSNY   Department o~
##  5     64226221 02/28/2025 11:30:00 PM 03/01/2025 12:46:00~ DSNY   Department o~
##  6     64216276 02/28/2025 11:20:32 PM 03/01/2025 10:04:27~ DSNY   Department o~
##  7     64217463 02/28/2025 11:17:37 PM 03/03/2025 02:28:53~ DSNY   Department o~
##  8     64217487 02/28/2025 11:12:34 PM 03/03/2025 01:29:05~ DSNY   Department o~
##  9     64218561 02/28/2025 11:07:50 PM 03/17/2025 02:38:33~ DSNY   Department o~
## 10     64217985 02/28/2025 11:05:04 PM 02/28/2025 11:26:01~ DSNY   Department o~
## # i 36 more variables: `Complaint Type` <chr>, Descriptor <chr>,
## #   `Location Type` <chr>, `Incident Zip` <dbl>, `Incident Address` <chr>,
## #   `Street Name` <chr>, `Cross Street 1` <chr>, `Cross Street 2` <chr>,
## #   `Intersection Street 1` <chr>, `Intersection Street 2` <chr>,
## #   `Address Type` <chr>, City <chr>, Landmark <chr>, `Facility Type` <chr>,
## #   Status <chr>, `Due Date` <chr>, `Resolution Description` <chr>,
## #   `Resolution Action Updated Date` <chr>, `Community Board` <chr>, ...
```

```
# Glimpse structure of the dataset
glimpse(complaints)
```

```
## Rows: 1,510,141
## Columns: 41
## $ `Unique Key`                  <dbl> 64217483, 64217484, 64219743, 6421977~
## $ `Created Date`                <chr> "02/28/2025 11:52:40 PM", "02/28/2025~
## $ `Closed Date`                 <chr> "03/03/2025 01:52:05 PM", "03/01/2025~
## $ Agency                        <chr> "DSNY", "DSNY", "DSNY", "DSNY", "DSNY~
## $ `Agency Name`                 <chr> "Department of Sanitation", "Departme~
## $ `Complaint Type`              <chr> "Illegal Dumping", "Illegal Dumping",~
## $ Descriptor                    <chr> "Removal Request", "Removal Request",~
## $ `Location Type`               <chr> "Sidewalk", "Sidewalk", "Sidewalk", "~
## $ `Incident Zip`                <dbl> 11103, 11433, 11433, 10001, 11433, 11~
## $ `Incident Address`            <chr> "36-11 30 AVENUE", "162-15 ARCHER AVE~
## $ `Street Name`                 <chr> "30 AVENUE", "ARCHER AVENUE", "168 PL~
## $ `Cross Street 1`              <chr> "36 STREET", "UNION HALL STREET", "92~
## $ `Cross Street 2`              <chr> "37 STREET", "GUY R BREWER BOULEVARD"~
## $ `Intersection Street 1`       <chr> "36 STREET", "UNION HALL STREET", "92~
## $ `Intersection Street 2`       <chr> "37 STREET", "GUY R BREWER BOULEVARD"~
## $ `Address Type`                <chr> "ADDRESS", "ADDRESS", "ADDRESS", "ADD~
```

```
## $ City                              <chr> "ASTORIA", "JAMAICA", "JAMAICA", "NEW~
## $ Landmark                          <chr> "30 AVENUE", "ARCHER AVENUE", "168 PL~
## $ `Facility Type`                   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Status                            <chr> "Closed", "Closed", "Closed", "Closed~
## $ `Due Date`                        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ `Resolution Description`          <chr> "The Department of Sanitation investi~
## $ `Resolution Action Updated Date` <chr> "03/01/2025 01:25:10 PM", "03/01/2025~
## $ `Community Board`                 <chr> "01 QUEENS", "12 QUEENS", "12 QUEENS"~
## $ BBL                               <chr> "4006300044", "4101020025", "41021102~
## $ Borough                           <chr> "QUEENS", "QUEENS", "QUEENS", "MANHAT~
## $ `X Coordinate (State Plane)`      <dbl> 1007185, 1040572, 1042226, 988237, 10~
## $ `Y Coordinate (State Plane)`      <dbl> 217977, 195503, 196391, 211435, 19697~
## $ `Open Data Channel Type`          <chr> "PHONE", "ONLINE", "ONLINE", "ONLINE"~
## $ `Park Facility Name`              <chr> "Unspecified", "Unspecified", "Unspec~
## $ `Park Borough`                    <chr> "QUEENS", "QUEENS", "QUEENS", "MANHAT~
## $ `Vehicle Type`                    <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ `Taxi Company Borough`            <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ `Taxi Pick Up Location`           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ `Bridge Highway Name`             <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ `Bridge Highway Direction`        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ `Road Ramp`                       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ `Bridge Highway Segment`          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Latitude                          <dbl> 40.76494, 40.70311, 40.70553, 40.7470~
## $ Longitude                         <dbl> -73.91721, -73.79687, -73.79089, -73.~
## $ Location                          <chr> "(40.76494318338899, -73.917205357506~
```

```
# Check if Created Date has missing values
sum(is.na(complaints$`Created Date`))
```

```
## [1] 0
```

```
# Check unique complaint types
unique(complaints$`Complaint Type`)
```

```
##  [1] "Illegal Dumping"
##  [2] "Litter Basket Request"
##  [3] "Derelict Vehicles"
##  [4] "Dirty Condition"
##  [5] "Vendor Enforcement"
##  [6] "Dead Animal"
##  [7] "Residential Disposal Complaint"
##  [8] "Obstruction"
##  [9] "Highway Condition"
## [10] "Dumpster Complaint"
## [11] "Missed Collection"
## [12] "Graffiti"
## [13] "Illegal Posting"
## [14] "Commercial Disposal Complaint"
## [15] "Abandoned Bike"
## [16] "Sanitation Worker or Vehicle Complaint"
## [17] "Lot Condition"
## [18] "Litter Basket Complaint"
## [19] "Street Sweeping Complaint"
```

```
## [20] "Institution Disposal Complaint"
## [21] "Recycling Basket Complaint"
## [22] "Snow or Ice"
## [23] "Incorrect Data"
## [24] "Oil or Gas Spill"
## [25] "Unspecified"
## [26] "Transfer Station Complaint"
## [27] "Adopt-A-Basket"
## [28] "Retailer Complaint"
## [29] "Seasonal Collection"
## [30] "DSNY Internal"
## [31] "Electronics Waste Appointment"
## [32] "Request Large Bulky Item Collection"
## [33] "Sanitation Condition"
## [34] "Dirty Conditions"
## [35] "Overflowing Litter Baskets"
## [36] "Other Enforcement"
## [37] "Litter Basket / Request"
## [38] "Missed Collection (All Materials)"
## [39] "Storm"
## [40] "Vacant Lot"
## [41] "Derelict Bicycle"
## [42] "Recycling Enforcement"
## [43] "Sweeping/Missed"
## [44] "Sweeping/Inadequate"
## [45] "Overflowing Recycling Baskets"
## [46] "Collection Truck Noise"
## [47] "Change Collection Schedule"
## [48] "Foam Ban Enforcement"
## [49] "Sweeping/Missed-Inadequate"
## [50] "Snow"
## [51] "Snow Removal"
```

```
# Check missingness in Latitude and Longitude
sum(is.na(complaints$Latitude))
```

```
## [1] 34249
```

```
sum(is.na(complaints$Longitude))
```

```
## [1] 34249
```

7 Detailed EDA on Public Restrooms Dataset

```
# View first 10 rows
head(public_restrooms, 10)
```

```
## # A tibble: 10 x 14
##    `Facility Name`    `Location Type` Operator Status Open  `Hours of Operation`
```

```
##    <chr>              <chr>           <chr>    <chr>  <chr> <chr>
##  1 East River Park Z~ Park            NYC Par~ Not O~ <NA>  "8am-4pm, Open late~
##  2 Passerelle Buildi~ Park            NYC Par~ Opera~ Year~ "8am-4pm, Open late~
##  3 The High Line Zon~ Park            NYC Par~ Opera~ Year~  <NA>
##  4 Corporal Thompson~ Park            NYC Par~ Opera~ Year~ "8am-4pm, Open late~
##  5 Bushwick Library,~ Library         BPL      Opera~ Year~ "Monday\t10 am - 6 ~
##  6 Rienzi Playground  Park            NYC Par~ Opera~ Year~ "8am-4pm, Open late~
##  7 55 East 52nd Stre~ Privately Owne~ Park Av~ Opera~ Year~ "Everyday 8:00 am-1~
##  8 Ottendorfer Libra~ Library         NYPL     Opera~ Year~ "Sunday: Closed \nM~
##  9 South Beach Zone 2 Park            NYC Par~ Opera~ Seas~ "8am-4pm, Open late~
## 10 Grand Central (Me~ Transit         LIRR-MNR Opera~ Year~ "Bathrooms are open~
## # i 8 more variables: Accessibility <chr>, `Restroom Type` <chr>,
## #   `Changing Stations` <chr>, `Additional Notes` <chr>, Website <chr>,
## #   Latitude <dbl>, Longitude <dbl>, Location <chr>
```

```
# Glimpse structure
glimpse(public_restrooms)
```

```
## Rows: 1,047
## Columns: 14
## $ `Facility Name`      <chr> "East River Park Zone 3", "Passerelle Building", ~
## $ `Location Type`      <chr> "Park", "Park", "Park", "Park", "Library", "Park"~
## $ Operator             <chr> "NYC Parks", "NYC Parks", "NYC Parks", "NYC Parks~
## $ Status               <chr> "Not Operational", "Operational", "Operational", ~
## $ Open                 <chr> NA, "Year Round", "Year Round", "Year Round", "Ye~
## $ `Hours of Operation` <chr> "8am-4pm, Open later seasonally", "8am-4pm, Open ~
## $ Accessibility        <chr> NA, NA, NA, "Fully Accessible", "Fully Accessible~
## $ `Restroom Type`      <chr> NA, NA, NA, "Multi-Stall W/M Restrooms", "Single-~
## $ `Changing Stations`  <chr> NA, NA, NA, NA, "Yes", NA, "No", "Yes", NA, NA, N~
## $ `Additional Notes`   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "MTA station ~
## $ Website              <chr> NA, NA, NA, NA, "https://www.bklynlibrary.org/loc~
## $ Latitude             <dbl> 40.71590, 40.75166, 40.74364, 40.63852, 40.70457,~
## $ Longitude            <dbl> -73.97519, -73.84325, -74.00687, -74.11793, -73.9~
## $ Location             <chr> "POINT (-73.975189 40.715899)", "POINT (-73.84325~
```

```
# Check missing values
sum(is.na(public_restrooms$Latitude))
```

```
## [1] 0
```

```
sum(is.na(public_restrooms$Longitude))
```

```
## [1] 0
```

```
sum(is.na(public_restrooms$Status))
```

```
## [1] 0
```

```
# Check unique values for Status
unique(public_restrooms$Status)
```

```
## [1] "Not Operational"         "Operational"
## [3] "Closed for Construction" "Closed"
```

```
# Check unique values for Accessibility
unique(public_restrooms$Accessibility)
```

```
## [1] NA                     "Fully Accessible"     "Partially Accessible"
## [4] "Not Accessible"
```

8 Cleaning Procedure:

9 Step 1: Cleaning 311 Complaints Dataset

```
# Convert to tibble (recommended)
complaints <- tibble::as_tibble(complaints)
```

```
library(dplyr)
library(lubridate)

# Cleaning 311 Complaints
complaints_filtered <- complaints %>%
  filter(`Complaint Type` %in% c("Dirty Condition", "Dirty Conditions", "Unsanitary Conditions")) %>%
  mutate(
    # Convert Created Date to Date-Time format
    Created_Date = mdy_hms(`Created Date`, tz = "America/New_York"),

    # Rename Complaint Type properly inside mutate itself
    Complaint_Type = `Complaint Type`,

    # Extract Hour
    Hour = hour(Created_Date),

    # Create Time Group based on Hour
    Time_Group = case_when(
      Hour >= 0  & Hour < 6  ~ "Late Night (00:00-05:59)",
      Hour >= 6  & Hour < 12 ~ "Morning (06:00-11:59)",
      Hour >= 12 & Hour < 18 ~ "Afternoon (12:00-17:59)",
      Hour >= 18 & Hour <= 23 ~ "Evening (18:00-23:59)"
    ),

    # Round Latitude and Longitude to 4 decimal places
    Latitude = round(Latitude, 4),
    Longitude = round(Longitude, 4)
  ) %>%
  # Now select without renaming
  dplyr::select(
    Created_Date,
    Complaint_Type,
    Hour,
```

```
    Time_Group,
    Latitude,
    Longitude,
    Borough
  )

# View cleaned data
head(complaints_filtered, 10)
```

```
## # A tibble: 10 x 7
##    Created_Date        Complaint_Type   Hour Time_Group       Latitude Longitude
##    <dttm>              <chr>           <int> <chr>               <dbl>     <dbl>
##  1 2025-02-28 23:20:32 Dirty Condition    23 Evening (18:00-~     40.7     -74.0
##  2 2025-02-28 22:40:20 Dirty Condition    22 Evening (18:00-~     40.7     -73.8
##  3 2025-02-28 22:37:08 Dirty Condition    22 Evening (18:00-~     40.5     -74.2
##  4 2025-02-28 21:31:09 Dirty Condition    21 Evening (18:00-~     40.7     -73.9
##  5 2025-02-28 20:36:47 Dirty Condition    20 Evening (18:00-~     40.8     -73.8
##  6 2025-02-28 20:32:40 Dirty Condition    20 Evening (18:00-~     40.8     -73.8
##  7 2025-02-28 20:29:01 Dirty Condition    20 Evening (18:00-~     40.8     -73.8
##  8 2025-02-28 20:24:28 Dirty Condition    20 Evening (18:00-~     40.8     -73.8
##  9 2025-02-28 20:20:34 Dirty Condition    20 Evening (18:00-~     40.8     -73.8
## 10 2025-02-28 20:14:33 Dirty Condition    20 Evening (18:00-~     40.7     -74.0
## # i 1 more variable: Borough <chr>
```

10 Step 2: Cleaning Public Restrooms Dataset

```
library(dplyr)
library(stringr)
library(lubridate)

# Convert to tibble for safe dplyr operations
public_restrooms <- tibble::as_tibble(public_restrooms)

# Cleaning Public Restrooms
public_restrooms_filtered <- public_restrooms %>%
  # 1. Keep only operational restrooms
  filter(Status == "Operational") %>%

  # 2. Clean Hours of Operation
  mutate(
    # Extract typical hours like "8am-4pm" (if available)
    Hours_Clean = str_extract(`Hours of Operation`, "\\d{1,2}:?\\d{0,2}(am|pm)-\\d{1,2}:?\\d{0,2}(am|pm)

    # Extract Open and Close times
    Open_Time = str_extract(Hours_Clean, "^\\d{1,2}:?\\d{0,2}(am|pm)"),
    Close_Time = str_extract(Hours_Clean, "(?<=-)\\d{1,2}:?\\d{0,2}(am|pm)"),

    # Convert Open and Close times to numeric 24-hour format
    Open_Hour = case_when(
      !is.na(Open_Time) & str_detect(Open_Time, "am") ~ as.numeric(str_remove(Open_Time, "am")),
```

```
      !is.na(Open_Time) & str_detect(Open_Time, "pm") ~ ifelse(as.numeric(str_remove(Open_Time, "pm")) =
                                                                 12,
                                                                 as.numeric(str_remove(Open_Time, "pm")) +
      TRUE ~ 8  # Default open at 8 AM if missing
    ),
    Close_Hour = case_when(
      !is.na(Close_Time) & str_detect(Close_Time, "am") ~ as.numeric(str_remove(Close_Time, "am")),
      !is.na(Close_Time) & str_detect(Close_Time, "pm") ~ ifelse(as.numeric(str_remove(Close_Time, "pm"
                                                                   12,
                                                                   as.numeric(str_remove(Close_Time, "pm"
      TRUE ~ 16  # Default close at 4 PM if missing
    ),

    #  Update close time to 6 PM (18) if it is still 4 PM (16)
    Close_Hour = ifelse(Close_Hour == 16, 18, Close_Hour),

    # Round Latitude and Longitude
    Latitude = round(Latitude, 4),
    Longitude = round(Longitude, 4),

    # 3. Rename columns here inside mutate
    Facility_Name = `Facility Name`,
    Location_Type = `Location Type`,
    Hours_of_Operation = `Hours of Operation`
  ) %>%

  # 4. Select only the needed columns now
  dplyr::select(
    Facility_Name,
    Location_Type,
    Operator,
    Status,
    Latitude,
    Longitude,
    Accessibility,
    Hours_of_Operation,
    Open_Hour,
    Close_Hour
  )

# View cleaned data
head(public_restrooms_filtered, 10)
```

```
## # A tibble: 10 x 10
##    Facility_Name  Location_Type Operator Status Latitude Longitude Accessibility
##    <chr>          <chr>         <chr>    <chr>     <dbl>     <dbl> <chr>
##  1 Passerelle Bu~ Park          NYC Par~ Opera~     40.8     -73.8 <NA>
##  2 The High Line~ Park          NYC Par~ Opera~     40.7     -74.0 <NA>
##  3 Corporal Thom~ Park          NYC Par~ Opera~     40.6     -74.1 Fully Access~
##  4 Bushwick Libr~ Library       BPL      Opera~     40.7     -73.9 Fully Access~
##  5 Rienzi Playgr~ Park          NYC Par~ Opera~     40.9     -73.9 Fully Access~
##  6 55 East 52nd ~ Privately Ow~ Park Av~ Opera~     40.8     -74.0 Partially Ac~
##  7 Ottendorfer L~ Library       NYPL     Opera~     40.7     -74.0 Not Accessib~
```

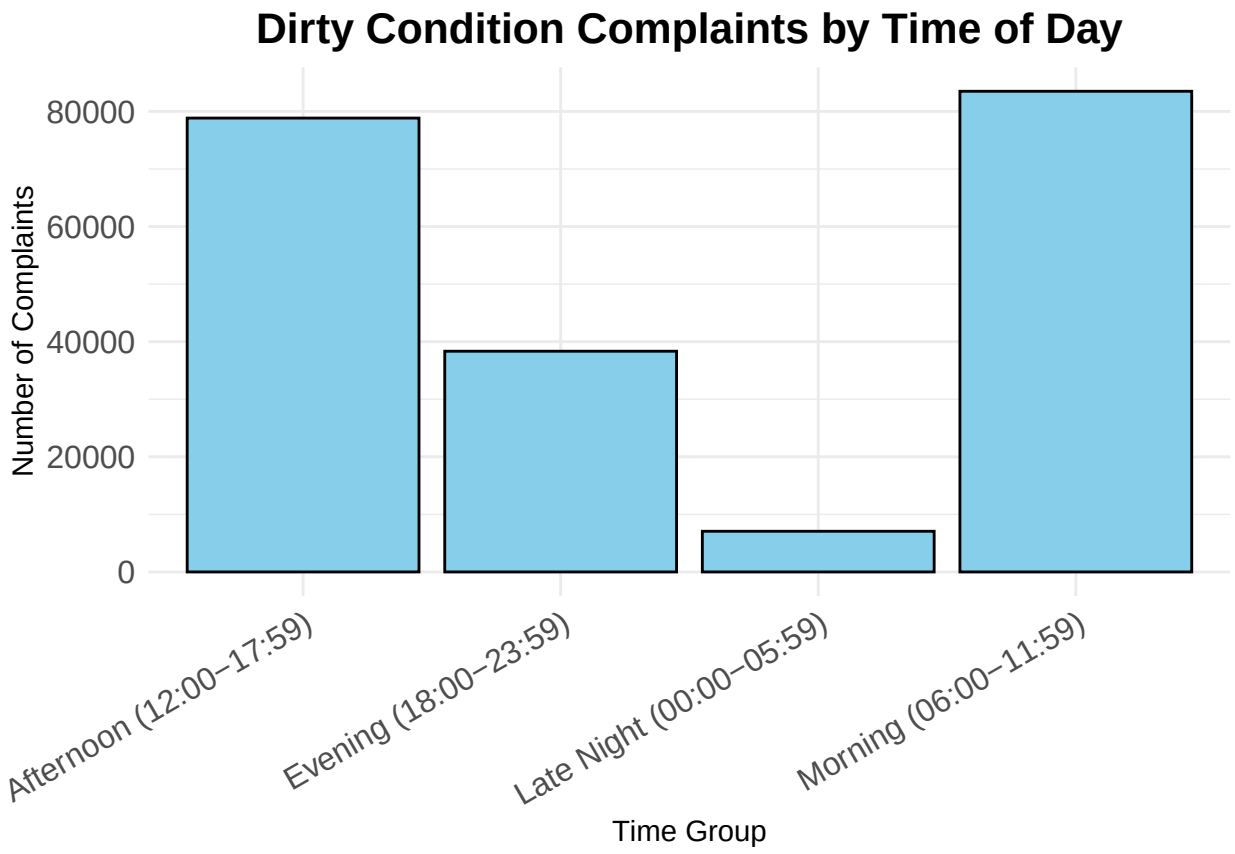
```
##  8 South Beach Z~ Park          NYC Par~ Opera~     40.6     -74.1 Fully Access~
##  9 Grand Central~ Transit       LIRR-MNR Opera~     40.8     -74.0 <NA>
## 10 St. Catherine~ Park          NYC Par~ Opera~     40.8     -74.0 Fully Access~
## # i 3 more variables: Hours_of_Operation <chr>, Open_Hour <dbl>,
## #   Close_Hour <dbl>
```

11 Step 3: EDA - Complaint Volume by Time Group

```
library(ggplot2)
library(dplyr)

# Check how many complaints in each Time Group
complaints_timegroup_summary <- complaints_filtered %>%
  group_by(Time_Group) %>%
  summarise(Complaint_Count = n()) %>%
  arrange(factor(Time_Group,
                 levels = c("Late Night (00:00-05:59)", "Morning (06:00-11:59)", "Afternoon (12:00-17:59
          )

# View summary table
print(complaints_timegroup_summary)
```

```
## # A tibble: 4 x 2
##   Time_Group               Complaint_Count
##   <chr>                              <int>
## 1 Late Night (00:00-05:59)            7064
## 2 Morning (06:00-11:59)              83500
## 3 Afternoon (12:00-17:59)            78846
## 4 Evening (18:00-23:59)              38342
```

```
# Plotting
ggplot(complaints_timegroup_summary, aes(x = Time_Group, y = Complaint_Count)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  theme_minimal() +
  labs(
    title = "Dirty Condition Complaints by Time of Day",
    x = "Time Group",
    y = "Number of Complaints"
  ) +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1, size = 12),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
```



11.0.1 Complaint Volume:

- Highest in **Morning** (~83,500 complaints) and **Afternoon** (~78,800 complaints)
- **Evening** is moderate (~38,300 complaints)
- **Late Night** has the least (~7,000 complaints)

The plot clearly shows the peak times — Morning and Afternoon.

---

12 Step 4: Map Restroom Availability to Time Groups

```
# Load necessary libraries
library(tidyverse)

# Step 4: Define function to map hours to time groups
get_available_timegroups <- function(open_hour, close_hour) {
  timegroups <- c()

  if (!is.na(open_hour) && !is.na(close_hour)) {
```

```
    # Normal cases
    if (open_hour <= 5 && close_hour > 0) {
      timegroups <- c(timegroups, "Late Night (00:00-05:59)")
    }
    if (open_hour <= 11 && close_hour > 6) {
      timegroups <- c(timegroups, "Morning (06:00-11:59)")
    }
    if (open_hour <= 17 && close_hour > 12) {
      timegroups <- c(timegroups, "Afternoon (12:00-17:59)")
    }
    if (open_hour <= 23 && close_hour > 18) {
      timegroups <- c(timegroups, "Evening (18:00-23:59)")
    }

    # If closing hour is smaller (open overnight)
    if (close_hour < open_hour) {
      timegroups <- c("Late Night (00:00-05:59)", "Morning (06:00-11:59)",
                      "Afternoon (12:00-17:59)", "Evening (18:00-23:59)")
    }
  }

  # Combine into single string
  paste(timegroups, collapse = ", ")
}

# Step 4: Apply the function to your public restroom dataset
public_restrooms_filtered <- public_restrooms_filtered %>%
  mutate(
    Available_Time_Groups = mapply(get_available_timegroups, Open_Hour, Close_Hour)
  ) %>%
  as_tibble()

# Step 4: View the selected important columns
public_restrooms_filtered %>%
  dplyr::select(Facility_Name, Open_Hour, Close_Hour, Available_Time_Groups) %>%
  head()
```

```
## # A tibble: 6 x 4
##   Facility_Name                Open_Hour Close_Hour Available_Time_Groups
##   <chr>                            <dbl>      <dbl> <chr>
## 1 Passerelle Building                  8         18 Morning (06:00-11:59), Afte~
## 2 The High Line Zone 1                 8         18 Morning (06:00-11:59), Afte~
## 3 Corporal Thompson Playground         8         18 Morning (06:00-11:59), Afte~
## 4 Bushwick Library, BPL                8         18 Morning (06:00-11:59), Afte~
## 5 Rienzi Playground                    8         18 Morning (06:00-11:59), Afte~
## 6 55 East 52nd Street POPS             8         18 Morning (06:00-11:59), Afte~
```

13 Step 5: Find Distance to Nearest Restroom

Goal: For each complaint, find the nearest public restroom based on latitude and longitude.

- Install the geosphere prior the using it, i.e., run the code : install.packages("geosphere")

```r
# Load necessary library
library(geosphere)

# Corrected function to find nearest restroom
find_nearest_restroom <- function(lat, lon, restrooms_df) {
  if (is.na(lat) || is.na(lon)) return(NA_real_)  # Handle missing

  # Matrix of complaint point
  complaint_point <- matrix(c(lon, lat), nrow = 1)

  # Matrix of restroom points
  restroom_points <- restrooms_df[, c("Longitude", "Latitude")] %>% as.matrix()

  # Calculate distance
  distances <- distHaversine(complaint_point, restroom_points)

  # Return minimum distance
  min(distances)
}

# Now apply safely
complaints_filtered <- complaints_filtered %>%
  mutate(
    Nearest_Restroom_Distance_m = mapply(find_nearest_restroom, Latitude, Longitude,
                                         MoreArgs = list(restrooms_df = public_restrooms_filtered))
  )
```

13.0.1 Summary Statistics of Distances

```r
# Add a new column for distance in kilometers
complaints_filtered <- complaints_filtered %>%
  mutate(
    Nearest_Restroom_Distance_km = Nearest_Restroom_Distance_m / 1000
  )
```

```r
# Basic summary
summary(complaints_filtered$Nearest_Restroom_Distance_km)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.008   0.226   0.352   0.405   0.513   3.795    4014
```

13.0.2 Summary of Nearest Restroom Distances

After calculating the distance between each complaint and the nearest public restroom, we summarized the distances in **kilometers**.

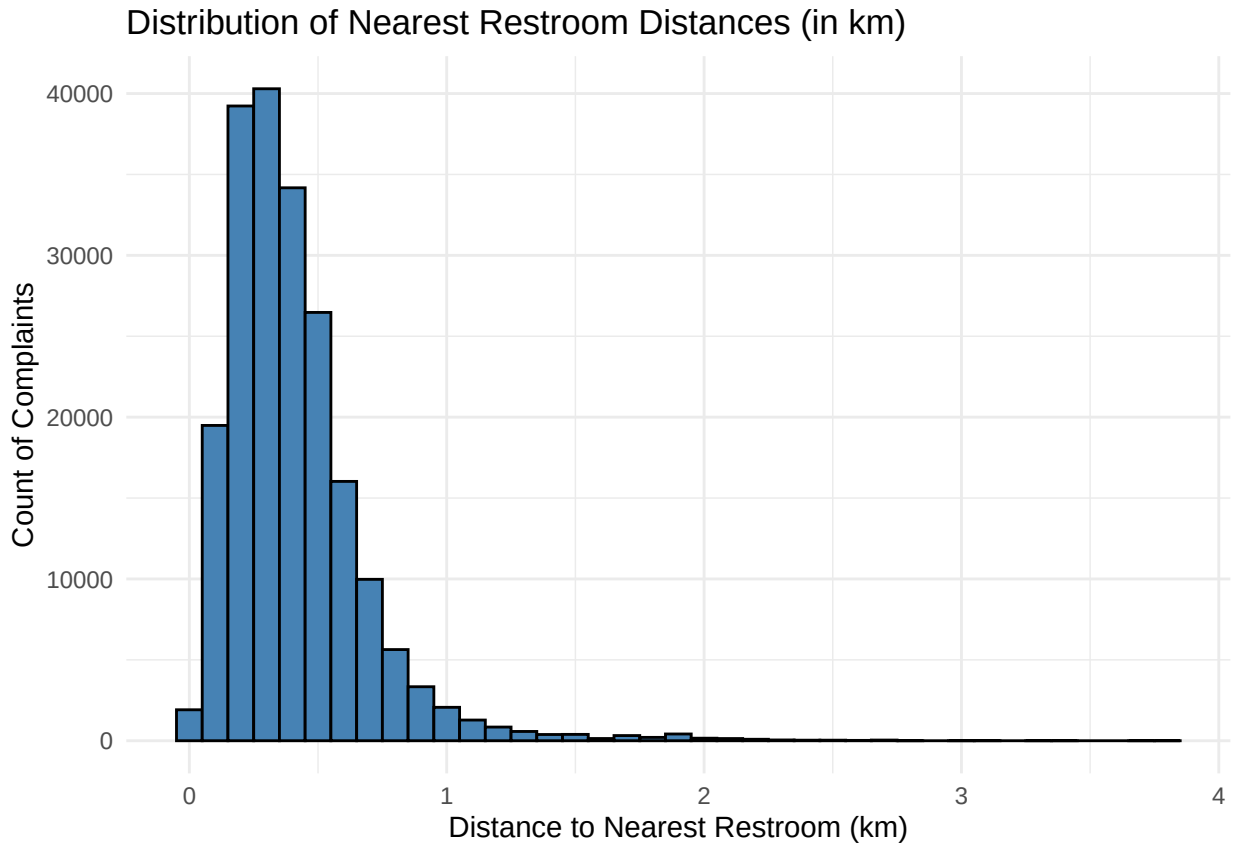
The key observations are:

- The **minimum** distance to a restroom was approximately **8 meters (0.008 km)**.
- The **median** distance was around **352 meters (0.352 km)**.

- On average, complaints were located about **405 meters (0.405 km)** away from the nearest restroom.
- **75%** of complaints had a restroom within **513 meters (0.513 km)**.
- The **maximum** distance to a restroom observed was around **3.8 kilometers**.
- There are **4014 complaints** with missing location data (either latitude or longitude), resulting in missing distance values.

This summary helps us understand that most complaints happened relatively **close to public restrooms**, but there are still a few cases where restrooms were farther away, possibly impacting complaint rates.

```
# Plot histogram of distances (in kilometers)

library(ggplot2)

ggplot(complaints_filtered, aes(x = Nearest_Restroom_Distance_km)) +
  geom_histogram(binwidth = 0.1, fill = "steelblue", color = "black") +
  labs(
    title = "Distribution of Nearest Restroom Distances (in km)",
    x = "Distance to Nearest Restroom (km)",
    y = "Count of Complaints"
  ) +
  theme_minimal()
```



13.0.3 Summary

The histogram shows the distribution of distances from complaint locations to the nearest public restroom.

- Most complaints are located within **0.5 km** of a restroom.
- A very large number of complaints cluster at distances between **0.2 km and 0.6 km**.
- Very few complaints occur farther than **1.5 km** away.
- The distribution is **right-skewed**, meaning extreme distance complaints are rare.

Interpretation of Distance Distribution:

The distance analysis shows that most "Dirty Condition" complaints occur relatively close to a public restroom — often within **500 meters** (0.5 km).
This suggests that the city has made reasonable efforts to **place** restrooms near populated areas.

However, the presence of a restroom nearby does **not necessarily** mean that it is **clean**, **accessible**, or **well maintained**.

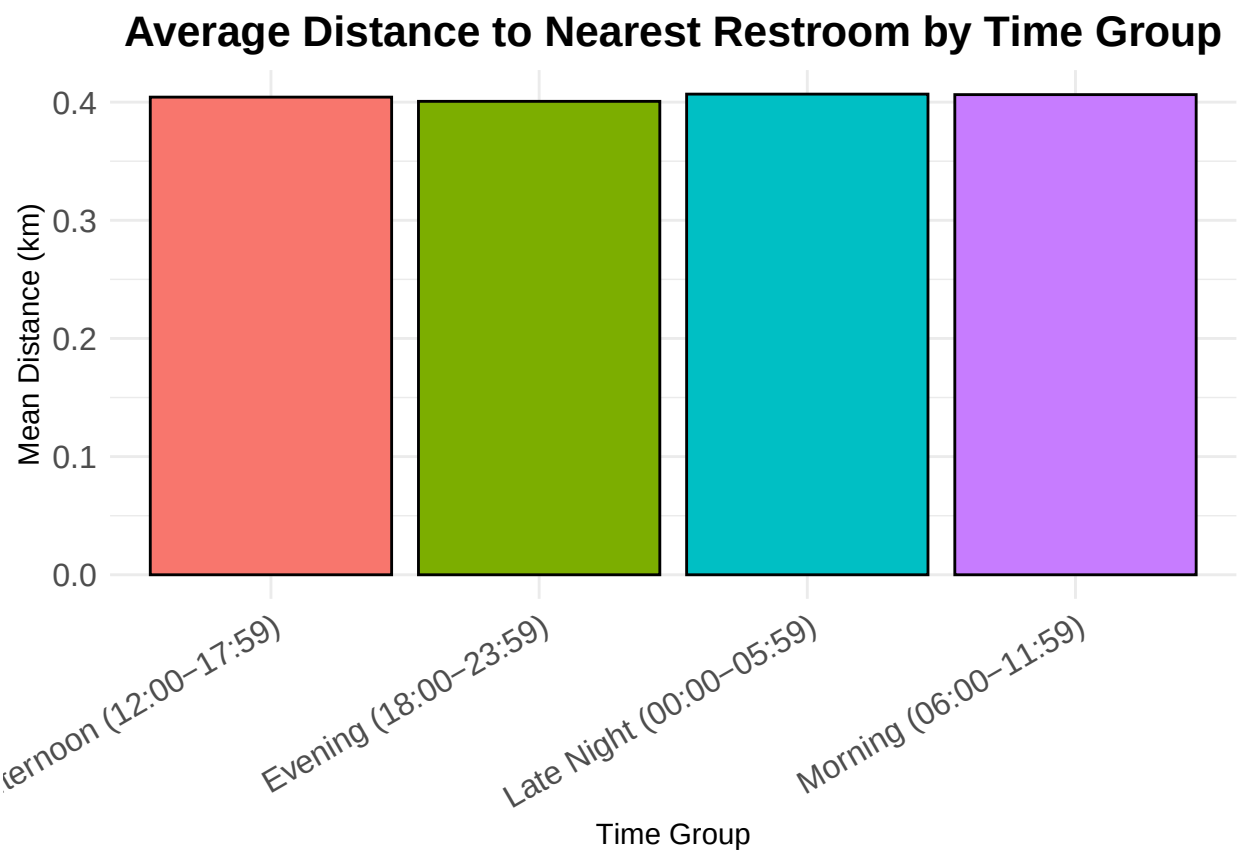
14 Step 6: Analyze Nearest Restroom Distance by Time Groups

```
library(dplyr)

# Group by Time_Group and summarize distance statistics
timegroup_distance_summary <- complaints_filtered %>%
  group_by(Time_Group) %>%
  summarise(
    Count = n(),
    Mean_Distance_km = mean(Nearest_Restroom_Distance_km, na.rm = TRUE),
    Median_Distance_km = median(Nearest_Restroom_Distance_km, na.rm = TRUE),
    Max_Distance_km = max(Nearest_Restroom_Distance_km, na.rm = TRUE)
  )

# View the summary table
print(timegroup_distance_summary)
```

```
## # A tibble: 4 x 5
##   Time_Group           Count Mean_Distance_km Median_Distance_km Max_Distance_km
##   <chr>                <int>            <dbl>              <dbl>           <dbl>
## 1 Afternoon (12:00-17~ 78846            0.404              0.355            3.33
## 2 Evening (18:00-23:5~ 38342            0.401              0.353            3.41
## 3 Late Night (00:00-0~  7064            0.407              0.358            2.68
## 4 Morning (06:00-11:5~ 83500            0.406              0.349            3.80
```

```
# Load necessary library
library(ggplot2)
# Plot Mean Distance by Time Group
ggplot(timegroup_distance_summary, aes(x = Time_Group, y = Mean_Distance_km, fill = Time_Group)) +
  geom_bar(stat = "identity", color = "black") +
  theme_minimal() +
```

```
  labs(
    title = "Average Distance to Nearest Restroom by Time Group",
    x = "Time Group",
    y = "Mean Distance (km)"
  ) +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1, size = 12),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "none"
  )
```



14.0.1 Observations:

- The **mean distance** to the nearest restroom is very consistent across all time groups, hovering around **0.40–0.41 km**.
- **Morning** and **Late Night** show slightly higher average distances (~0.406 km), compared to **Afternoon** (~0.404 km) and **Evening** (~0.400 km).
- **Maximum distances** can reach up to **3.7 km** in the Morning period, suggesting that some locations still lack close restroom access.
- **Conclusion**: Accessibility is fairly even across time periods, though there is slightly **less coverage during Morning and Late Night** hours.

15 Step 7: Match Each Complaint with Nearest Restroom and Availability Check

After calculating the nearest restroom for each complaint, we now check whether the nearest restroom was operational during the complaint's time group (Morning, Afternoon, Evening, Late Night).

We perform the following:

- For each complaint, we find the closest public restroom based on geospatial distance (using the Haversine formula for great-circle distance).
- We check the restroom's available operating hours against the complaint's time group.
- If the restroom was available during that time, we flag it as `Yes`; otherwise, `No`.

This helps assess whether restroom unavailability might be linked to cleanliness complaints.

A new column called `Restroom_Available` is added:

- `Yes` → restroom open during the complaint time.
- `No` → restroom closed or unavailable during that period.

```
# Load libraries
library(dplyr)
library(purrr)
library(stringr)
library(geosphere)

# Pre-compute the matrix of restrooms outside mutate
restroom_coords <- as.matrix(public_restrooms_filtered[, c("Longitude", "Latitude")])

# Proceed with complaint matching
complaints_nearest_joined <- complaints_filtered %>%
  mutate(
    Nearest_Restroom_ID = map2_dbl(Latitude, Longitude, function(lat, lon) {
      if (is.na(lat) || is.na(lon)) return(NA_real_)
      complaint_point <- matrix(c(lon, lat), nrow = 1)
      distances <- distHaversine(complaint_point, restroom_coords)
      which.min(distances)
    })
  ) %>%
  mutate(
    Available_Time_Groups = public_restrooms_filtered$Available_Time_Groups[Nearest_Restroom_ID]
  ) %>%
  rowwise() %>%
  mutate(
    Restroom_Available = case_when(
      !is.na(Available_Time_Groups) & {
        split_groups <- str_split(Available_Time_Groups, ",")[[1]] %>% str_trim()
        Time_Group %in% split_groups
      } ~ "Yes",
      TRUE ~ "No"
    )
```

```
  ) %>%
  ungroup()

# Quick preview
complaints_nearest_joined %>%
  dplyr::select(Time_Group, Available_Time_Groups, Restroom_Available) %>%
  head(20)
```

```
## # A tibble: 20 x 3
##    Time_Group            Available_Time_Groups                Restroom_Available
##    <chr>                 <chr>                                <chr>
##  1 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  2 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  3 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  4 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  5 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  6 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  7 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  8 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
##  9 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 10 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 11 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 12 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 13 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 14 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 15 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 16 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 17 Evening (18:00-23:59) <NA>                                 No
## 18 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 19 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
## 20 Evening (18:00-23:59) Morning (06:00-11:59), Afternoon (1~ No
```

```
# Group by Time Group and Restroom Availability
complaints_nearest_joined %>%
  group_by(Time_Group, Restroom_Available) %>%
  summarise(Count = n()) %>%
  arrange(Time_Group)
```

```
## # A tibble: 8 x 3
## # Groups:   Time_Group [4]
##   Time_Group               Restroom_Available Count
##   <chr>                    <chr>              <int>
## 1 Afternoon (12:00-17:59)  No                  1799
## 2 Afternoon (12:00-17:59)  Yes                77047
## 3 Evening (18:00-23:59)    No                 38018
## 4 Evening (18:00-23:59)    Yes                  324
## 5 Late Night (00:00-05:59) No                  7059
## 6 Late Night (00:00-05:59) Yes                    5
## 7 Morning (06:00-11:59)    No                  1470
## 8 Morning (06:00-11:59)    Yes                82030
```

We analyzed whether a restroom was available at the time each complaint was registered.

15.0.1 Key Observations:

- A majority of complaints in the **Afternoon** (12:00–17:59) and **Morning** (06:00–11:59) had a nearby restroom available.
- However, a small portion of complaints (~2%) in the **Afternoon** and **Morning** reported **no restroom availability**.
- In the **Evening** (18:00–23:59) and **Late Night** (00:00–05:59), restroom availability was noticeably lower.
- Particularly in the **Evening**, most complaints did not have a nearby restroom available, highlighting a coverage gap after 6 PM.

Interpretation:

- "No restroom available" means that the nearest restroom was either closed at the time of the complaint or its operational hours did not align with the complaint's time group.
- This does not imply that there were absolutely no restrooms in the city, but rather a local mismatch at that specific location and time.

Overall Conclusion:

- Restroom accessibility is strong during daytime (Morning and Afternoon).
- Evening and Late Night periods show room for improvement, possibly by extending restroom operating hours or installing additional 24-hour facilities.

16 Step 8: Statistical Modeling

To understand relationship between variables like:

Time_Group

Restroom_Available (Yes/No)

Borough

16.1 Multiple Linear Regression Analysis

```
# Load necessary library
library(dplyr)

# Prepare data: keep only needed columns and remove NA distances
regression_data <- complaints_nearest_joined %>%
  dplyr::select(Nearest_Restroom_Distance_km, Time_Group, Restroom_Available, Borough) %>%
  filter(!is.na(Nearest_Restroom_Distance_km))  # Remove missing distance records

# Convert Time_Group, Restroom_Available, and Borough to factors
regression_data <- regression_data %>%
  mutate(
    Time_Group = as.factor(Time_Group),
```

```r
    Restroom_Available = as.factor(Restroom_Available),
    Borough = as.factor(Borough)
  )

# Build the multiple linear regression model
distance_model <- lm(Nearest_Restroom_Distance_km ~ Time_Group + Restroom_Available + Borough, data = re

# Display the model summary
summary(distance_model)
```

```
##
## Call:
## lm(formula = Nearest_Restroom_Distance_km ~ Time_Group + Restroom_Available +
##     Borough, data = regression_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7030 -0.1500 -0.0297  0.1173  3.3373
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         0.384480   0.013025  29.518  < 2e-16 ***
## Time_GroupEvening (18:00-23:59)    -0.040331   0.012931  -3.119  0.00182 **
## Time_GroupLate Night (00:00-05:59) -0.039185   0.013298  -2.947  0.00321 **
## Time_GroupMorning (06:00-11:59)    -0.001146   0.001231  -0.932  0.35156
## Restroom_AvailableYes              -0.037594   0.012952  -2.903  0.00370 **
## BoroughBROOKLYN                     0.049366   0.001668  29.602  < 2e-16 ***
## BoroughMANHATTAN                   -0.063345   0.001836 -34.509  < 2e-16 ***
## BoroughQUEENS                       0.112139   0.001793  62.557  < 2e-16 ***
## BoroughSTATEN ISLAND                0.370105   0.002369 156.236  < 2e-16 ***
## BoroughUnspecified                  0.068037   0.022507   3.023  0.00250 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.245 on 203728 degrees of freedom
## Multiple R-squared:  0.166,  Adjusted R-squared:  0.166
## F-statistic:  4506 on 9 and 203728 DF,  p-value: < 2.2e-16
```

16.2 Multiple Linear Regression Summary

We built a multiple linear regression model to predict the **Nearest Restroom Distance (in km)** based on:

- **Time Group** (Morning, Afternoon, Evening, Late Night)
- **Restroom Availability** (Yes/No)
- **Borough** (Brooklyn, Manhattan, Queens, Staten Island, Unspecified)

Significance stars

- *** means very important (p-value < 0.001).
- ** means important (p-value < 0.01).

- – means marginally important (p-value < 0.05).
- No stars means not significant.

The model output:

- **Intercept**: Complaints during the Afternoon, with no restroom available, have a base distance of approximately **0.384 km**.
- **Time Group Effects**:
  - Complaints in the **Evening** and **Late Night** have slightly **lower distances** to restrooms (~0.04 km closer).
  - Complaints during **Morning** hours did not show a significant difference compared to Afternoon.
- **Restroom Availability**:
  - If a restroom was **available**, the nearest restroom was about **0.037 km closer** on average.
- **Borough Effects**:
  - Complaints in **Brooklyn** and **Queens** were associated with **larger distances** compared to the baseline borough.
  - Complaints in **Manhattan** had **closer restroom distances** by about 0.063 km.
  - **Staten Island** showed the highest positive effect on distance (0.370 km farther).

16.2.1 Model Performance:

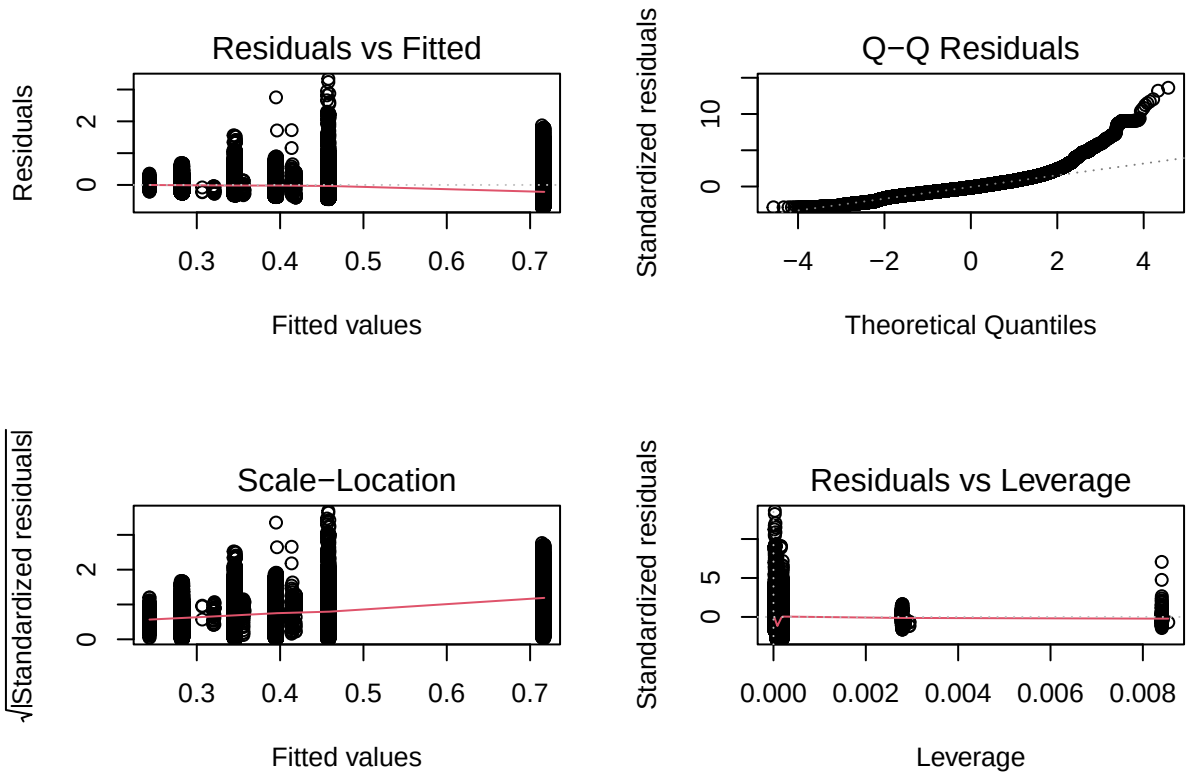
- **R-squared** = 0.166 → The model explains about **16.6%** of the variance in restroom distance.
- **Residual standard error** = 0.245 km → Average prediction error is around **245 meters**.
- **Overall model p-value** < 2.2e-16 → The model is **statistically significant**.

16.3 Regression Diagnostics

To further evaluate the quality of the regression model, we plot standard diagnostic plots:

- Residuals vs Fitted
- Normal Q-Q
- Scale-Location
- Residuals vs Leverage

```
# Base R plotting without cluttered labels
par(mfrow = c(2, 2))  # Arrange 4 plots in a 2x2 layout

# Plot individual diagnostics
plot(distance_model, which = 1, cook.levels = NULL, id.n = 0)  # Residuals vs Fitted
plot(distance_model, which = 2, cook.levels = NULL, id.n = 0)  # Normal Q-Q
plot(distance_model, which = 3, cook.levels = NULL, id.n = 0)  # Scale-Location
plot(distance_model, which = 5, cook.levels = NULL, id.n = 0)  # Residuals vs Leverage
```




```r
par(mfrow = c(1, 1))  # Reset plotting layout back to default
```

16.3.1 Interpretation of Regression Diagnostics

The regression diagnostic plots provide insight into the assumptions underlying the multiple linear regression model:

- **Residuals vs Fitted Plot** shows a relatively random scatter around the horizontal line at zero, indicating that the model captures the main structure of the data reasonably well, although there may be some non-constant variance at certain fitted values.
- **Normal Q-Q Plot** shows that most residuals lie along the diagonal line, suggesting approximate normality of errors. However, there are slight deviations at the extremes, implying some minor departures from perfect normality.
- **Scale-Location Plot** (also called Spread-Location) indicates that the variance of residuals is fairly consistent across fitted values, supporting the assumption of homoscedasticity (constant variance).
- **Residuals vs Leverage Plot** highlights that there are no extreme high-leverage points significantly influencing the model, suggesting that the model is robust and no single observation unduly biases the results.

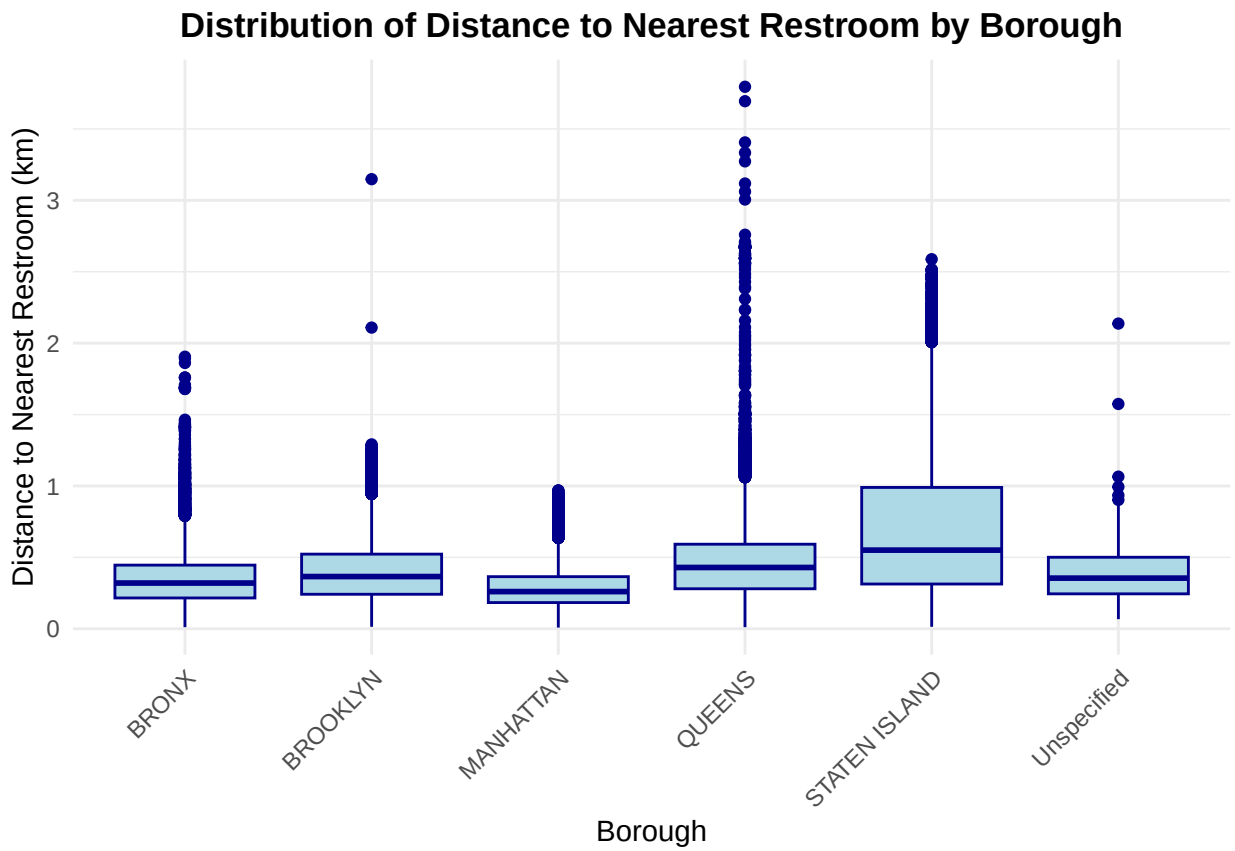
Overall, the diagnostics suggest that the multiple linear regression model is reasonably valid for interpretation and further inference, with only mild deviations that are acceptable for large datasets like this one.

16.3.2 Conclusion:

While restroom distance is moderately influenced by Borough, Time Group, and Availability, a large part of the variation remains unexplained, suggesting other factors (like geography, urban density, and park design) may also affect restroom accessibility.

16.4 Visualization: Borough vs. Restroom Distance

```
library(ggplot2)

# Basic plot
regression_data %>%
  ggplot(aes(x = Borough, y = Nearest_Restroom_Distance_km)) +
  geom_boxplot(fill = "lightblue", color = "darkblue") +
  labs(
    title = "Distribution of Distance to Nearest Restroom by Borough",
    x = "Borough",
    y = "Distance to Nearest Restroom (km)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



16.4.1 Observations:

- *Manhattan* shows the shortest median distance to public restrooms, indicating a very high restroom density.
- *Brooklyn* and *Bronx* have slightly longer distances compared to Manhattan, but still relatively well-served.
- *Queens* and *Staten Island* have noticeably higher median distances and wider spread, suggesting less restroom accessibility.
- *Staten Island* especially shows many extreme values (outliers), confirming larger restroom gaps.
- *Unspecified* boroughs show mixed accessibility patterns.

Conclusion:
Public restroom accessibility is best in Manhattan and relatively moderate in Brooklyn and Bronx, but is poorer in Queens and Staten Island.

16.5 ANOVA TEST

```
# Perform ANOVA to test if restroom distances differ by Borough
borough_anova <- aov(Nearest_Restroom_Distance_km ~ Borough, data = regression_data)

# View summary of ANOVA results
summary(borough_anova)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Borough          5   2434   486.8    8107 <2e-16 ***
## Residuals   203732  12234     0.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

16.6 Choosing the Correct Statistical Test

When deciding between **Two-Sample t-Test** and **ANOVA**, the main factor to consider is the **number of groups** are comparing:

Test	Best For	Example
Two-Sample t-Test	Comparing **two groups**	Comparing *Manhattan* vs *Queens*
ANOVA + Post Hoc (Tukey)	Comparing **three or more groups**	Comparing *Bronx, Brooklyn, Queens, etc.*

16.6.1 Why We Used ANOVA:

- In this analysis, we are comparing **multiple Boroughs** (*Bronx, Brooklyn, Manhattan, Queens, Staten Island, Unspecified*).
- **ANOVA** is appropriate when testing whether **there is any significant difference** in restroom distances among **more than two groups**.

- Running multiple t-tests between each pair would **increase the chance of false positives** (inflated Type I error).
- ANOVA controls for this and provides a global test of differences.

16.6.2 When to Use Two-Sample t-Test:

- If we specifically want to compare **only two Boroughs** (e.g., *Brooklyn vs Queens*), a Two-Sample t-Test is suitable.
- However, in such cases, corrections for multiple testing must be considered if conducting many pairwise comparisons.

16.6.3 Next Step After Significant ANOVA:

- Since our ANOVA result was significant (p-value < 0.001), it tells us **at least one Borough** has a different mean distance.
- To find **which Borough pairs differ**, we should use a **Tukey HSD (Honest Significant Difference) Post Hoc Test**.
- Tukey test adjusts for multiple comparisons safely.

Summary:

- Use **ANOVA** for 3+ groups first.
- If significant, use **Tukey HSD** for detailed pairwise comparisons.
- Use **t-Test** only when comparing exactly **two groups** intentionally.

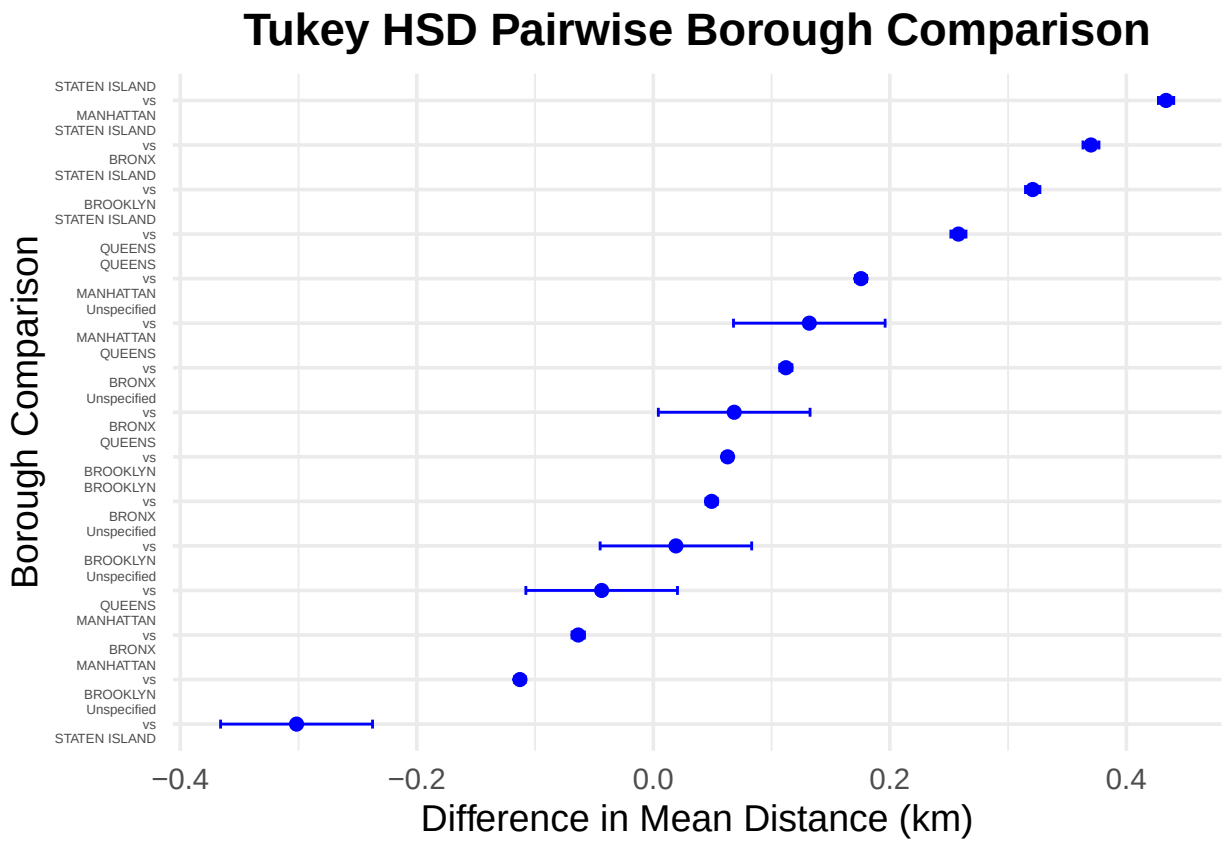
16.7 Post-Hoc Test (Tukey HSD):

```
# Tukey's Honest Significant Difference (HSD) Test
borough_tukey <- TukeyHSD(borough_anova)

# View Tukey Test Results
borough_tukey
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Nearest_Restroom_Distance_km ~ Borough, data = regression_data)
##
## $Borough
```

```
##                                  diff          lwr         upr     p adj
## BROOKLYN-BRONX              0.04929129  0.044540050  0.05404253 0.0000000
## MANHATTAN-BRONX            -0.06351471 -0.068742709 -0.05828671 0.0000000
## QUEENS-BRONX                0.11208330  0.106975945  0.11719065 0.0000000
## STATEN ISLAND-BRONX         0.37007496  0.363328015  0.37682191 0.0000000
## Unspecified-BRONX           0.06838693  0.004252208  0.13252166 0.0287128
## MANHATTAN-BROOKLYN         -0.11280600 -0.117158386 -0.10845361 0.0000000
## QUEENS-BROOKLYN             0.06279201  0.058585306  0.06699871 0.0000000
## STATEN ISLAND-BROOKLYN      0.32078367  0.314690021  0.32687732 0.0000000
## Unspecified-BROOKLYN        0.01909564 -0.044973649  0.08316494 0.9581279
## QUEENS-MANHATTAN            0.17559801  0.170859436  0.18033658 0.0000000
## STATEN ISLAND-MANHATTAN     0.43358967  0.427117399  0.44006194 0.0000000
## Unspecified-MANHATTAN       0.13190164  0.067795230  0.19600805 0.0000001
## STATEN ISLAND-QUEENS        0.25799166  0.251616451  0.26436688 0.0000000
## Unspecified-QUEENS         -0.04369636 -0.107793049  0.02040032 0.3759360
## Unspecified-STATEN ISLAND -0.30168803 -0.365936151 -0.23743991 0.0000000
```

```r
# install.packages("multcompView")
# install.packages("broom")
# install.packages("forcats")


# Load libraries
library(dplyr)
library(tidyr)
library(ggplot2)
library(broom)
library(stringr)
library(forcats)

# 1. Tidy Tukey HSD
tidy_tukey <- broom::tidy(borough_tukey)

# 2. Split contrast into Borough1 and Borough2
tidy_tukey <- tidy_tukey %>%
  separate(contrast, into = c("Borough1", "Borough2"), sep = "-", remove = FALSE) %>%
  mutate(
    comparison_label = paste(Borough1, "vs", Borough2, sep = "\n")
  )

# 3. Plot nicely
ggplot(tidy_tukey, aes(x = estimate, y = fct_reorder(comparison_label, estimate))) +
  geom_point(color = "blue", size = 2) +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.2, color = "blue") +
  labs(
    title = "Tukey HSD Pairwise Borough Comparison",
    x = "Difference in Mean Distance (km)",
    y = "Borough Comparison"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text.y = element_text(size = 5)
  )
```



16.7.1 Interpretation of Tukey HSD Pairwise Comparison

The Tukey HSD (Honestly Significant Difference) test helps identify which boroughs have statistically different average distances to the nearest restroom. In the plot, each dot represents the mean difference between two boroughs, and the horizontal lines show the 95% confidence interval.

If a borough pair's confidence interval does **not cross zero**, it means there is a statistically significant difference in restroom distance between them. If the interval **crosses zero**, the difference is not statistically significant.

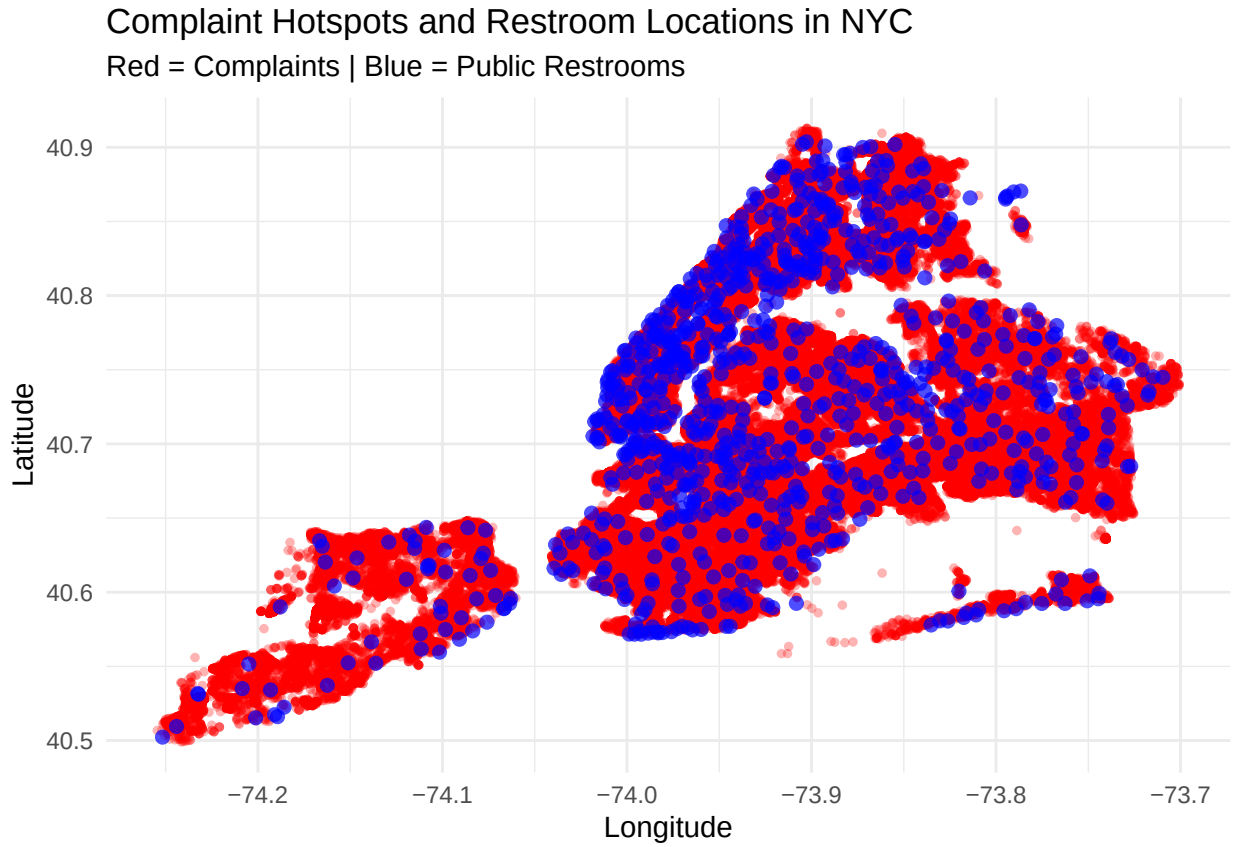
From the plot, it is clear that Staten Island consistently shows significantly greater distances compared to other boroughs like Manhattan, Brooklyn, and Queens, indicating poorer restroom accessibility. In contrast, boroughs like Brooklyn, Bronx, and Manhattan show smaller mean differences and often overlap, suggesting similar restroom accessibility levels between them.

Overall, this test confirms that restroom accessibility varies meaningfully across boroughs, with Staten Island needing particular attention for improvements.

16.8 Mapping Complaint Hotspots and Restroom Locations

```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Filter out complaints and restrooms with missing coordinates
```

```r
complaints_map_data <- complaints_nearest_joined %>%
  filter(!is.na(Latitude) & !is.na(Longitude))

restrooms_map_data <- public_restrooms_filtered %>%
  filter(!is.na(Latitude) & !is.na(Longitude))

# Plot the map
ggplot() +
  # Complaints as points
  geom_point(data = complaints_map_data, aes(x = Longitude, y = Latitude),
             color = "red", alpha = 0.3, size = 1) +
  # Restrooms as points
  geom_point(data = restrooms_map_data, aes(x = Longitude, y = Latitude),
             color = "blue", alpha = 0.7, size = 2) +
  labs(
    title = "Complaint Hotspots and Restroom Locations in NYC",
    subtitle = "Red = Complaints | Blue = Public Restrooms",
    x = "Longitude",
    y = "Latitude"
  ) +
  theme_minimal()
```



16.8.1 Mapping Complaint Hotspots and Public Restroom Locations

The visual map shows the distribution of cleanliness complaints (red dots) alongside public restroom locations (blue dots) across NYC.

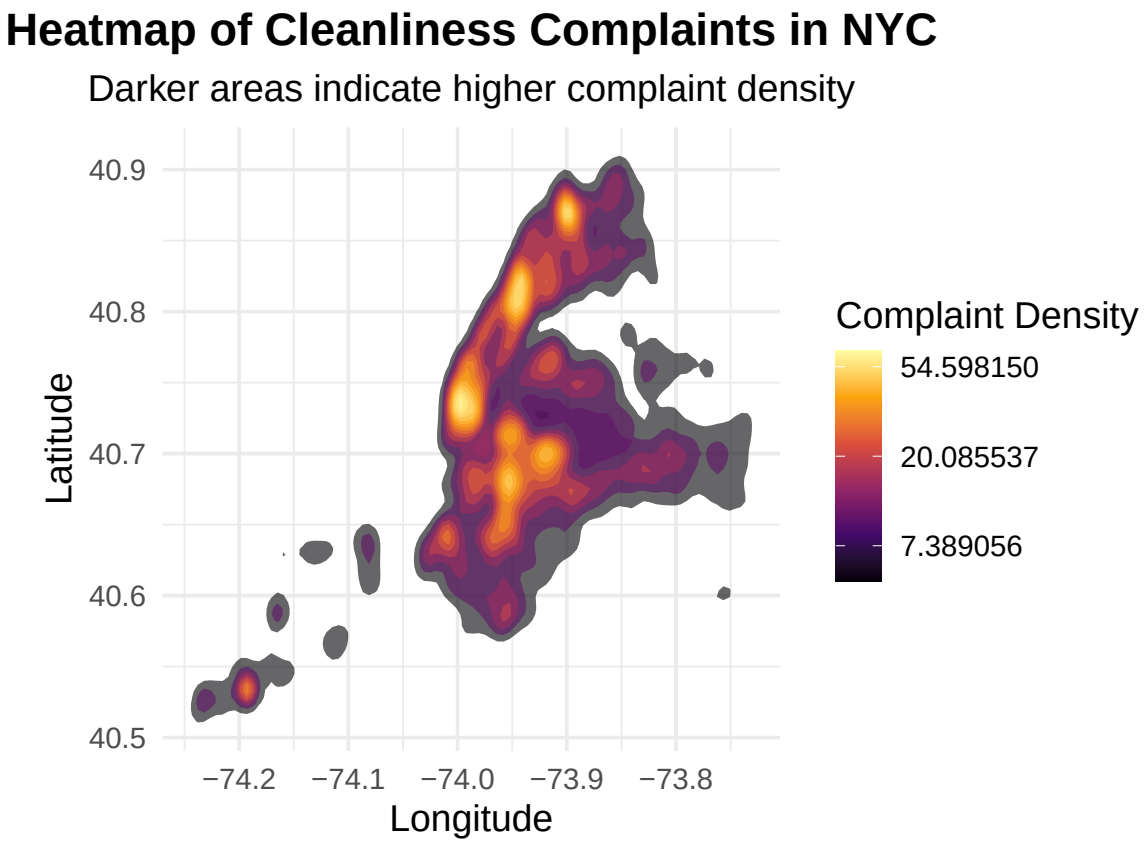
Key Observations: - Dense urban areas like Manhattan, Brooklyn, and Queens have a high volume of complaints.

- Public restrooms are also concentrated in these dense boroughs but do not fully cover all complaint-prone areas.
- Peripheral regions like parts of Queens and Staten Island show a relative scarcity of restrooms despite complaints being reported.
- Overlaps between complaints and restrooms in central areas may indicate high foot traffic and usage stresses, potentially leading to more reported issues.

This mapping highlights spatial gaps and potential regions for prioritizing new restroom installations or improved maintenance efforts.

16.9 Complaint Heatmap

```
# Load libraries
library(ggplot2)

# Plot Complaint Heatmap
ggplot(complaints_filtered, aes(x = Longitude, y = Latitude)) +
  stat_density2d(aes(fill = ..level..), geom = "polygon", alpha = 0.6) +
  scale_fill_viridis_c(option = "inferno", trans = "log") +  # log transform for better contrast
  coord_fixed(1.3) +
  labs(
    title = "Heatmap of Cleanliness Complaints in NYC",
    subtitle = "Darker areas indicate higher complaint density",
    x = "Longitude",
    y = "Latitude",
    fill = "Complaint Density"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5)
  )
```



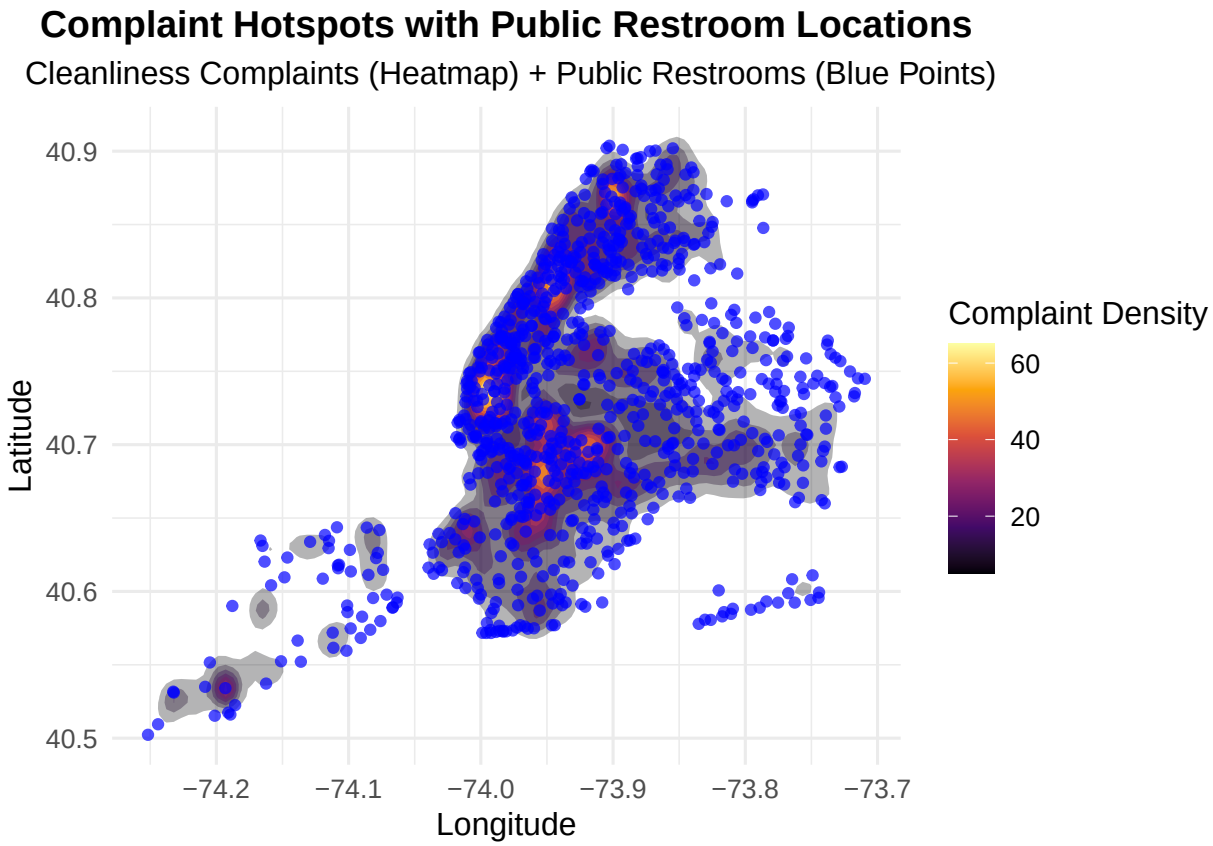
16.9.1 Observations from the Heatmap

The heatmap visualization highlights where cleanliness complaints are most concentrated across NYC:

- **Hotspot Areas**:
  - The highest densities (bright yellow/orange areas) are concentrated around **midtown and downtown Manhattan**, some parts of **Brooklyn**, and a few locations in **Queens**.
  - **Staten Island** shows relatively fewer complaint hotspots compared to other boroughs.
- **Interpretation**:
  - These densely colored regions suggest areas where restroom availability might be insufficient relative to usage, potentially leading to cleanliness complaints.
  - It also points to potential needs for either better restroom maintenance or additional restroom facilities in those busy zones.
- **Next Steps**:
  - Compare hotspot zones with public restroom coverage.
  - Prioritize interventions (installing new restrooms or improving maintenance) in areas with high complaint density and sparse restroom availability.

16.10 Heatmap of complaints and public restrooms

```r
# Load libraries
library(ggplot2)

# Create combined heatmap + points
ggplot() +
  # Heatmap layer for complaints
  stat_density_2d(
    data = complaints_filtered,
    aes(x = Longitude, y = Latitude, fill = ..level.., alpha = ..level..),
    geom = "polygon",
    color = NA,
    contour = TRUE
  ) +
  scale_fill_viridis_c(option = "inferno", name = "Complaint Density") +
  scale_alpha(range = c(0.3, 0.8), guide = FALSE) +

  # Overlay restrooms as blue points
  geom_point(
    data = public_restrooms_filtered,
    aes(x = Longitude, y = Latitude),
    color = "blue", size = 1.5, alpha = 0.7
  ) +

  # Labels and theme
  labs(
    title = "Complaint Hotspots with Public Restroom Locations",
    subtitle = " Cleanliness Complaints (Heatmap) + Public Restrooms (Blue Points)",
    x = "Longitude",
    y = "Latitude"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5)
  )
```



16.10.1 Insights from Mapping Complaint Hotspots and Restroom Locations

By overlaying public restroom locations (blue points) onto the complaint density heatmap, several key insights emerge:

- Areas with **a higher density of public restrooms** generally correspond to **lower complaint density**, suggesting that the availability of restrooms helps mitigate cleanliness complaints.
- Conversely, **complaint hotspots** (dark regions in the heatmap) often appear in locations where **few or no restrooms** are present, implying that the absence of accessible facilities could contribute to dissatisfaction and reported cleanliness issues.
- Boroughs like **Manhattan** and parts of **Brooklyn** exhibit relatively good restroom coverage and lighter complaint densities, whereas outer areas of **Queens** and **Staten Island** show clusters of complaints with fewer restrooms nearby.
- Overall, the spatial patterns highlight a potential **positive impact of restroom accessibility** on urban cleanliness perceptions and suggest opportunities for **targeted restroom placement** in underserved regions.

This geospatial analysis strengthens the hypothesis that **better restroom infrastructure correlates with reduced cleanliness complaints** across NYC.

17 Conclusion and Recommendations

- his study investigated the relationship between the availability of public restrooms and cleanliness-related complaints across New York City. Using 311 complaint data and public restroom datasets, we conducted an extensive geospatial and statistical analysis.
- Our initial findings showed that the average distance from a complaint location to the nearest public restroom is approximately 0.4 kilometers — relatively accessible on a citywide scale. However, this average masks significant borough-level disparities. Regression and ANOVA analysis revealed that boroughs like Staten Island and Queens have significantly greater distances to restrooms compared to Manhattan, where public restrooms are more densely distributed.
- The multiple linear regression further confirmed that both borough and restroom availability at the complaint time significantly influence the distance to the nearest restroom. The Tukey post-hoc test provided detailed pairwise comparisons, affirming that borough differences are statistically significant.
- Spatial heatmaps visualizing complaint density and restroom availability showed that areas with higher restroom density, such as Manhattan and parts of Brooklyn, tend to have lower complaint volumes. Conversely, boroughs with fewer restrooms relative to population and activity levels — notably Staten Island and Queens — exhibited clusters of complaints without proximate facilities.

Based on these insights, the *following recommendations are proposed:*

- **Expand Restroom Infrastructure**: Invest in installing additional public restrooms in underserved boroughs, prioritizing Queens and Staten Island, where restroom coverage gaps are prominent.
- **Target Hotspot Areas**: Use complaint density maps to strategically place restrooms in areas with high complaint frequencies but limited restroom access.
- **Extend Restroom Operating Hours**: Many public restrooms close by early evening. Extending operational hours, particularly in busy commercial or residential zones, may reduce cleanliness-related complaints during evening and late-night periods.
- **Improve Accessibility**: Ensure that new restrooms are fully accessible to individuals with disabilities, aligning with equity and inclusion goals.
- **Continuous Monitoring**: Regularly update and cross-reference 311 complaint data with restroom operational data to dynamically adjust restroom deployment strategies.

In conclusion, while New York City's public restroom system serves many neighborhoods effectively, targeted improvements guided by data can help bridge accessibility gaps, enhance public satisfaction, and promote cleaner urban environments citywide.