Journal of Data Science 0 (0), 1–15 DOI: 0000
2025

Supplementary Materials for:

Principles for Open Data Curation: A Case Study with the New York City 311 Service Request Data

David Tussey[1] and Jun Yan[2]

[1]*NYC Department of Information Technology and Telecommunications, USA*
[2]*Department of Statistics, University of Connecticut, USA*

Keywords *Data cleansing; Data quality; Data science; Data Validation; NYC Open Data*

1 Overview

This document provides supplementary materials accompanying the article *Principles for Open Data Curation: A Case Study with the New York City 311 Service Request Data*. It includes additional figures, tables, extended methodological details, selected console output, and reproducibility information that support—but are not included in—the main document. The supplementary materials are intended to improve transparency, facilitate reproducibility, and provide additional technical context for readers interested in the data-cleaning and validation processes applied in this study.

The below examples represent typical charts and tables created by the R programs. A full listing of the tables produced is available in the *reference* named text files stored on GitHub. The charts shown below are representative of the types of anomalies discovered and the associated Pareto and box plot visualizations. A full listing of the tables produced is available in the reference console output files stored on GitHub. The charts shown are representative of the types of anomalies discovered and the associated Pareto, bar, violin, and box plot visualizations.

- Data preparation:
  https://github.com/tusseyd/nyc_311_data_cleaning/blob/main/reference_console_output/reference_JDS_data_prep_console_output.txt
- Data cleaning:
  https://github.com/tusseyd/nyc_311_data_cleaning/blob/main/reference_console_output/reference_JDS_datacleaning_console_output.txt

2 Anomaly Detection: Visualizations, Statistics, and Analysis

This section presents illustrative examples of the visualizations and analytical approaches employed throughout the data cleansing program. While the full analysis generates dozens of charts across multiple data quality dimensions, the examples below demonstrate both the visualizations and the interpretive framework applied to identify and characterize data quality issues.

The analysis relies heavily on Pareto charts that stratify quality metrics and anomalies—such as invalid `community_board` values or suspicious durations—by NYC agency. As such, these charts serve a diagnostic purpose: they identify whether an anomaly concentrates within a few agencies or a single agency, suggesting an agency-specific data entry error or system configuration issue and thus affording targeted correction. Second, they reveal whether anomalies

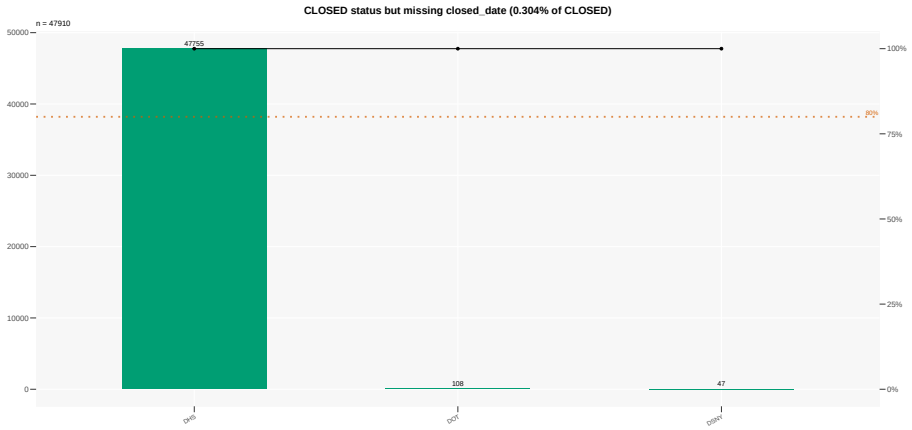

Figure S1: Pareto chart of SRs with a status of CLOSED, but missing a `closed_date`, by Agency.

distribute proportionally across Agencies mirroring the overall 311 Service Request (SR) volume distribution, indicating a systemic problem affecting the entire dataset rather than isolated to particular agencies.

Beyond Pareto charts, the program generates complementary visualizations including box plots (revealing distributional characteristics and outliers), violin plots (showing probability density across agencies), and histograms (exposing temporal or magnitude patterns). Each chart type supports specific analytical questions about data quality. The examples below illustrate how these chart types combine to enable comprehensive data quality assessment and inform targeted remediation strategies.

2.1 Service Request Status (SR) and Closure Analysis

Figure S1, shows a classic example of identifying the source agency associated with an anomaly. In this case, the anomaly is an SR that has a `status` of *closed*, but is missing the `closed_date` entry. The Pareto chart reveals that this is a problem located almost exclusively within the NYC Department of Homeless Services (DHS), where 99.7% such anomalies occur.

Conversely, Figure S2 displays the anomaly of having a `closed_date`, but *not* having a `status` of CLOSED. This anomaly is shown to primarily exist within the Department of Transportation (DOT) representing a full 77%, with smaller, yet significant, anomalies occurring within the Department of Buildings (DOB) and the Department of Sanitation (DSNY).

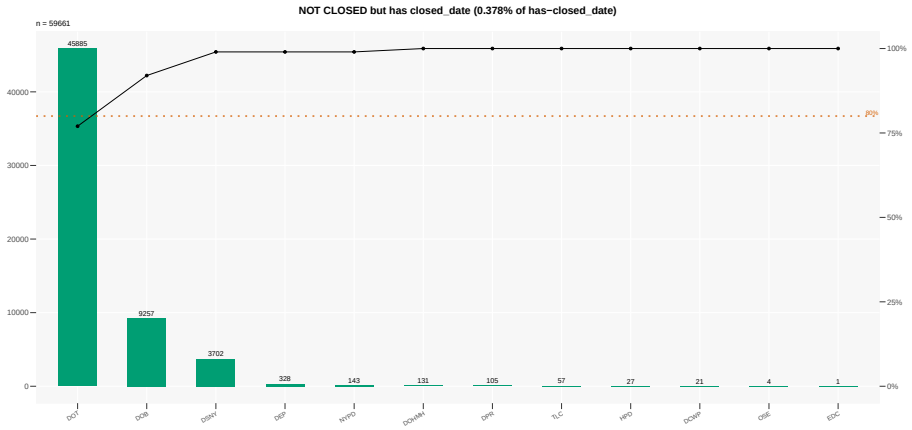

Figure S2: Pareto chart of SRs missing a CLOSED status, but with a `closed_date`, by Agency.

2.2 Temporal Pattern Analysis

The charts can also reveal trends in an anomaly. As noted in the reference document, there is a pronounced spike in the number of SRs closed *exactly* at midnight (00:00:00). This spike potentially indicates an automated bulk-action software program that appears to close selected SRs exactly at midnight. Note, however, that as shown in Figure S3 the yearly distribution of midnight `closed_date`(s) appears to be decreasing in the more current years.

Supplementing that discovery is Figure S4, a Pareto chart of the midnight closing anomaly by agency. The chart shows that the majority of the SRs closed-at-midnight occur within just two Agencies, DSNY and DOB, which collectively are responsible for 97.6% of such anomalies. Such information reveals a point-of-action for further investigation.

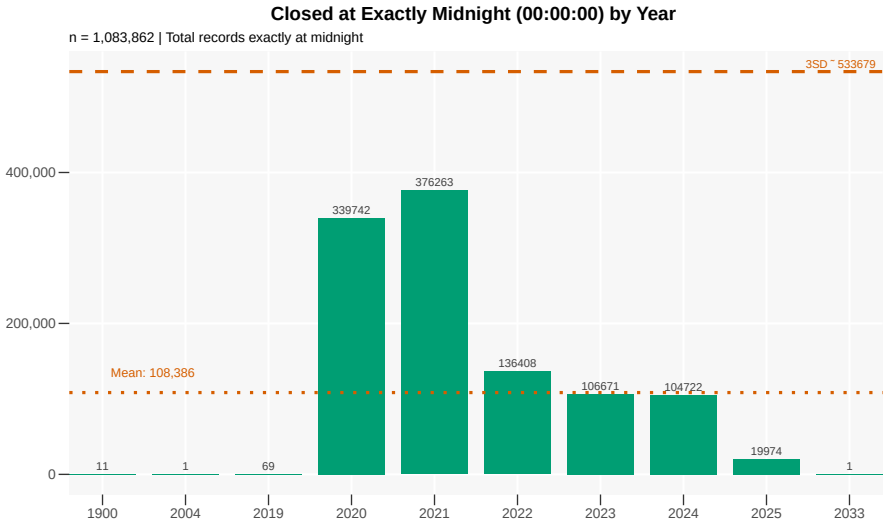

Figure S3: Yearly distribution of SRs with a midnight `closed_date`.

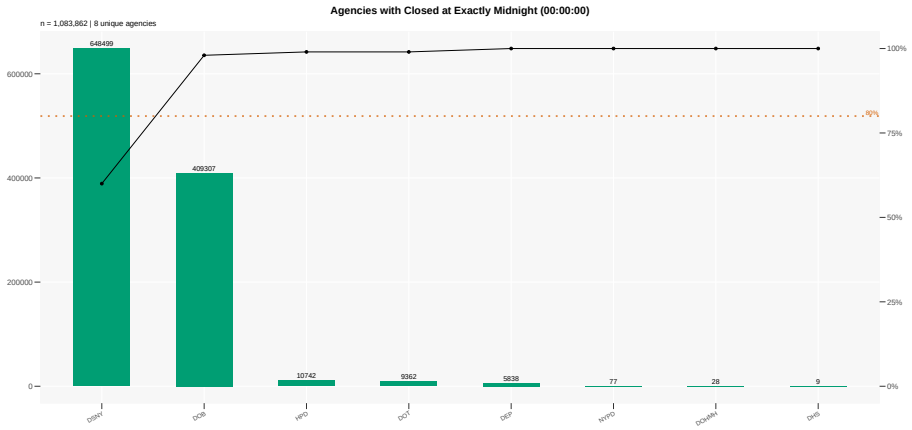

Figure S4: Pareto chart of SRs with a midnight `closed_date`, by Agency.

2.3 Service Request (SR) Positive Duration Analysis

One anomaly discussed in the referenced document is the presence of unusually long time spans between an SR being opened and when it is closed, referred to as the duration or response time. Such durations are frequently used to measure the performance of an agency's customer service. The following is an extended analysis of this long duration phenomenon that reveals not only the value of various visualizations, but also demonstrates the methodological approach to an underlying issue—a textbook study of Simpson's paradox where aggregate data distorts or conceals the underlying details.

In Figure S5 the box plot shows durations calculated as `closed_date` − `created_date`. The anomaly measured is SRs that have large positive durations, for this box plot between two and five years (730–1825 days) — duration limits that seem reasonable to count as *large*. This chosen range also excludes 1240 extreme durations, many of which appear to be caused by data entry errors, such as a `created_date` of 1900-01-01. There are 139,628 such *large* durations SRs. As shown in Figure S5, the box plot shows that the Department of Parks and Recreation (DPR) is responsible for most of these large durations, followed closely by the Department of Health and Mental Hygiene (DOHMH), the Economic Development Council (EDC) and DSNY.

Not only do these four Agencies contribute significantly to the long-duration SR metric, but the statistical mean of such durations, and the spread are also quite high. Meanwhile, as Figure S5 reveals, the New York Police Department (NYPD) has a large number of large duration SRs, but relatively small such durations limited in magnitude with the mean duration being just over the two year lower limit at 771 days and a tight σ of 79. Other agency's σ values are 2-4X that magnitude, indicating a broader spread. All agency values can be obtained by observing the associated box plot chart, supplemented by two tables produced in the console output `.tex` file.

Further insight emerges from analyzing *all* SRs with positive durations, regardless of magnitude— 15.3 million observations comprising 94% of the dataset. Figure S6 presents a histogram of these durations on a logarithmic scale, revealing a bimodal distribution. Hartigan's dip test confirms this bimodality is statistically significant ($D = 0.027$, $p < 2.2 \times 10^{-16}$), a near-zero p value that decisively rejects the null hypothesis of unimodality. Indeed, visual inspection of Figure **??** confirms that there is a distinct bimodal distribution, which would seem to require further analysis. Also of note is the separation of the median from the mean, indicating a skewed distribution to the right (Bowley skewness = 0.8079). Note also the large spread between the

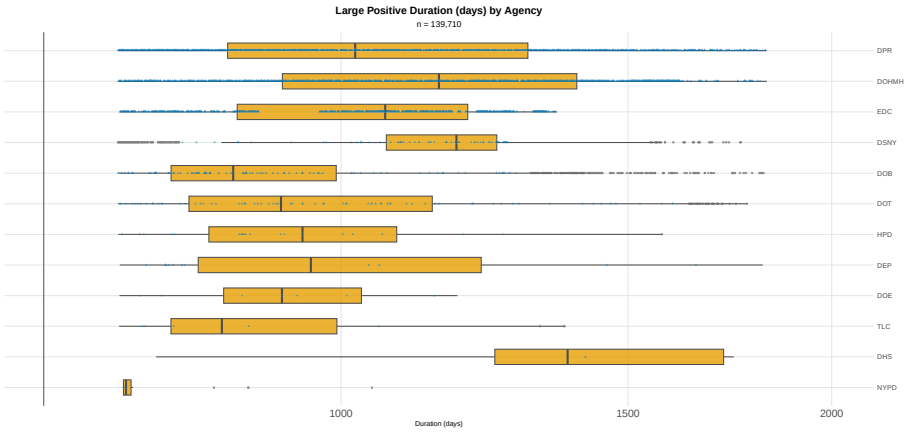

Figure S5: Distribution of SRs with Large (730–1,825 days) Durations, by Agency.

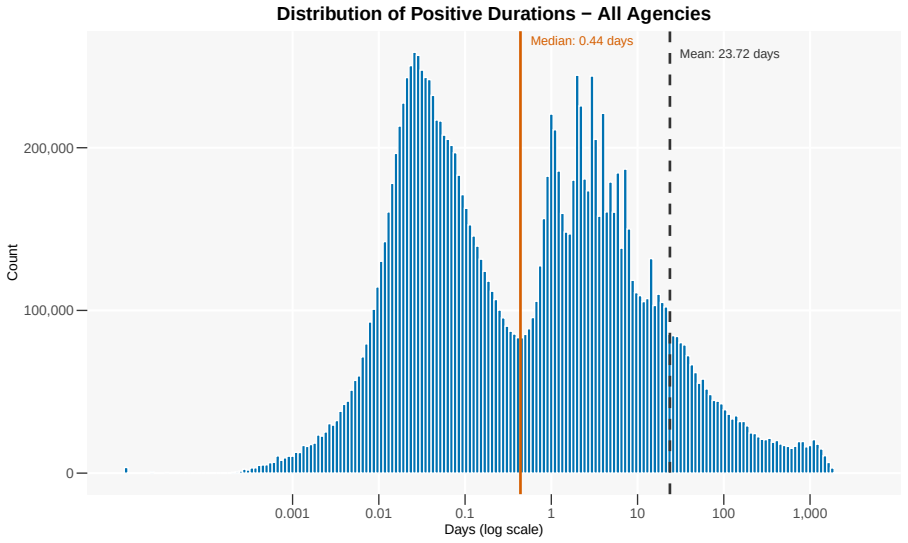

Figure S6: Overall Distribution of SRs with Positive Durations (log scale).

median (0.44 days) and the mean (23.72 days) further indicating the presence of outliers to the right.

In order to objectively identify the separation threshold between the two modes, the kernel density estimation method is employed. Using log-transformed durations the kernel density method located the local minimum between the two peaks. This valley-based approach yielded a threshold of 0.432 days (10.4 hours), which provides a criterion for classifying service requests into two different operational categories: *rapid-resolution* (< 0.432 days duration) and *standard* work orders ($\geqslant 0.432$ days duration).

Table T1 presents summary statistics for each operational mode, standard and rapid-resolution. Based on the separation threshold of 0.432 days, the rapid-resolution mode comprises 49.9% of requests (n = 7.66M) with a median duration of 0.87 hours and mean of 1.72 hours. The standard mode accounts for 50.1% of requests (n = 7.69M) with a median of 101.1 hours and mean of 1135 hours days. This near-perfect 50/50 split masks dramatically different operational models

Table T1: Summary Statistics by Operational Mode

Mode	N	%	Median (days)	Mean (days)	95th %ile (days)
Rapid-resolution	7,663,869	49.9	0.036	0.072	0.28
Standard	7,689,398	50.1	4.2	47.3	211.0
Total	15,353,267	100.0	51.6	569.0	—

Table T2: Distribution of Service Requests Across Operational Modes by Agency

	Rapid-resolution		Standard		
Agency	N	%	N	%	Total N
NYPD	6,685,153	96.8	220,228	3.2	6,905,381
DSNY	134,259	7.1	1,764,460	92.9	1,898,719
HPD	119,439	3.9	2,934,707	96.1	3,054,146
DOT	179,113	19.7	728,215	80.3	907,328
DEP	269,193	34.8	504,885	65.2	774,078
DPR	56,294	9.9	511,518	90.1	567,812
DOB	36,295	8.9	372,975	91.1	409,270
DOHMH	28,816	12.5	201,532	87.5	230,348
DHS	104,170	64.1	58,468	35.9	162,638
EDC	3,968	3.0	126,227	97.0	130,195
DCWP	2,373	2.0	113,744	98.0	116,117
TLC	10,119	8.6	106,997	91.4	117,116
All agencies	7,663,281	49.9	7,689,986	50.1	15,353,267

across city agencies—a clear example of Simpson's Paradox, where aggregate statistics obscure the true underlying structure when different, distinct populations are inappropriately combined.

Table T2 provides a further breakout of durations by agency. It reveals that NYPD dominates the rapid-resolution mode, accounting for 87% of all fast closures (6.69M out of 7.66M requests). Critically, 96.8% of NYPD's SRs close within 10.2 hours (0.43 days) — based on the bimodal threshold of 0.432 days — reflecting rapid responses to quality-of-life complaints. In stark contrast, infrastructure and regulatory agencies operate primarily in standard mode: Housing Preservation and Development (HPD) (96.1% standard), DSNY (92.9% standard), DOB (91.1% standard), and DPR (90.1% standard). It seems reasonable to assume that the nature of the SRs for these agencies require more physical service delivery, construction permits, code enforcement, or ongoing case investigation, thus potentially creating longer SR durations.

Identifying this bimodal structure has significant implications for data quality assessment and performance measurement. It means that the overall aggregate mean duration of 23.7 days is a misleading summary statistic—it falls in between the two operational modes and represents neither reality. Performance metrics, anomaly detection thresholds, and quality benchmarks must account for this operational difference to avoid spurious findings and invalid cross-agency comparisons.

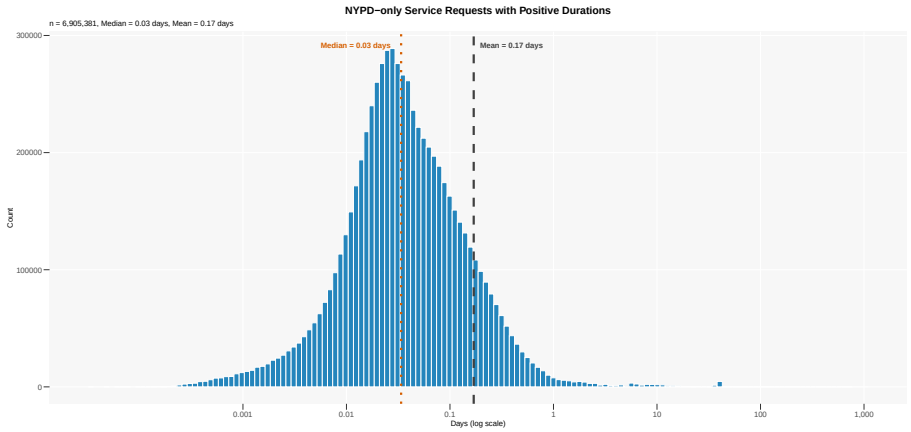

Figure S7: NYPD-Only Distribution of SRs with Positive Durations (log scale).

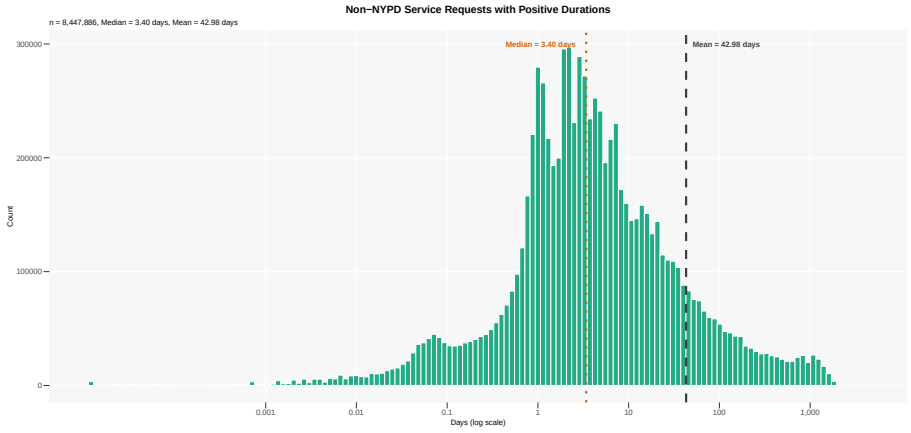

Figure S8: All non-NYPD Agencies Distribution of SRs with Positive Durations (log scale).

Three key conclusions emerge from this analysis: (1) the overall SR duration distribution is inaccurate and conceals two distinctly different operational modes—rapid-response and standard work-order processing; (2) NYPD is responsible for the vast majority of rapid-response durations; and (3) all other agencies combined constitute the vast majority of standard durations. A deeper can help discover the underling, distinct distributions. Visualizing how the durations of these two operational modes requires separating the durations of the NYPD versus all other Agencies. Figure S7 show NYPD-only positive distributions. Of note is that the median and mean lines are separated, indicating a right skewness possibly driven by a small number of large durations which have an outsized effect on the mean statistic.

In contrast to the NYPD-only durations is the aggregation of all other NYC Agencies, as shown in Figure S8, reveals a broader range of durations, with an exceptional number of very large, almost extreme, durations in the >1000 days (note that the x-axis is a logarithmic scale).

A clearer picture emerges as the two distributions (NYPD vs. all other Agencies) are superimposed on the same chart. Figure‘S9 clearly shows the bimodal distribution, reflecting the distinctly different operational protocols for the two categories — rapid-response and standard. Keep in mind that the NYPD is the single largest user of the NYC 311 non-emergency service,

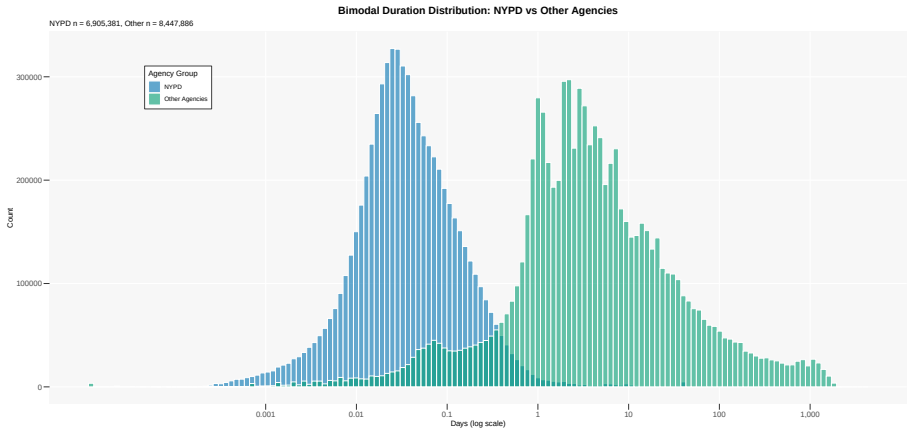

Figure S9: Combined Duration Profiles: NYPD and All Other Agencies (log scale).

representing 43% of all SRs. Thus, so it is perhaps not unusual to see that one agency have an outsized impact on the overall metric.

2.4 Service Request Negative Duration Analysis

A similar situation is illustrated in Figure S10 showing negative durations — where the `closed_date` occurs before the `created_date` resulting in a nonsensical negative duration. In Figure S10, the box plot shows DOT to be the largest offender. However, the mean negative duration is far greater for the Department of Environmental Protection (DEP) as shown on a logarithmic x-axis. This box plot visualization captures both the count and the extent of the negative duration anomaly.

Violin charts of these same distributions are also produced and available in the `charts` directory upon completion of the data cleansing program. Figure S11 displays a violin chart that accompanies the negative duration box plot, revealing the density of negative durations by magnitude. Every box plot has an accompanying violin chart.

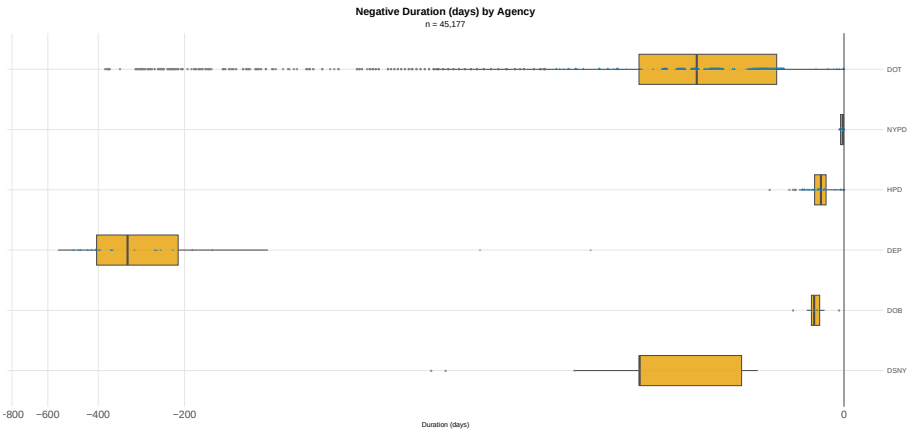

Figure S10: Distribution of Negative Durations (closed < created date), by Agency.

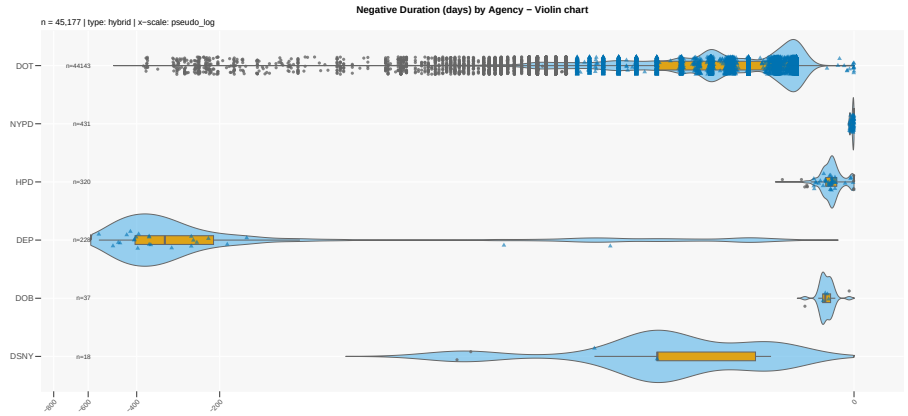

Figure S11: Combined Violin and Boxplot for SRs with Negative Durations, by Agency.

2.5 Zero and Near-Zero/Suspicious Duration Analysis

One challenging area was determining which duration values are impossible, improbable, or suspicious. Some duration categories are obvious, such as zero-second durations where the `created_date` and `closed_date` are identical, to the second. Indeed, there are 387,329 such impossible-duration SRs. Additionally, there are 3,460 SRs with a duration of exactly one second, a highly improbable if not impossible duration. Finally, there are SRs with quite small durations, e.g., just a few seconds, which are highly suspicious. These are classified as near-zero durations or suspicious durations.

The results of this analysis are shown, sorted by the statistical detection threshold method, in Figure S12 As noted in the referenced document, a quantitative analysis revealed a boundary of 28 seconds as an outlier based on the (LogNormal_3σ) method. Using this value reveals 14,741 suspicious durations. The referenced document also contains rationale for the use of this outlier methodology.

Figure S12 and Figure S13 visualize this suspicious boundary. These two charts show the overall distribution of near-zero duration SRs from 2—90 seconds; one showing cumulative and

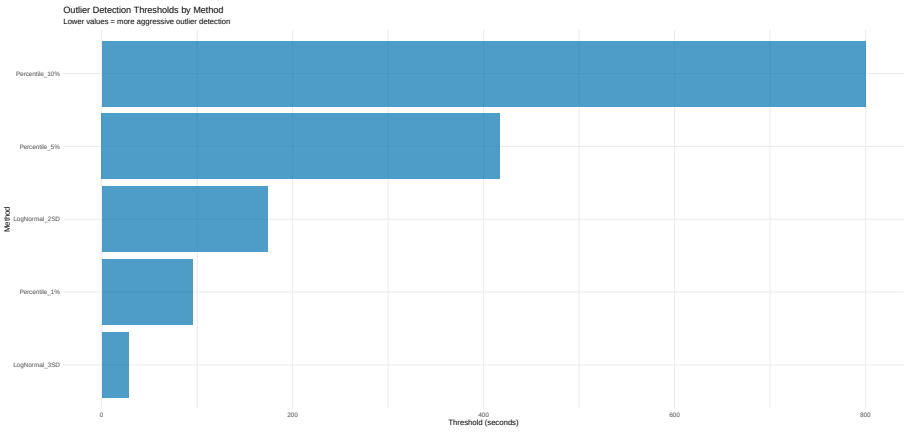

Figure S12: Outlier Detection Thresholds by Method. LogNormal 3SD Returned a Usable Result.

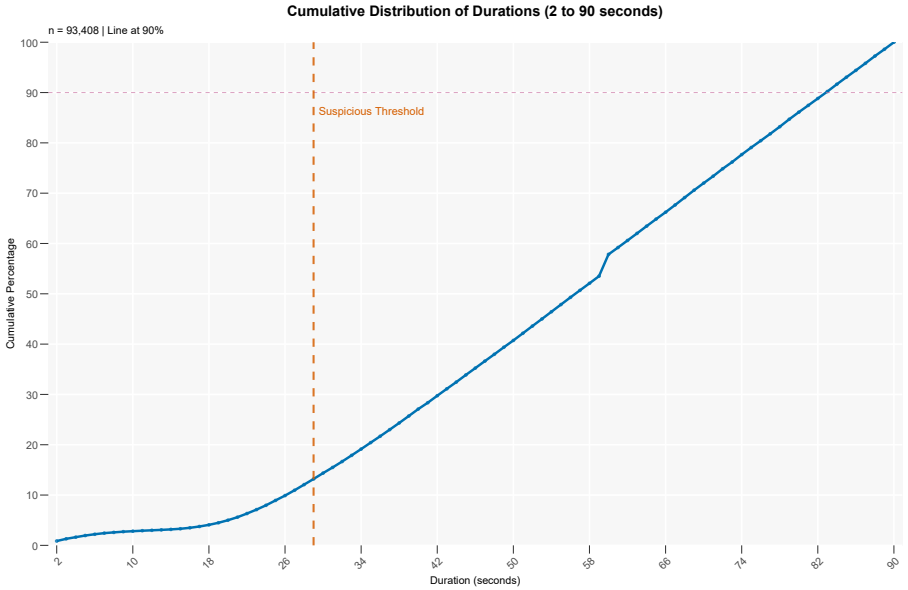

Figure S13: Cumulative Distribution of Small SR Durations (2–90 seconds).

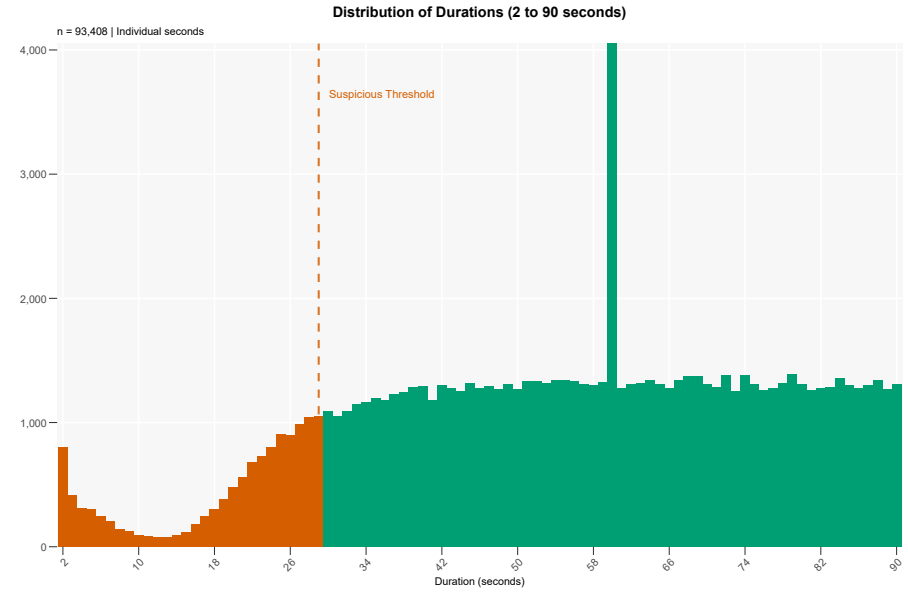

Figure S14: Distribution of Small SR Durations (2–90 seconds) by Count.

the other showing durations counts. The cumulative distribution shows the suspicious threshold of 28 seconds representing approximately 12% of SRs with durations inside that 2—90 time span.

2.6 Dataset Field usage by Agency

A complete field-usage-by-agency matrix is provided as a CSV file, rather than due printed to the console, due to its size and dimensionality. Each row in the file corresponds to a single data field, and each column corresponds to an NYC agency. The cell values indicate the count of records in

Figure S15: Backlog of SR — Open at the Beginning of the Year

which the given field is populated for the specified agency. The final column contains the total count across all agencies.

Field usage varies substantially by agency, reflecting differences in operational responsibilities, reporting practices, and data-collection requirements. Core administrative fields (i. e., `unique_key`, `created_date`, `status`. `agency`, and `complaint_type`) are fully populated across all agencies, whereas location-specific, infrastructure-related, and program-specific fields are populated only by agencies for which they are operationally relevant. The CSV file is available at: https://figshare.com/s/9f878b50687c9e4c540a

3 SR Backlog

An interesting discovery during this analysis was the volume of *backlogged* SRs. An SR is classified as backlogged if it is open at the end of the calendar year, and thus must be rolled over and (hopefully) closed during the following year. While the total number of SRs grew by 17.5% from 2020–2024, the backlog growth greatly exceeded that rate, increasing by over 200% in raw count. Figure S15 shows this dramatic growth with backlogged SRs representing 1.7% of all SRs entering 2024, as compared to 0.7% entering 2020.

4 Reproducibility and Code Access

System Requirements

- R version 4.5.2 or higher (https://cran.r-project.org/)
- RStudio version 2026.01.0 build 392 or higher (recommended): https://posit.co/download/rstudio-desktop/
- Minimum RAM: 16 GB (tested and verified; 32 GB provides faster execution)
- Disk space: ∼15 GB (∼9 GB uncompressed data) + intermediate files + outputs)
- Operating System: Tested on Windows 10/11; macOS/Linux users will need to modify file paths in `base_dir`

Data and Code Repository

- Source code: https://github.com/tusseyd/nyc_311_data_cleaning
- NYC 311 dataset ((∼1.6 GB zipped, (∼9 GB uncompressed): https://figshare.com/s/e8d479c391edb7224bfa
- USPS Zip Codes (4.5 MB): https://figshare.com/s/8aea027d06f4903f2227
- Reference console output for validation is available in GitHub repository.

Required R Packages

Both scripts automatically check for and install missing packages. The two R programs require the following R packages:

- data.table
- arrow
- fasttime
- here
- zoo
- ggpmisc
- ggpattern
- ggrastr
- qcc
- qicharts2
- grid
- gridExtra
- sf
- stringdist
- tidyverse
- bslib
- shiny
- DT
- gt
- styler
- rlang
- renv
- remotes
- moments
- diptest

Analysis Reproduction Overview

As noted in the *Principles for Open Data Curation: A Case Study with the New York City 311 Service Request Data* document there are two separate R programs which should be executed in order:

1. `data_prep_for_jds_datacleansing.R` — Prepares the raw data for analysis
2. `jds_datacleansing.R` — Performs data cleansing and quality assessments

Each of these programs produces a console output file in simple text format, specifically:

- `JDS_data_prep_console_output.txt`
- `JDS_datacleaning_console_output.txt`

These two files, along with the associated charts provide additional insight into the anomalies observed during data cleansing. Upon running the `jds_datacleansing.R` program, there will be 92 charts in .PDF format in the `charts` directory and a `field_usage_summary_table.csv` file in the `analytics` directory.

For validation purposes, a reference copy of the two console output files is available on GitHub. These reference files can also be used to view the full console output of the two programs, without needing to run the .R programs. Unfortunately, the full set of 92 charts can only be duplicated by running the `jds_datacleansing.R` program, as it is not possible to include all charts in either the main document nor in this supplementary document.

Steps for Reproducing the Analysis

Source files are available at:

- R source code (GitHub): https://github.com/tusseyd/nyc_311_data_cleaning
- USPS Zip Code dataset (Figshare): https://figshare.com/articles/dataset/Zip_Code_file_for_the_Journal_of_Data_Science_project/31053361?file=60974260

- NYC 311 dataset (Figshare): https://figshare.com/articles/dataset/Data_set_in_ZIP_format_for_the_311_nyc_data_cleaning_project_for_the_Journal_of_Data_Science/30699800?file=59813192
- Data preparation console output: https://github.com/tusseyd/nyc_311_data_cleaning/blob/main/reference_console_output/reference_JDS_data_prep_console_output.txt
- Data cleaning console output: https://github.com/tusseyd/nyc_311_data_cleaning/blob/main/reference_console_output/reference_JDS_datacleaning_console_output.txt

Step 1: Download the two data Files

1. Download the NYC 311 and USPS Zip Code datasets from Figshare. (links above)
2. Do not rename these files as the scripts expect original filenames:
   - `5-year_311SR_01-01-2020_thru_12-31-2024_AS_OF_10-10-2025.zip`
   - `zip_code_database.csv`
3. Unzip the zip file (~9 GB uncompressed) to create a .CSV file. This .CSV file should be placed into the raw_data directory. (See directory structure below.)
4. The zip_code.csv file should also be placed into the raw_data directory. Both of these .CSV files will be modified by the data prep program to the .RDS format for ease in processing.

Step 2: Set Up Project Structure

1. Clone or download the entire GitHub repository: https://github.com/tusseyd/nyc_311_data_cleaning
2. Important: Ensure you download all files from the GitHub repo including:
   - `code/data_prep_for_jds_datacleansing.R`
   - `code/jds_datacleansing.R`
   - All files in `code/functions/` directory (required dependencies)
3. Running the data preparation program `data_prep_for_jds_datacleansing.R` will automatically create the following directory structure. The individual files shown below will be created by running the data cleansing program. Below is the directory structure and the files they will contain after running the data cleansing program `code/jds_datacleansing.R`.

```
/ (User's R Working Directory)
└── nyc_311_data_cleaning/ (Project Root)
    ├── analytics/
    │   └── field_usage_summary_table.csv
    ├── charts/
    │   └── [92 PDF files]
    ├── code/
    │   ├── data_prep_for_jds_datacleansing.R
    │   ├── jds_datacleansing.R
    │   └── functions/
    │       └── [function files]
    ├── console_output/
    │   ├── JDS_data_prep_console_output.txt
    │   └── JDS_datacleansing_console_output.txt
    └── data/
```

```
    ├── 5-year_311SR_01-01-2020_thru_12-31-2024_AS_OF_10-10-2025.rds
    ├── USPS_zipcodes.rds
    └── raw_data/
        ├── 5-year_311SR_01-01-2020_thru_12-31-2024_AS_OF_10-10-2025.csv
        └── zip_code_database.csv
└── reference_outputs/
    ├── JDS_data_prep_console_output.txt
    └── JDS_datacleansing_console_output.txt
```

Step 3: Open RStudio and open `code/data_prep_for_jds_datacleansing.R` Modify the `base_dir` variable (line 31) to match your local path:

```
base_dir <- file.path("your", "path", "here", "nyc_311_data_cleaning")
```

Note: Non-Windows users should use forward slashes or platform-appropriate path separators

Step 4: Place the downloaded data files in the following directories:

- `5-year_311SR_01-01-2020_thru_12-31-2024_AS_OF_10-10-2025.csv` → `data/raw_data/`
- `zip_code_database.csv` → `data/raw_data/`

Step 5: Run the Data Preparation R program

1. Execute `data_prep_for_jds_datacleansing.R` in RStudio
2. The script will create the directory structure (if missing) and install required R packages.
3. Expected runtime: Approximately 90 minutes on a system with 16 GB RAM; (∼30 min on a system with 32 GB RAM *Note: Initial data loading is memory-intensive; the script optimizes memory usage after initial processing*
4. Upon completion, verify these files exist:
   - `data/5-year_311SR_01-01-2020_thru_12-31-2024_AS_OF_10-10-2025.rds`
   - `data/USPS_zipcodes.rds`
   - `console_output/JDS_data_prep_console_output.txt`

Step 6: Run the Data Cleansing Analysis R program

1. Execute `code/jds_datacleansing.R` in RStudio
2. Expected runtime: Approximately 60 minutes on a system with 16 GB RAM; ∼31 minutes on a system with 32 GB.
3. Monitor progress in RStudio's plot pane as charts are generated.
4. The script will produce:
   - 92 PDF charts in `charts/`
   - `field_usage_summary_table.csv` in `data/analytical_files/`
   - `console_output/JDS_datacleaning_console_output.txt`

Validation

Compare your directory structure along with the associated files shown in the above tree structure. Compare the charts to those in the original document and in this Supplemental document. Compare your console output files against the reference outputs provided in the GitHub repository to verify successful reproduction. The console outputs contain many tables of information which should match your run. Validation console output files are located at:

- Data preparation console output: https://github.com/tusseyd/nyc_311_data_cleaning/blob/main/reference_console_output/reference_JDS_data_prep_console_output.txt

- Data cleaning console output: https://github.com/tusseyd/nyc_311_data_cleaning/blob/main/reference_console_output/reference_JDS_datacleaning_console_output.txt

Notes

- Total processing time (data preparation and cleansing): Approximately 2.5 hours on a system with 16 GB RAM; ∼2X faster on systems with 32 GB.
- The scripts use `data.table` for memory-efficient processing of 15+ million records
- All file paths are configured relative to `base_dir` for portability
- Memory management: Initial data loading requires substantial RAM; the script reduces memory footprint after preprocessing.