Assignment 2 - Assess, Filter, Clean, & Present Data

Juan Maldonado Franco

February 02, 2026

Contents

```r
# Install only if needed:
# install.packages(c("readr","readxl","dplyr","tidyr","janitor","knitr","kableExtra","stringr"

library(readr)
library(readxl)
library(dplyr)
library(tidyr)
library(janitor)
library(knitr)
library(kableExtra)
library(stringr)
library(Amelia)
library(lubridate)
```

Executive Summary

This report continues the Exploratory Data Analysis (EDA) process on the DOHMH New York City Restaurant Inspection Results dataset. The objective of this work is to assess missingness, remove observations and variables that exceed a missingness threshold, apply defensible imputation for remaining gaps, and verify that the resulting analytical dataset is complete. These actions are foundational in EDA because missing values can distort descriptive statistics, bias model estimates, and produce misleading visual conclusions if they are left unexamined or handled inconsistently (Little & Rubin, 2019).

The analysis proceeded by loading and typing variables, standardizing categorical representations so that blank strings were treated as missing, visualizing missingness to evaluate its structure, and quantifying row level and column level completeness under a twenty percent threshold rule. High missingness variables were removed first, followed by high missingness observations. This ordering is methodologically defensible because removing variables can reduce the proportion of missingness per row, which preserves more usable observations. Remaining missing values were imputed using a type aware approach consistent with the permitted methods.

Numeric variables were imputed using the median because a distribution check of the inspection score indicated skewness and the presence of extreme values. Categorical variables were imputed using the mode because it preserves valid category membership and avoids nonsensical numeric substitution. Date variables were imputed using the median date after conversion to a numeric time representation, then returned to Date form. A second missingness map and a direct missing value count confirmed that the final dataset contains no missing values.

The discussion section explains how data cleaning affects the validity of exploratory conclusions and downstream decision making, and it compares deletion, mean imputation, median imputation, and mode imputation with attention to missingness mechanisms and interpretability (Little & Rubin, 2019). Decisions related to outliers and extreme observations are documented to preserve analytical transparency and substantive interpretability. Overall, the result is an analysis ready dataset suitable for subsequent descriptive analysis and modeling, with each cleaning and imputation decision documented and justified.

Introduction

Exploratory Data Analysis is a disciplined approach for maximizing insight into a dataset by uncovering structure, detecting anomalies, and testing assumptions through a sequence of iterative checks (Princeton Univ NJ Dept of Statistics & Tukey, 1993). A core principle of EDA is that data quality constraints determine the boundary of valid inference, meaning that analytical conclusions are only as defensible as the structure and completeness of the underlying data (National Institute of Standards and Technology, 2012). Missing values are one of the most common threats to that boundary because missingness can change distributions, attenuate relationships, and create artifacts that appear meaningful but are actually the result of incomplete measurement.

This analysis operationalizes missing data handling as a structured workflow. The process begins by visualizing missingness to understand its shape and concentration. It then applies a clear threshold rule to remove observations and variables that are too incomplete to support defensible analysis. Finally, it imputes the remaining gaps using simple, transparent methods and verifies that the final dataset is complete. This sequencing reflects standard EDA practice because visualization and threshold based filtering reduce the risk of imputing in situations where the data are too sparse to support stable replacement values.

This dataset has been used in prior public health research to evaluate sanitation outcomes and behavioral responses to restaurant grading systems, making it well suited for exploratory and descriptive analysis in a regulatory context (Wong et al., 2015).

Analytical Environment and Dataset Acquisition

3.1 Files Used for This Analysis

This analysis uses the same dataset as the prior week.

- DOHMH_New_York_City_Restaurant_Inspection_Results_20260127.csv

The data dictionary is not required to execute the missingness workflow. However, it remains useful as optional documentation for distinguishing identifiers from true quantitative measures. The conditional download block is retained for reproducibility.

3.2 Conditional File Acquisition

```
# File names (local)
data_file <- "DOHMH_New_York_City_Restaurant_Inspection_Results_20260127.csv"
dict_file <- "RestaurantInspectionDataDictionary_09242018.xlsx"

# Source URLs (NYC Open Data)
data_url <- "https://data.cityofnewyork.us/api/views/43nn-pn8j/rows.csv?accessType=DOWNLOAD"
dict_url <- "https://data.cityofnewyork.us/api/views/43nn-pn8j/files/RestaurantInspectionDataD

# Conditional download: inspection results
if (!file.exists(data_file)) {
  message("Inspection results file not found.  Downloading...")
  download.file(url = data_url, destfile = data_file, mode = "wb")
} else {
  message("Inspection results file found locally.")
```

```
}

# Optional: conditional download of data dictionary (retained for portability)
if (!file.exists(dict_file)) {
  message("Data dictionary file not found.  Downloading...")
  download.file(url = dict_url, destfile = dict_file, mode = "wb")
} else {
  message("Data dictionary file found locally.")
}
```

3.3 Load the Dataset

```
df_raw <- read_csv(file = data_file, show_col_types = FALSE, progress = FALSE) %>%
  clean_names()

df <- df_raw

dim(df)
```

```
## [1] 296854     27
```

Missing Data Visualization

Before applying thresholds or imputation, missingness must be made visible. In addition to explicit NA values, this dataset may contain blank strings in categorical fields. Blank strings are not treated as missing by default, so they must be standardized to NA prior to visualization and computation.

4.1 Type Alignment for Safe Imputation

To avoid invalid imputations, administrative identifiers and categorical variables are explicitly typed as factors. This prevents accidental numeric imputation on identifiers such as CAMIS or ZIPCODE.

```
df_typed <- df %>%
  mutate(
    camis = as.factor(camis),
    zipcode = as.factor(zipcode),
    phone = as.factor(phone),
    bin = as.factor(bin),
    bbl = as.factor(bbl),
    community_board = as.factor(community_board),
    council_district = as.factor(council_district),
    census_tract = as.factor(census_tract),
    boro = as.factor(boro),
    critical_flag = as.factor(critical_flag),
    action = as.factor(action),
    violation_code = as.factor(violation_code),
    violation_description = as.factor(violation_description),
    cuisine_description = as.factor(cuisine_description),
```

```
    inspection_type = as.factor(inspection_type),
    nta = as.factor(nta),
    grade = factor(grade, levels = c("A","B","C","P","Z"), ordered = TRUE),
    inspection_date = as.Date(inspection_date, format = "%m/%d/%Y"),
    grade_date = as.Date(grade_date, format = "%m/%d/%Y"),
    record_date = as.Date(record_date, format = "%m/%d/%Y")
  )

str(df_typed[c("camis","boro","score","grade","inspection_date")])
```

```
## tibble [296,854 x 5] (S3: tbl_df/tbl/data.frame)
##  $ camis          : Factor w/ 30702 levels "30075445","30191841",..: 22098 14636 18462 4669
##  $ boro           : Factor w/ 6 levels "0","Bronx","Brooklyn",..: 3 3 5 5 5 5 6 3 3 4 ...
##  $ score          : num [1:296854] 22 46 53 11 9 34 NA 10 31 26 ...
##  $ grade          : Ord.factor w/ 5 levels "A"<"B"<"C"<"P"<..: NA NA 3 1 1 NA NA 1 NA NA ..
##  $ inspection_date: Date[1:296854], format: "2024-03-21" "2025-12-23" ...
```

4.2 Standardize Blank Categories to Missing

```
blank_to_na_chr <- function(x) {
  x <- as.character(x)
  x <- stringr::str_trim(x)
  x[x == ""] <- NA_character_
  x
}

# Document the "null level" issue before conversion.
blank_counts_before <- df_typed %>%
  summarise(across(where(is.factor), ~ sum(stringr::str_trim(as.character(.)) == ""))) %>%
  pivot_longer(cols = everything(), names_to = "factor_variable", values_to = "blank_count") %
  arrange(desc(blank_count))

kable(
  head(blank_counts_before, 15),
  caption = "Top factor variables containing blank strings before standardization."
) %>%
  kable_styling(full_width = FALSE)
```

Table 1: Top factor variables containing blank strings before standardization.

factor_variable	blank_count
camis	0
boro	0
critical_flag	0

zipcode	NA
phone	NA
cuisine_description	NA
action	NA
violation_code	NA
violation_description	NA
grade	NA
inspection_type	NA
community_board	NA
council_district	NA
census_tract	NA
bin	NA

```
df_step1 <- df_typed %>%
  mutate(
    across(where(is.character), blank_to_na_chr),
    across(where(is.factor), ~ {
      y <- blank_to_na_chr(.)
      factor(y)
    })
  )

blank_counts_after <- df_step1 %>%
  summarise(across(where(is.factor), ~ sum(stringr::str_trim(as.character(.)) == ""))) %>%
  pivot_longer(cols = everything(), names_to = "factor_variable", values_to = "blank_count") %>
  arrange(desc(blank_count))

kable(
  head(blank_counts_after, 15),
  caption = "Top factor variables containing blank strings after standardization."
) %>%
  kable_styling(full_width = FALSE)
```

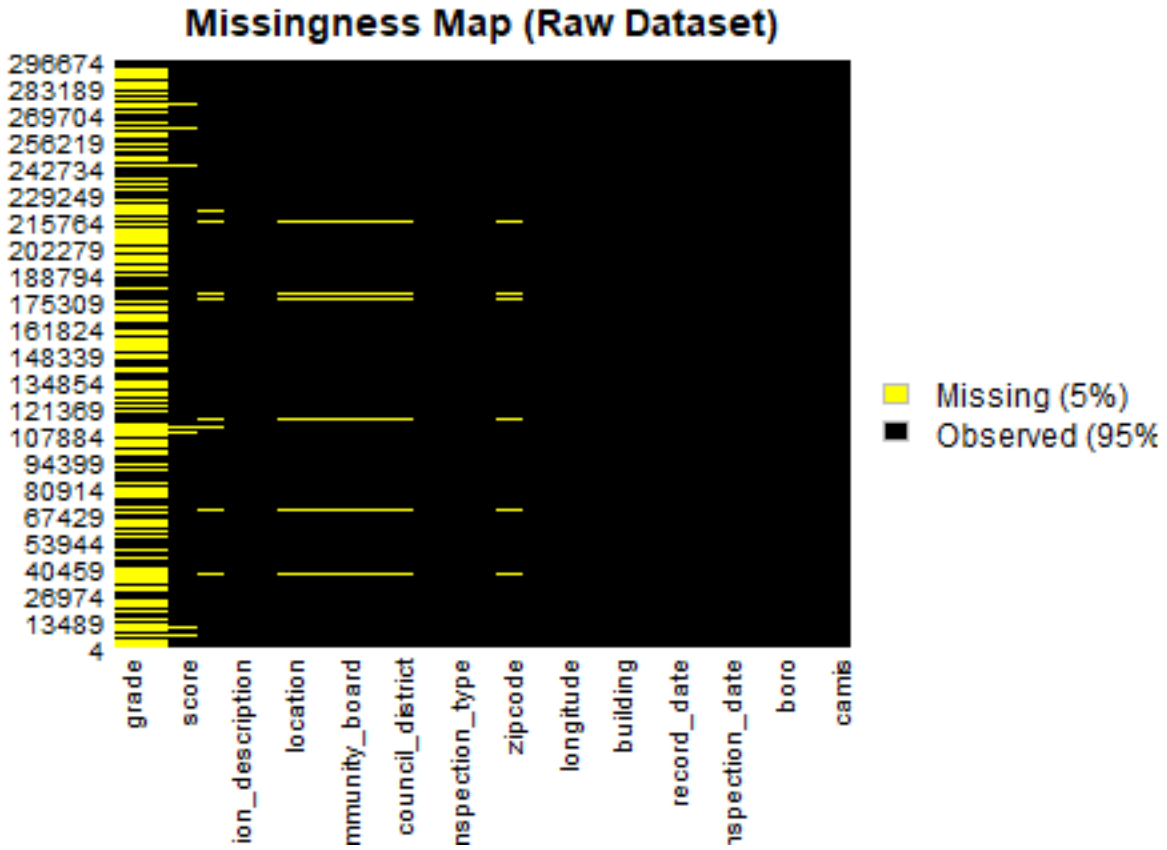
Table 2: Top factor variables containing blank strings after standardization.

factor_variable	blank_count
camis	0
boro	0
critical_flag	0
zipcode	NA
phone	NA
cuisine_description	NA
action	NA
violation_code	NA

violation_description	NA
grade	NA
inspection_type	NA
community_board	NA
council_district	NA
census_tract	NA
bin	NA

4.3 Amelia Missingness Map (Raw)

```
Amelia::missmap(
  as.data.frame(df_step1),
  main = "Missingness Map (Raw Dataset)",
  col = c("yellow", "black"),
  legend = TRUE
)
```



Evaluation of Row-Level and Column-Level Missingness

Rows and columns were assessed using a twenty percent missingness rule. This threshold provides a clear and repeatable criterion for removing observations and variables that are too incomplete to support defensible descriptive analysis or stable imputation (Little & Rubin, 2019).

```r
row_missing_pct <- rowMeans(is.na(df_step1))
col_missing_pct <- colMeans(is.na(df_step1))

rows_20_idx <- which(row_missing_pct >= 0.20)
cols_20_names <- names(col_missing_pct[col_missing_pct >= 0.20])

length(rows_20_idx)
```

```
## [1] 7547
```

```r
length(cols_20_names)
```

```
## [1] 2
```

```r
# Summarize missingness by column for reporting.
col_missing_summary <- data.frame(
  column_name = names(col_missing_pct),
  missing_pct = round(100 * col_missing_pct, 2),
  n_missing = colSums(is.na(df_step1)),
  stringsAsFactors = FALSE
) %>%
  arrange(desc(missing_pct))

kable(
  head(col_missing_summary, 15),
  caption = "Top 15 variables by percent missing before filtering."
) %>%
  kable_styling(full_width = FALSE)
```

Table 3: Top 15 variables by percent missing before filtering.

	column_name	missing_pct	n_missing
grade	grade	53.83	159810
grade_date	grade_date	53.83	159802
score	score	5.51	16364
bin	bin	1.94	5764
violation_code	violation_code	1.87	5547
violation_description	violation_description	1.87	5548
community_board	community_board	1.51	4472
nta	nta	1.51	4472
location	location	1.51	4472
council_district	council_district	1.50	4445
census_tract	census_tract	1.50	4445

cuisine_description	cuisine_description	1.10	3262
action	action	1.10	3254
inspection_type	inspection_type	1.10	3254
zipcode	zipcode	1.05	3126

```
# Explicitly list variables that exceed the threshold.
cols_over_20 <- col_missing_summary %>% filter(missing_pct >= 20)

kable(
  cols_over_20,
  caption = "Variables exceeding the 20 percent missingness threshold."
) %>%
  kable_styling(full_width = FALSE)
```

Table 4: Variables exceeding the 20 percent missingness threshold.

	column_name	missing_pct	n_missing
grade	grade	53.83	159810
grade_date	grade_date	53.83	159802

Threshold-Based Data Reduction

The dataset was filtered using the threshold rule. Columns at or above twenty percent missingness were removed first. Rows at or above twenty percent missingness were removed second. This ordering is important because dropping high missingness variables can reduce the missingness proportion for many rows, which reduces the risk of unstable imputation and preserves usable observations (Little & Rubin, 2019).

```
df_step3 <- df_step1

# Remove columns with >= 20% missing.
if (length(cols_20_names) > 0) {
  df_step3 <- df_step3 %>% select(-all_of(cols_20_names))
}

# Recompute row missingness after column removal.
row_missing_pct_step3 <- rowMeans(is.na(df_step3))
rows_20_idx_step3 <- which(row_missing_pct_step3 >= 0.20)

# Remove rows with >= 20% missing.
if (length(rows_20_idx_step3) > 0) {
  df_step3 <- df_step3 %>% slice(-rows_20_idx_step3)
}

dim(df_step3)
```

```
## [1] 289301     25
```

```
# Report what was removed.
removed_summary <- data.frame(
  item = c("Columns removed (>= 20% missing)", "Rows removed (>= 20% missing after column remov
  count = c(length(cols_20_names), length(rows_20_idx_step3)),
  stringsAsFactors = FALSE
)

kable(removed_summary, caption = "Threshold based removals applied to the dataset.") %>%
  kable_styling(full_width = FALSE)
```

Table 5: Threshold based removals applied to the dataset.

item	count
Columns removed (>= 20% missing)	2
Rows removed (>= 20% missing after column removal)	7553

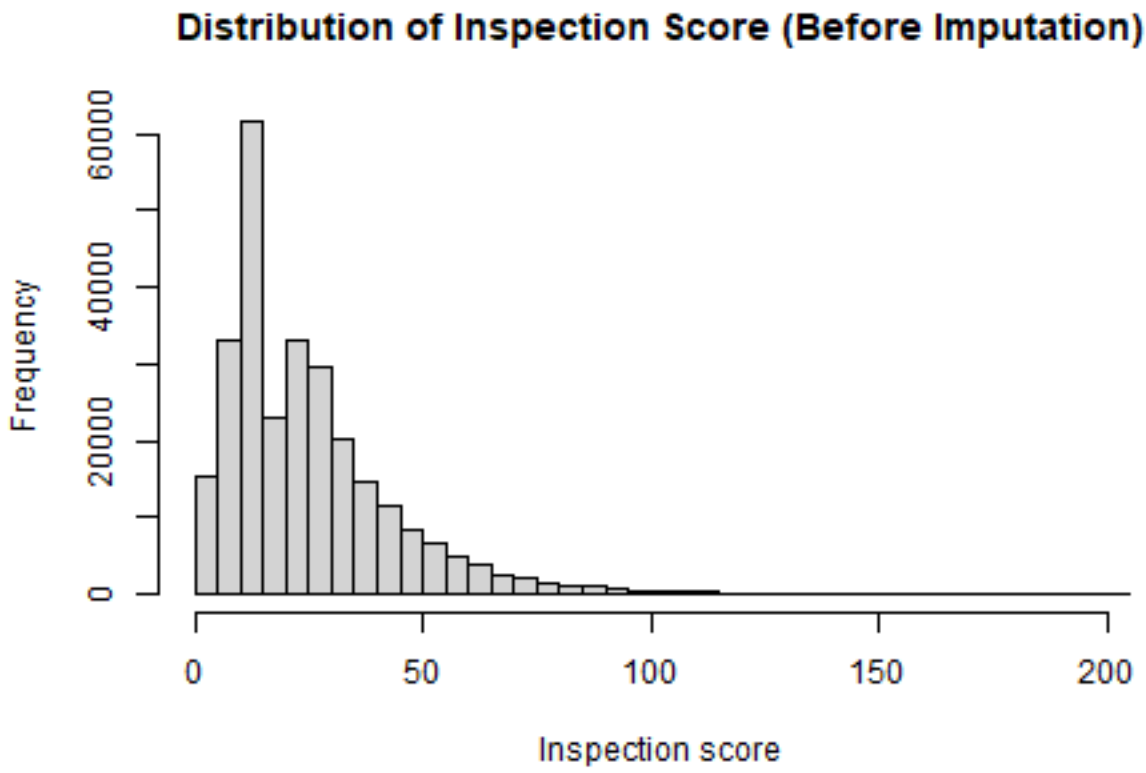
Imputation of Remaining Missing Values

After threshold removals, remaining missing values were imputed using simple, transparent methods. Numeric variables were imputed using the median. Median imputation is robust to skew and outliers, and it preserves a realistic central value when extreme values are present, particularly in administrative datasets where distributions are rarely symmetric (Little & Rubin, 2019). Categorical variables were imputed using the mode because it maintains valid category membership and avoids introducing invalid values. Date variables were imputed using the median date, which is defensible when missingness is limited and the goal is to preserve a realistic temporal center without introducing extreme dates.

6.1 Evidence Based Justification for Median Imputation

The median was selected for numeric imputation after verifying that key numeric outcomes are not perfectly symmetric. The inspection score is an administrative compliance metric where extreme values can occur. A histogram and a simple summary based skewness indicator provide direct evidence that median replacement is more robust than mean substitution in this context (Little & Rubin, 2019).

```
df_score_nonmissing <- df_step3 %>% filter(!is.na(score))

hist(
  df_score_nonmissing$score,
  main = "Distribution of Inspection Score (Before Imputation)",
  xlab = "Inspection score",
  breaks = 50
)
```




```
score_iqr <- IQR(df_score_nonmissing$score, na.rm = TRUE)
score_skew_indicator <- mean(df_score_nonmissing$score, na.rm = TRUE) - median(df_score_nonmiss

kable(
  data.frame(
    metric = c("Mean minus median (skew indicator)", "IQR"),
    value = c(round(score_skew_indicator, 4), round(score_iqr, 4))
  ),
  caption = "Diagnostics supporting median imputation for numeric variables."
) %>%
  kable_styling(full_width = FALSE)
```

Table 6: Diagnostics supporting median imputation for numeric variables.

metric	value
Mean minus median (skew indicator)	4.2318
IQR	21.0000

6.2 Helper Functions

```r
mode_value <- function(x) {
  x_no_na <- x[!is.na(x)]
  if (length(x_no_na) == 0) return(NA)
  ux <- unique(x_no_na)
  ux[which.max(tabulate(match(x_no_na, ux)))]
}

impute_median_numeric <- function(x) {
  x[is.na(x)] <- median(x, na.rm = TRUE)
  x
}

impute_median_date <- function(x) {
  xn <- as.numeric(x)
  xn[is.na(xn)] <- median(xn, na.rm = TRUE)
  as.Date(xn, origin = "1970-01-01")
}

impute_mode_factor <- function(x) {
  m <- mode_value(x)
  x[is.na(x)] <- m
  x
}
```

6.3 Apply Imputation

```r
df_step4 <- df_step3

# Identify variable groups.
num_cols <- names(df_step4)[sapply(df_step4, is.numeric)]
date_cols <- names(df_step4)[sapply(df_step4, inherits, what = "Date")]
factor_cols <- names(df_step4)[sapply(df_step4, is.factor)]

# Impute numeric columns with median.
if (length(num_cols) > 0) {
  df_step4 <- df_step4 %>%
    mutate(across(all_of(num_cols), impute_median_numeric))
}

# Impute Date columns with median date.
if (length(date_cols) > 0) {
  df_step4 <- df_step4 %>%
    mutate(across(all_of(date_cols), impute_median_date))
}

# Impute factor columns with mode.
if (length(factor_cols) > 0) {
```

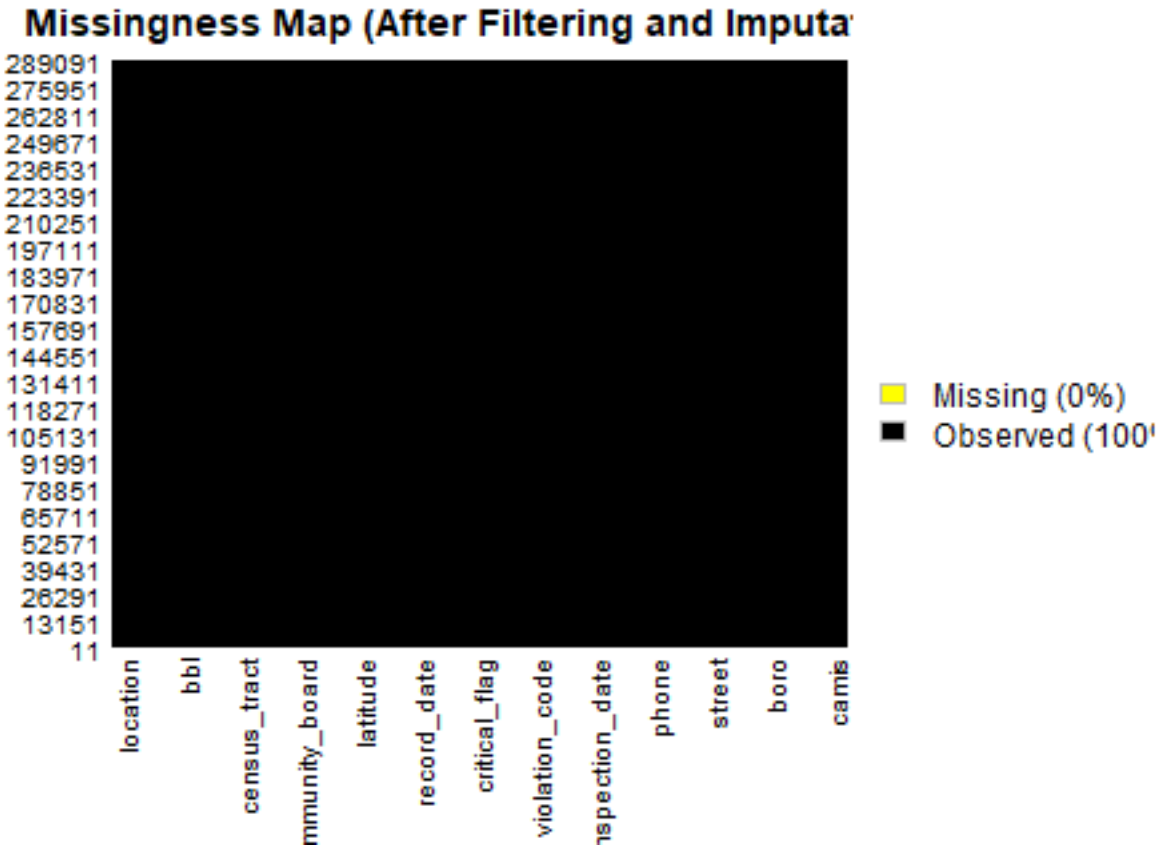
```r
  df_step4 <- df_step4 %>%
    mutate(across(all_of(factor_cols), impute_mode_factor))
}

# Final cleaned dataset.
df_final <- df_step4

sum(is.na(df_final))
```

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

Verification of Dataset Completeness

The dataset was re visualized using the same missingness map approach used earlier. The goal of this section is to verify that threshold filtering and imputation produced a complete dataset, and that no missingness remains hidden in categorical variables or derived date fields (Little & Rubin, 2019).

```r
Amelia::missmap(
  as.data.frame(df_final),
  main = "Missingness Map (After Filtering and Imputation)",
  col = c("yellow", "black"),
  legend = TRUE
)
```




```
# Confirm completeness.
total_missing <- sum(is.na(df_final))
kable(
  data.frame(metric = "Total missing values in final dataset", value = total_missing),
  caption = "Verification of missingness after cleaning."
) %>%
  kable_styling(full_width = FALSE)
```

Table 7: Verification of missingness after cleaning.

metric	value
Total missing values in final dataset	0

The Role of Data Cleaning in Exploratory Data Analysis

Data cleaning is one of the most consequential components of Exploratory Data Analysis because EDA is fundamentally a process of discovery that relies on accurate representation of structure rather than premature model imposition (Princeton Univ NJ Dept of Statistics & Tukey, 1993). If the dataset contains large pockets of missingness, inconsistent encodings, or blank categorical values that masquerade as legitimate categories, then exploratory summaries and visualizations can become distorted. This distortion is not trivial. It affects the shape of distributions, the apparent frequency of categories, and the stability of relationships between variables.

The quality of the data determines whether an analyst is observing real structure or measurement artifact. A histogram of an inspection score can shift if missing values are deleted in a nonrandom way. A bar chart of cuisine categories can become misleading if blank values are treated as a valid cuisine rather than as missing information. In applied decision environments, these distortions become operational risks. If the cleaned dataset is used to monitor compliance patterns or to allocate inspection resources, then biased summaries can cause misallocation and weaken the validity of the decisions derived from the analysis.

Cleaning also has a direct relationship to reproducibility. EDA is not simply about producing attractive figures. It is about documenting how the dataset was transformed into an analytical object, and ensuring that another analyst could reproduce the same cleaned dataset and arrive at the same conclusions. For that reason, missingness visualization, threshold rules, and imputation choices must be explicit, consistent, and defensible (National Institute of Standards and Technology, 2012). This perspective aligns with the foundational EDA philosophy that emphasizes open exploration prior to model commitment (Princeton Univ NJ Dept of Statistics & Tukey, 1993).

Effects of Imputation Methods

Different missing data strategies can meaningfully change a dataset. Deletion and imputation represent tradeoffs between bias, variance, and interpretability (Little & Rubin, 2019).

9.1 Deletion and Threshold Filtering

Deletion of rows or columns is simple and transparent, but it reduces sample size and can bias results if missingness is not completely random. Row deletion is especially costly when many rows have small amounts of missingness spread across variables. Column deletion can remove important predictors or outcomes, and it can narrow the analytical scope of the dataset. The twenty percent rule used here provides a balance by removing only those rows and variables that are too incomplete to support stable analysis (Little & Rubin, 2019).

9.2 Mean, Median, and Mode Imputation

Mean imputation is commonly used for numeric data, but it can artificially shrink variance and weaken correlations because it pulls missing observations toward a single central value. If the distribution is skewed or contains outliers, mean imputation can also produce an imputed value that is not representative of a typical observation (Little & Rubin, 2019). Median imputation addresses this risk by using a central value that is robust to extreme observations. Median imputation still reduces variance to some extent because it introduces repeated central values, but it is generally less sensitive to skew and outliers than mean substitution (Little & Rubin, 2019).

Mode imputation is the most direct approach for categorical variables because it preserves valid category membership. Its primary disadvantage is that it can inflate the most common category and reduce the apparent diversity of categories, especially when missingness is not small. For that reason, categorical imputation should be combined with threshold based removals so that imputation is used only when the remaining missingness is limited (Little & Rubin, 2019).

9.3 Missingness Mechanisms: MCAR, MAR, and MNAR

The impact of deletion and imputation depends on the mechanism that generates missing values. Missing Completely At Random (MCAR) implies that missingness is unrelated to both observed and unobserved data. Missing At Random (MAR) implies that missingness can be explained by observed variables, such as missing grades occurring more frequently for newly added restaurants that have not yet received a complete grading cycle. Missing Not At Random (MNAR) implies

that missingness is related to the unobserved value itself, such as missing grades occurring disproportionately in establishments with unusually poor inspection outcomes (Little & Rubin, 2019).

In this dataset, missingness patterns are plausibly a mixture of MAR and MNAR. That matters because deletion can introduce bias when missingness is systematic. Simple imputation can also introduce bias if missing values occur disproportionately in high risk establishments or specific inspection types. For that reason, missingness visualization is a necessary first step because it makes structural patterns visible and supports a more cautious interpretation of imputed values (National Institute of Standards and Technology, 2012). When inference is the primary objective, multiple imputation is often preferred because it preserves uncertainty more effectively than a single completed dataset by explicitly modeling missingness variability (Rubin, 1987).

9.4 Treatment of Outliers and Extreme Observations

Decisions related to outliers and extreme observations were explicitly documented to preserve analytical transparency and substantive interpretability. In this analysis, outliers were not capped or removed because the inspection score is a compliance indicator where extreme values can be substantively meaningful. Very high scores may correspond to unusually poor inspection outcomes or repeated violations. Removing or winsorizing those values during exploratory analysis can hide precisely the operational risk patterns that the dataset is designed to capture.

Instead, the approach taken here was to use robust descriptive strategies during cleaning. Median imputation reduces sensitivity to extreme values during missing value replacement. Subsequent modeling or reporting steps can incorporate additional robust methods, such as trimmed summaries or robust regression, if outliers are found to dominate results (Little & Rubin, 2019).

Conclusion

This analysis demonstrates a complete missing data workflow within an EDA context. Missingness was made visible using Amelia missing data maps, then controlled using a clear threshold rule for removing rows and columns with substantial incompleteness. Remaining missing values were imputed using a defensible, type aware strategy that respects variable meaning and measurement constraints. A post cleaning missingness map confirmed that the final dataset is complete and ready for subsequent descriptive and modeling workflows.

References

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data* (3rd ed.). Wiley.

National Institute of Standards and Technology. (2012). *NIST/SEMATECH e-Handbook of Statistical Methods: Exploratory data analysis (EDA).* https://www.itl.nist.gov/div898/handbook/eda/

Princeton Univ NJ Dept of Statistics, & Tukey, J. W. (1993). *Exploratory data analysis: Past, present, and future.* DTIC and NTIS.

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys.* Wiley. https://doi.org/10.1002/9780470316696

Wong, M. R., McKelvey, W., Ito, K., Schiff, C., Jacobson, J. B., & Kass, D. (2015). Impact of a letter grade program on restaurant sanitary conditions and diner behavior in New York City. *American Journal of Public Health, 105*(3), e81-e87. https://doi.org/10.2105/AJPH.2014.302404