Assignment 1 — Evaluate a New Data Set in the Analytical Environment

Juan Maldonado Franco

January 31, 2026

Contents

Executive Summary

This report evaluates the structural integrity and measurement properties of the DOHMH New York City Restaurant Inspection Results dataset, a regulatory repository containing approximately 296,854 observations of food service establishment inspections. The primary objective of this analysis was not merely to summarize the data, but to rigorously validate its readiness for advanced statistical modeling. By applying a structured Exploratory Data Analysis (EDA) framework within the R analytical environment, this study transforms the raw, software-inferred dataset into a semantically accurate analytical object.

The evaluation followed a four-stage algorithmic workflow. First, the environment was initialized to conditionally acquire the dataset and its associated data dictionary, ensuring portability and reproducibility. Second, a structural audit was conducted utilizing str() and summary() functions alongside metadata verification to identify discrepancies between the stored data types and their conceptual definitions. Third, a semantic reclassification was performed to manually recode misclassified variables—specifically transforming nominal identifiers stored as integers into factors and encoding the inspection grade as an ordered factor. Finally, each variable was formally mapped to Stevens' (1946) measurement scales (nominal, ordinal, interval, ratio) to define the boundaries of valid statistical inference.

The structural audit revealed significant misalignment between the raw data import and the dataset's logical structure. Notably, administrative identifiers such as CAMIS (entity ID) and ZIPCODE were automatically typed as numeric doubles. If left uncorrected, this would permit nonsensical arithmetic operations, such as calculating the "average zip code." Furthermore, the analysis clarified distinct measurement hierarchies: the SCORE variable was identified as a discrete ratio variable where lower values indicate better compliance, whereas GRADE was identified as an ordinal variable with unequal intervals between ranks. This means that the dataset, in its transformed state, is now a reliable foundation for public health surveillance and compliance modeling (Wong et al., 2015).

Introduction

Exploratory Data Analysis (EDA) is an iterative and conceptually disciplined approach to understanding a dataset prior to formal statistical modeling or inference. In contrast to confirmatory analysis, which evaluates predefined hypotheses, EDA emphasizes structural discovery: identifying variable characteristics, detecting anomalies and irregularities, and clarifying which analytical operations are valid given the dataset's measurement properties. A central requirement of rigorous EDA is semantic alignment between a variable's conceptual meaning and its computational representation. When variables are imported with incorrect data types—for example, categorical codes treated as numeric measurements—subsequent summaries, visualizations, and models may become invalid or misleading.

This report evaluates a public-health dataset within the R analytical environment using a structured EDA workflow aligned with foundational steps emphasized in the course. Specifically, the analysis focuses on importing and interpreting dataset documentation, assessing the structure of the available variables, and classifying variables according to (a) qualitative versus quantitative status, (b) Stevens' measurement typologies (nominal, ordinal, interval, ratio), and (c) discrete versus continuous granularity. The intended outcome is an analytically coherent representation of the dataset that supports defensible descriptive analysis and establishes a reliable foundation for subsequent modeling.

Analytical Environment and Dataset Acquisition

All analyses are conducted using the R statistical computing language within the RStudio integrated development environment. R is widely used for exploratory and statistical workflows due to its robust support for structured data manipulation, missing-data handling, and reproducible reporting through R Markdown (Wickham, 2014). R Markdown integrates narrative text, executable code, and analytical output in a single document, supporting transparency, reproducibility, and auditability of analytic decisions.

The goal of this section is to establish a fully configured analytical environment by acquiring, validating, and loading both the inspection results dataset and its associated documentation. Although the present assignment focuses primarily on dataset structure and variable typologies, loading both resources at this stage ensures continuity across subsequent analyses.

2.1 Dataset Selection

The dataset evaluated in this assignment is the DOHMH New York City Restaurant Inspection Results dataset, hosted on the NYC Open Data platform. This dataset captures regulatory inspection outcomes for food service establishments in New York City and is commonly used in public-health surveillance and compliance analytics. It supports typical EDA objectives such as evaluating inspection outcomes, identifying temporal inspection patterns, and examining the structure and measurement properties of administrative and observational variables.

2.2 Files Used for This Assignment

Two files are used in this analysis:

- DOHMH_New_York_City_Restaurant_Inspection_Results_20260127.csv: This file contains the inspection results data and constitutes the primary dataset to be evaluated during the EDA process.
- RestaurantInspectionDataDictionary_09242018.xlsx: This Excel-based data dictionary provides detailed documentation of the dataset's variables, including column names, definitions, and expected value encodings. The data dictionary is used to establish variable semantics prior to classification and recoding. At the PhD level, this step is critical, as variable type decisions must be justified by documented meaning rather than software-assigned defaults.

2.3 Library Setup

The following libraries are used for data acquisition, documentation parsing, and table presentation. Package installation commands are included but commented out to avoid unnecessary reinstallation.

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

library(readr)
library(readxl)
library(dplyr)
library(janitor)
library(knitr)
library(kableExtra)
library(stringr)
```

2.4 Conditional File Acquisition

To support reproducibility and portability of the analysis, both files are conditionally downloaded only if they are not already present in the working directory. If local copies exist, they are used directly.

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

# Source URLs (NYC Open Data)
# Note: Raw URL strings used to ensure reliable download
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/RestaurantInspectionDataDi

# 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.")
}

# Conditional download: data dictionary
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.")
}
```

2.5 Loading the Data Dictionary

The data dictionary is loaded from the “Column Info” tab of the Excel file. This tab contains the variable-level metadata required to interpret column meanings and guide classification decisions.

```r
dict_raw <- read_excel(dict_file, sheet = "Column Info", col_names = FALSE)

# Extract the second row as column names
new_colnames <- dict_raw %>%
  slice(2) %>%
  unlist(use.names = FALSE) %>%
  as.character()

# Apply column names and remove first two rows to retain definitions
dict_clean <- dict_raw %>%
  slice(-c(1, 2)) %>%
  setNames(new_colnames) %>%
  clean_names()
```

```
# Standardize dictionary into a minimal joinable form
dict <- dict_clean %>%
  transmute(
    column_name_raw = column_name,
    column_description = column_description,
    column_name = janitor::make_clean_names(column_name_raw)
  ) %>%
  filter(!is.na(column_name_raw))

# Preview
head(dict, 5)
```

```
## # A tibble: 5 x 3
##   column_name_raw column_description                                   column_name
##   <chr>           <chr>                                                <chr>
## 1 CAMIS           Unique identifier for the establishment (restaura~ camis
## 2 DBA             Establishment (restaurant) name                      dba
## 3 BORO            Borough of establishment (restaurant) location       boro
## 4 BUILDING        Building number for establishment (restaurant) lo~ building
## 5 STREET          Street name for establishment (restaurant) locati~ street
```

At this stage, no transformations, re-coding, or filtering are applied. The dataset is loaded in its raw form to preserve the integrity of subsequent exploratory steps.

2.6 Loading the Inspection Results Dataset

The inspection results dataset is read into R and assigned to a working data frame. Column names are standardized using clean_names() to facilitate consistent referencing throughout the analysis.

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

df <- df_raw

# Basic verification
nrow(df)
```

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

```
ncol(df)
```

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

At the conclusion of this section, both the inspection results dataset and its accompanying documentation are available within the analytical environment. Subsequent sections build on this foundation to assess dataset structure, identify misclassified variables, and assign formal measurement typologies.

Dataset Structure and Initial Inspection

The first analytical step after data acquisition is a systematic inspection of the dataset's structure. This step serves two purposes. First, it establishes a baseline understanding of the dataset as it has been imported into the analytical environment, including the number of observations, number of variables, and software-assigned data types. Second, it provides early signals of potential issues—such as misclassified variables, unexpected missingness, or anomalous value patterns—that must be addressed before formal variable classification and recoding.

3.1 Dimensionality of the Dataset

The inspection results dataset was loaded in its raw form without any filtering, transformation, or recoding. The dimensionality of the dataset was assessed by examining the number of rows (observations) and columns (variables).

```r
dim(df)
```

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

These values establish the overall scale of the dataset and inform later decisions regarding visualization, summarization, and computational feasibility.

3.2 Software-Assigned Structure and Data Types

The internal structure of the dataset was examined using R's str() function, which reports each variable's name, assigned data type, and a preview of its values. This step reflects how the data are currently represented in memory, not how they should be represented conceptually.

```r
str(df)
```

```
## spc_tbl_ [296,854 x 27] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ camis                : num [1:296854] 50145220 50106103 50127358 41582828 50079841 ...
##  $ dba                  : chr [1:296854] "FELIZ TEA HOUSE" "XIANG JU RESTAURANT" "LA BAKER"
##  $ boro                 : chr [1:296854] "Brooklyn" "Brooklyn" "Queens" "Queens" ...
##  $ building             : chr [1:296854] "5718" "5112" "72-08" NA ...
##  $ street               : chr [1:296854] "5 AVENUE" "8 AVENUE" "BROADWAY" "LA GUARDIA AIRPO
##  $ zipcode              : num [1:296854] 11220 11220 11372 11369 11101 ...
##  $ phone                : chr [1:296854] "9172151523" "9173880989" "3478087000" "3478675394"
##  $ cuisine_description  : chr [1:296854] "Chinese" "Chinese" "Bakery Products/Desserts" "Sa
##  $ inspection_date      : chr [1:296854] "03/21/2024" "12/23/2025" "09/30/2024" "07/27/2017"
##  $ action               : chr [1:296854] "Violations were cited in the following area(s)." "
##  $ violation_code       : chr [1:296854] "06E" "04L" "06F" "10H" ...
##  $ violation_description: chr [1:296854] "Sanitized equipment or utensil, including in-use 
##  $ critical_flag        : chr [1:296854] "Critical" "Critical" "Critical" "Not Critical" ..
##  $ score                : num [1:296854] 22 46 53 11 9 34 NA 10 31 26 ...
##  $ grade                : chr [1:296854] NA NA "C" "A" ...
##  $ grade_date           : chr [1:296854] NA NA "09/30/2024" "07/27/2017" ...
##  $ record_date          : chr [1:296854] "01/27/2026" "01/27/2026" "01/27/2026" "01/27/2026"
##  $ inspection_type      : chr [1:296854] "Pre-permit (Operational) / Initial Inspection" "Cy
##  $ latitude             : num [1:296854] 40.6 40.6 40.7 NA 40.8 ...
##  $ longitude            : num [1:296854] -74 -74 -73.9 NA -73.9 ...
##  $ community_board      : num [1:296854] 307 307 403 NA 401 407 501 309 315 104 ...
##  $ council_district     : chr [1:296854] "38" "43" "25" NA ...
##  $ census_tract         : chr [1:296854] "007400" "010800" "026500" NA ...
##  $ bin                  : num [1:296854] 3015687 3013766 4029786 NA 4004620 ...
##  $ bbl                  : num [1:296854] 3.01e+09 3.01e+09 4.01e+09 NA 4.00e+09 ...
##  $ nta                  : chr [1:296854] "BK32" "BK34" "QN50" NA ...
##  $ location             : chr [1:296854] "POINT (-74.014567977722 40.641028000405)" "POINT
##  - attr(*, "spec")=
##   .. cols(
##   ..   CAMIS = col_double(),
##   ..   DBA = col_character(),
##   ..   BORO = col_character(),
##   ..   BUILDING = col_character(),
##   ..   STREET = col_character(),
##   ..   ZIPCODE = col_double(),
##   ..   PHONE = col_character(),
##   ..   ‘CUISINE DESCRIPTION‘ = col_character(),
##   ..   ‘INSPECTION DATE‘ = col_character(),
##   ..   ACTION = col_character(),
##   ..   ‘VIOLATION CODE‘ = col_character(),
##   ..   ‘VIOLATION DESCRIPTION‘ = col_character(),
##   ..   ‘CRITICAL FLAG‘ = col_character(),
##   ..   SCORE = col_double(),
##   ..   GRADE = col_character(),
##   ..   ‘GRADE DATE‘ = col_character(),
##   ..   ‘RECORD DATE‘ = col_character(),
##   ..   ‘INSPECTION TYPE‘ = col_character(),
##   ..   Latitude = col_double(),
##   ..   Longitude = col_double(),
##   ..   ‘Community Board‘ = col_double(),
##   ..   ‘Council District‘ = col_character(),
##   ..   ‘Census Tract‘ = col_character(),
##   ..   BIN = col_double(),
##   ..   BBL = col_double(),
##   ..   NTA = col_character(),
##   ..   Location = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

At this stage, the dataset typically contains a mixture of character, numeric, integer, and date-like variables. Because R infers types based on observed values during import, these assignments may not align with the documented meaning of each variable. Consequently, the output of str() is treated as a diagnostic artifact rather than a definitive classification.

3.3 Summary Statistics and Initial Distributions

To complement the structural overview, summary statistics were generated for all variables using the summary() function. For numeric variables, this includes measures of central tendency and dispersion; for categorical variables, it includes frequency counts and missing-value indicators.

```r
summary(df)
```

```
##      camis              dba                boro             building
##  Min.   :30075445   Length:296854      Length:296854      Length:296854
##  1st Qu.:50003242   Class :character   Class :character   Class :character
##  Median :50090490   Mode  :character   Mode  :character   Mode  :character
##  Mean   :48032170
##  3rd Qu.:50128889
##  Max.   :50181188
##
##     street             zipcode         phone           cuisine_description
##  Length:296854      Min.   : 6605   Length:296854      Length:296854
##  Class :character   1st Qu.:10023   Class :character   Class :character
##  Mode  :character   Median :11101   Mode  :character   Mode  :character
##                     Mean   :10707
##                     3rd Qu.:11232
##                     Max.   :69361
##                     NA’s   :3126
##  inspection_date       action          violation_code     violation_description
##  Length:296854      Length:296854      Length:296854      Length:296854
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  critical_flag         score           grade            grade_date
##  Length:296854      Min.   :  0.00   Length:296854      Length:296854
##  Class :character   1st Qu.: 12.00   Class :character   Class :character
##  Mode  :character   Median : 21.00   Mode  :character   Mode  :character
##                     Mean   : 25.17
##                     3rd Qu.: 33.00
##                     Max.   :203.00
##                     NA’s   :16364
##  record_date        inspection_type      latitude       longitude
##  Length:296854      Length:296854      Min.   : 0.00   Min.   :-74.25
##  Class :character   Class :character   1st Qu.:40.69   1st Qu.:-73.99
##  Mode  :character   Mode  :character   Median :40.73   Median :-73.96
##                                        Mean   :40.30   Mean   :-73.16
##                                        3rd Qu.:40.76   3rd Qu.:-73.89
##                                        Max.   :40.91   Max.   :  0.00
##                                        NA’s   :1356    NA’s   :1356
##  community_board council_district   census_tract            bin
##  Min.   :101     Length:296854     Length:296854     Min.   :1000000
##  1st Qu.:106     Class :character   Class :character   1st Qu.:1051222
##  Median :302     Mode  :character   Mode  :character   Median :3022124
##  Mean   :255                                           Mean   :2586330
##  3rd Qu.:401                                           3rd Qu.:4011431
##  Max.   :595                                           Max.   :5799501
##  NA’s   :4472                                          NA’s   :5764
##       bbl                nta              location
##  Min.   :1.000e+00   Length:296854      Length:296854
##  1st Qu.:1.011e+09   Class :character   Class :character
##  Median :3.008e+09   Mode  :character   Mode  :character
##  Mean   :2.479e+09
##  3rd Qu.:4.006e+09
##  Max.   :5.270e+09
##  NA’s   :1356
```

This output provides an initial view of value ranges, category distributions, and the presence of missing values. Importantly, summary statistics at this stage are interpreted cautiously, as misclassified variables may produce misleading summaries (e.g., numeric summaries for coded categorical variables).

3.4 Alignment with Data Dictionary Metadata

To support principled variable classification, the dataset structure was aligned with the official data dictionary. The "Column Info" tab of the data dictionary provides variable definitions and contextual information that are essential for interpreting the meaning and intended measurement level of each column.

A structural audit table was constructed that combines software-assigned information from the dataset with documented metadata from the data dictionary.

```r
# Structural audit of dataset variables
dataset_structure <- data.frame(
  column_name = names(df),
  r_class = sapply(df, function(x) paste(class(x), collapse = ", ")),
  n_unique = sapply(df, dplyr::n_distinct),
  n_missing = sapply(df, function(x) sum(is.na(x))),
  stringsAsFactors = FALSE
)

# Align dataset structure with dictionary metadata
structure_with_dict <- dataset_structure %>%
  left_join(dict, by = c("column_name" = "column_name"))

kable(
  structure_with_dict %>% select(column_name, r_class, n_unique, column_description),
  caption = "Dataset structure aligned with DOHMH data dictionary metadata."
) %>%
  kable_styling(full_width = FALSE)
```

Table 1: Dataset structure aligned with DOHMH data dictionary metadata.

column_name	r_class	n_unique	column_description
camis	numeric	30702	Unique identifier for the establishment (restaurant)
dba	character	24263	Establishment (restaurant) name
boro	character	6	Borough of establishment (restaurant) location
building	character	8533	Building number for establishment (restaurant) location
street	character	2157	Street name for establishment (restaurant) location
zipcode	numeric	250	Zip code of establishment (restaurant) location
phone	character	27216	Phone number
cuisine_description	character	91	Establishment (restaurant) cuisine
inspection_date	character	2062	NA
action	character	6	Action associated with each establishment (restaurant) inspectio
violation_code	character	158	Violation code associated with an establishment (restaurant) ins
violation_description	character	252	Violation description associated with an establishment (restaura
critical_flag	character	3	Indicator of critical violation
score	numeric	145	Total score for a particular inspection
grade	character	7	Grade associated with the inspection
grade_date	character	1753	Date when grade was issued to the establishment (restaurant)
record_date	character	1	Date record was added to dataset
inspection_type	character	35	A combination of the inspection program and the type of inspec
latitude	numeric	24606	NA
longitude	numeric	24605	NA
community_board	numeric	70	NA
council_district	character	52	NA
census_tract	character	1184	NA
bin	numeric	21251	NA
bbl	numeric	20856	NA
nta	character	194	NA
location	character	24605	NA

This table provides a consolidated view of each variable's name and software-assigned type, number of distinct values, extent of missingness, and documented definition. Identifying the true nature of the variables requires this juxtaposition of computational status and semantic definition.

3.5 Initial Observations and Implications for Variable Classification

The combined inspection of str(df), summary(df), and the data dictionary highlights the need for careful variable classification in subsequent steps. In particular, variables with low cardinality but numeric storage, variables representing administrative identifiers, and variables encoding outcomes or categories using text or codes are flagged for closer evaluation. Similarly, variables with substantial missingness or unexpected value ranges warrant further scrutiny before inclusion in descriptive analysis or modeling. At this stage, no variables are recoded or reclassified. Instead, this section establishes the empirical and documentary basis for the variable classification and measurement typology decisions that follow.

Variable Classification and Typologies

The alignment of R-native data types with the conceptual intent of the DOHMH dataset requires manual intervention. Based on the structural audit performed in Section 3, several variables were identified as being stored in formats that contradict their measurement level—specifically administrative identifiers stored as numeric doubles and categorical flags stored as character strings. This misalignment can lead to erroneous statistical assumptions if not corrected.

4.1 Data Transformation and Recoding

In this step, the variables were reclassified to ensure that subsequent statistical operations are mathematically and logically sound. This includes converting nominal identifiers to character strings (or factors) to prevent accidental arithmetic operations and transforming qualitative attributes into factors.

```r
# Reclassify variables based on documented semantics
df_clean <- df %>%
  mutate(
    # Reclassify IDs and codes as Factors (Categorical)
    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),

    # Transform qualitative attributes into factors for categorical analysis
    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),

    # Establish ordinality for Grade
    grade = factor(grade, levels = c("A", "B", "C", "P", "Z"), ordered = TRUE),

    # Standardize temporal data
    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")
  )
```

```r
# Verify the final R classes for key variables
str(df_clean[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" ...
```

The transformation of grade into an ordered factor is particularly important. Since grade implies a hierarchy (A is better than B), treating it as a standard nominal character string would lose this inherent information during analysis.

4.2 Measurement Scale and Typology Assignment

The following table summarizes the formal classification of all 27 variables within the DOHMH dataset. Each assignment is based on Stevens' (1946) typologies, ensuring that the chosen statistical methods align with the underlying properties of the data.

```r
# Comprehensive Typology for All Variables
typology_all <- structure_with_dict %>%
  select(column_name, column_description) %>%
  mutate(
    `Quant/Qual` = case_when(
      column_name %in% c("score", "latitude", "longitude") ~ "Quantitative",
      column_name %in% c("inspection_date", "grade_date", "record_date") ~ "Quantitative (Inte
      TRUE ~ "Qualitative"
    ),
    `Stevens Typology` = case_when(
      column_name %in% c("score") ~ "Ratio",
      column_name %in% c("latitude", "longitude") ~ "Interval", # Coordinates
      column_name %in% c("inspection_date", "grade_date", "record_date") ~ "Interval",
      column_name == "grade" ~ "Ordinal",
      TRUE ~ "Nominal"
    ),
    `Granularity` = case_when(
      column_name %in% c("latitude", "longitude") ~ "Continuous",
      TRUE ~ "Discrete"
    ),
    `Rationale` = case_when(
      column_name == "score" ~ "Point total; lower values indicate better compliance. Zero is 
      column_name == "grade" ~ "Ranked categories (A > B > C).",
      column_name %in% c("latitude", "longitude") ~ "Geospatial coordinates.",
      column_name %in% c("inspection_date", "grade_date", "record_date") ~ "Dates; differences
      TRUE ~ "Identifier, label, or unordered category."
    )
  )
```

```
kable(typology_all, caption = "Complete Variable Classification and Measurement Typologies") %
  kable_styling(latex_options = "hold_position", full_width = FALSE, font_size = 10)
```

Table 2: Complete Var
pologies

column_name	column_description	Quant/Qual
camis	Unique identifier for the establishment (restaurant)	Qualitative
dba	Establishment (restaurant) name	Qualitative
boro	Borough of establishment (restaurant) location	Qualitative
building	Building number for establishment (restaurant) location	Qualitative
street	Street name for establishment (restaurant) location	Qualitative
zipcode	Zip code of establishment (restaurant) location	Qualitative
phone	Phone number	Qualitative
cuisine_description	Establishment (restaurant) cuisine	Qualitative
inspection_date	NA	Quantitative (In
action	Action associated with each establishment (restaurant) inspection	Qualitative
violation_code	Violation code associated with an establishment (restaurant) inspection	Qualitative
violation_description	Violation description associated with an establishment (restaurant) inspection	Qualitative
critical_flag	Indicator of critical violation	Qualitative
score	Total score for a particular inspection	Quantitative
grade	Grade associated with the inspection	Qualitative
grade_date	Date when grade was issued to the establishment (restaurant)	Quantitative (In
record_date	Date record was added to dataset	Quantitative (In
inspection_type	A combination of the inspection program and the type of inspection performed	Qualitative
latitude	NA	Quantitative
longitude	NA	Quantitative
community_board	NA	Qualitative
council_district	NA	Qualitative
census_tract	NA	Qualitative
bin	NA	Qualitative
bbl	NA	Qualitative
nta	NA	Qualitative
location	NA	Qualitative

This classification framework serves as the blueprint for all future analysis. This means that any future attempt to calculate the mean of grade or the sum of camis identifiers will be flagged as analytically invalid, preserving the integrity of the research.

Importance of Variable Types in EDA

A fundamental pillar of Exploratory Data Analysis (EDA) is the recognition that variable type dictates the boundaries of valid inference. EDA is widely defined as an approach for maximizing insight into a dataset by uncovering structure, detecting anomalies, testing assumptions, and guiding subsequent modeling decisions (NIST, 2012). While the conceptual foundations of EDA were introduced by Princeton Univ NJ Dept Of Statistics and Tukey (1993), later treatments emphasize that exploratory analysis is not merely descriptive but methodologically consequential: incorrect assumptions about measurement levels can systematically distort statistical conclusions, visual interpretation, and model behavior.

In contemporary data science workflows, the importance of correctly identifying variable types spans three interrelated domains: statistical validity, visualization and automated discovery, and feature engineering for predictive modeling.

Statistical and Analytical Validity

The application of measurement typologies, first formalized by Stevens (1946), remains a requirement for mathematical coherence in statistical analysis. Measurement scales determine which operations are permissible and which inferential procedures are valid. For example, calculating the mean of a Zip Code or a categorical identifier produces a numerically defined quantity that lacks any real-world interpretation. Similarly, treating an ordinal variable as if it were measured on a ratio or interval scale implicitly assumes equal spacing between categories—an assumption that is rarely justified.

Violations of these assumptions can lead to biased estimates, invalid hypothesis tests, and misleading measures of association. As emphasized in modern statistical guidance, exploratory analysis must explicitly evaluate whether the underlying scale of measurement supports the intended analytical operation before applying parametric methods (NIST, 2012; Tukey, 1977).

Constraints on Visualization and Automated Discovery

Variable types also act as a primary constraint on effective visualization and heuristic-driven discovery. Appropriate graphical representations depend directly on measurement level: nominal variables are meaningfully summarized using frequency-based charts, while quantitative variables require distributional displays such as histograms, boxplots, or density plots. When these constraints are ignored, visualizations may obscure patterns, exaggerate differences, or create the illusion of structure where none exists.

Recent work on automated exploratory data analysis (AutoEDA) systems highlights that correct identification of variable types is essential for algorithmic selection of visual encodings and summary statistics (Staniak & Biecek, 2019). These systems build on data-structuring principles articulated by Wickham (2014), where variables correspond to columns and observations to rows. Misclassification at the variable-typing stage therefore propagates directly into automated visualization pipelines, increasing the risk of misleading or uninterpretable outputs.

Implications for Feature Engineering and Predictive Modeling

In the transition from EDA to predictive modeling, variable types define the semantic contract of the data and determine appropriate encoding strategies. Nominal variables require representation through one-hot encoding or embeddings, ordinal variables must preserve rank information, and quantitative variables may require scaling or transformation depending on model assumptions. Errors introduced during variable classification can cascade through the modeling pipeline, affecting both performance and interpretability.

Sambasivan et al. (2021) describe such upstream errors as “data cascades,” showing that seemingly minor data preparation decisions often become a dominant source of technical debt and model failure in high-stakes systems. From this perspective, the typology decisions made during EDA are not ancillary but foundational: they constrain which models can be applied, how features are encoded, and how results can be interpreted with confidence.

Conclusion

The transition from raw data to actionable intelligence requires a disciplined focus on structural discovery. This evaluation demonstrates that the DOHMH New York City Restaurant Inspection Results dataset, while rich in observational detail, required significant semantic intervention to be rendered “analysis-ready.”

Through the systematic application of EDA principles, this report identified and corrected critical type mismatches that would have otherwise compromised the validity of downstream statistical operations. By aligning the computational representation of variables like CAMIS, SCORE, and GRADE with their conceptual definitions, we have established a robust data foundation. This process confirms the central thesis of the assignment: that the validity of any statistical model is predetermined by the rigorous classification of its constituent variables. The dataset is now formally structured, semantically aligned, and prepared for advanced descriptive and predictive analysis.

References

NIST. (2012). NIST/SEMATECH e-Handbook of statistical methods: Exploratory data analysis (EDA). National Institute of Standards and Technology. https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm

NYC Open Data. (2026). DOHMH New York City restaurant inspection results [Data set]. https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j

Princeton Univ Nj Dept Of Statistics, & Tukey, J. W. (1993). Exploratory Data Analysis: Past, Present and Future. DTIC AND NTIS.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021). “Everyone wants to do the model work, not the data work”: Data cascades in high-stakes AI. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 1–15. https://doi.org/10.1145/3411764.3445518

Staniak, M., & Biecek, P. (2019). The Landscape of R Packages for Automated Exploratory Data Analysis. https://doi.org/10.32614/RJ-2019-033

Stevens, S. S. (1946). On the Theory of Scales of Measurement. Science, 103(2684), 677–680.

Wickham, H. (2014). Tidy data. Journal of Statistical Software, 59(10), 1–23. https://doi.org/10.18637/jss.v059.i10

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