Juan Cruz Ferreyra
PPUA 5262 | Northeastern University
October 2024

Boston Food Establishments Inspections dataset

This document includes a brief introduction to the data source, a fun facts section highlighting key insights, and a visualization section. An appendix with the analysis code is also provided.

Overview

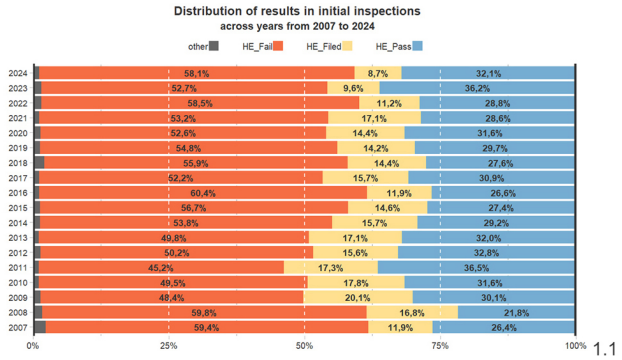
The Boston Food Establishment Inspections dataset contains the outcomes of food establishment inspections conducted in the city. These inspections are carried out by the Health Division of the Department of Inspectional Services, tasked with ensuring compliance with local sanitary codes. Every business serving food is inspected at least once a year, with high-risk establishments receiving follow-up inspections. Health inspections are also triggered by complaints regarding unsanitary conditions or illness.

Fun Facts

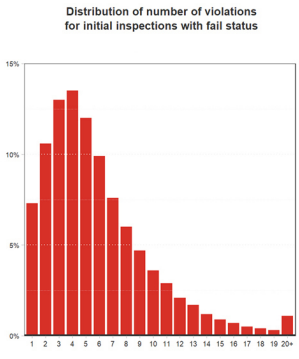
- The dataset includes records from inspections conducted since 2006. Each record typically represents a single violation, though some records reflect the overall inspection outcome (e.g., no violations or incomplete inspections). As of September 13, 2024, the dataset contains 807,805 records.[1]
- The dataset contains 27 variables, covering business administration, establishment location, inspection details, and specific violation information. When certain variables don't apply to a particular record—such as violation fields for inspections with no violations—those fields are marked as NA (empty).[2]
- The columns include identifiers, spatial and temporal data, categorical values (e.g., violation severity, inspection outcomes), and free text fields with violation descriptions or inspector comments.[3]
- The variables licenseno (establishment license number) and resultdttm (inspection date and time) allow us to group records from the same inspection. By aggregating the data using these columns, we identify 195,221 unique inspections conducted during the dataset's time period.[4]
- Inspections with a "Fail" status typically result in a follow-up, which occurs within the following month in almost 97% of cases. Additional follow-ups may be triggered if violations are not properly addressed or new ones are found. By linking these inspections, we can trace the sequence of follow-ups back to the initial "Fail" inspection and infer their order within the series.[5]
- The most common outcome for the first inspection in a series is "Fail," occurring in 53.5% of cases. This is followed by "Pass" (no violations found) at 30.4% and "Filed" (minor violations with no follow-up required) at 14.8%. Only 1.3% of initial inspections result in a different status *(fig. 1.1)*.[6]
- The distribution of violations from the first inspection in each series resembles a Poisson distribution. Most inspections report 6 or fewer violations, with 4 being the most frequent. As the number of violations increases, the occurrence of such inspections decreases *(fig 1.2)*.[7]
- Food establishments in Boston are concentrated in high-density areas like Downtown and Back Bay, while lower-density neighborhoods to the west and south feature fewer restaurants. In these lower-density areas, restaurants are typically situated along major roads and avenues.[8]
- In the Common SENSES study area, a significant number of establishments are concentrated around Blue Hill Avenue. Our analysis identified 12 clusters of businesses, each with at least 3 establishments, with 5 of those clusters located along the mentioned avenue *(fig 2.1)*.[9]
- Among the 12 identified clusters, 2 appear to function as food production facilities for various businesses. These locations predominantly have inspections with no violations recorded, likely benefiting from effective operational practices or political influence *(fig 3.1)*.[10]

Visualizations

The following two visualizations depict the proportion of outcomes in initial inspections by year *(fig. 1.1)*[11] and the number of violations associated with "Fail" status results *(fig 1.2)*[12].



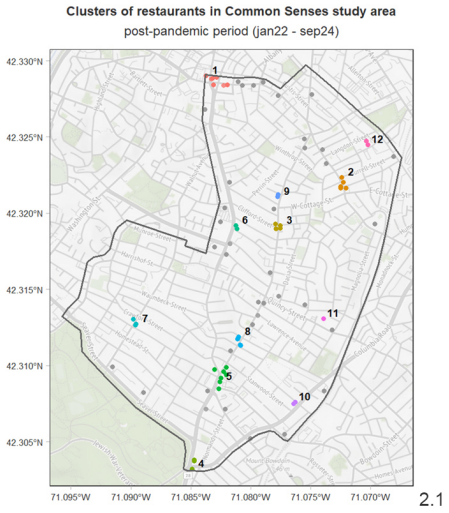
1.1



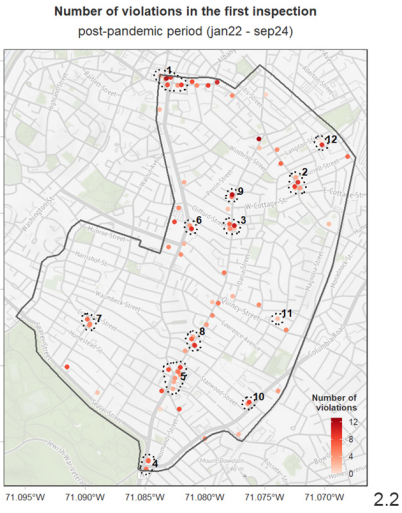
1.2

These two graphs include the initial inspections from each series in the dataset. The inspection outcomes plot reveals some variation in distribution across the years; however, a sustained pattern in these changes is not evident. The violations plot indicates that over half of the inspections with "Fail" status recorded 5 or fewer violations.

Next, we present the distribution of post-pandemic inspections within the Common SENSES study area and how they can be clustered through further analysis *(fig 2.1)*[13].



2.1



2.2

Five of the clusters are situated along Blue Hill Avenue. To complement the graph depicting the distribution of initial inspections and their associated violation counts *(fig 2.2)*[14], we aggregate the inspections by cluster and present the distribution using a box plot *(fig 3.1)*[15].

3.1

Appendix A: Data Dictionary

Business admin relevant columns:

- *licenseno:* licence number of the food establishment.
- *descript:* type of food establishment.

Inspection relevant columns:

- *result:* result of inspection.
- *resultdttm:* date on which the results were generated.

Violation record relevant columns:

- *violdesc:* reason of violation.
- *viol_level:* level of violation.
- *viol_status:* status for violation.
- *comments:* comments given to the food establishment for improvement.

Establishment location relevant columns:

- *location:* latitude and longitude of the food establishment.

Appendix B: Code

```
# 0: Import packages
library(ggmap)
library(ggplot2)
library(lubridate)
library(sf)
library(tidyverse)
library(tmap)
library(tmaptools)
```

```
# 0: Load dataset
filepath <- "path/to/file.csv"
df <- read.csv(filepath)
```

Fun Facts

```
# 1: Count number of records
df %>%
  nrow()
```

```
# 2: Show summary of the data
df %>%
  summary()
```

```
# 3: Show sample of 5 rows
set.seed(42) # for reproducibility

df %>%
  sample(5)
```

```
# 4: Aggregate data at inspection level and count
df_inspections <- df %>%
  mutate(violation_fail = ifelse(viol_status == "Fail", 1, 0))
  group_by(licenseno, result, resultdttm) %>%
  summarise(n_fail = sum(violation_fail, na.rm = TRUE), .groups = 'drop')

df_inspections %>%
  nrow()
```

```
# 5: Link follow-ups to initial inspections
# define result categories
nonreq_results <- c("HE_NotReq", "HE_OutBus", "DATAERR")
failed_results <- c(
  "HE_Fail", "HE_FailExt","HE_Hearing", "HE_Closure", "HE_TSOP", "HE_VolClos"
  )

# separate inspections not performed
df_inspections_req <- df_inspections %>%
  filter(!(result%in%nonreq_results))
df_inspections_nonreq <- df_inspections %>%
  filter(result%in%nonreq_results)
```

```r
# order data and create auxiliar columns
df_inspections_req <- df_inspections_req %>%
  arrange(licenseno, resultdttm)

df_inspections_req$same_as_prev <- (
  df_inspections_req$licenseno == lag(df_inspections_req$licenseno)
  )
df_inspections_req$time_diff <- (
  df_inspections_req$resultdttm - lag(df_inspections_req$resultdttm)
  )
df_inspections_req$previous_fail <- (
  lag(df_inspections_req$result)%in%failed_results
  )

df_inspections_req <- df_inspections_req %>%
  mutate(time_diff = seconds_to_period(time_diff))

# find initial inspections
df_inspections_req$first_inspection <- (!(
  df_inspections_req$same_as_prev &
  df_inspections_req$time_diff < ddays(60) &
  df_inspections_req$previous_fail
)) %>%
  replace_na(TRUE)

# assign inspection number to follow-ups
df_inspections_req <- df_inspections_req %>%
  mutate(inspection_number = ifelse(first_inspection, 1, NA)) %>%
  mutate(series = cumsum(first_inspection)) %>%
  group_by(series) %>%
  mutate(inspection_number = ifelse(
    is.na(inspection_number), row_number(), inspection_number)
    ) %>%
  ungroup() %>%
  select(-same_as_prev, -time_diff, -previous_fail, -first_inspection, -series)

# join processed dataframe with non required inspections
df_inspections_nonreq$inspection_number <- 0

df_inspections <- df_inspections_req %>%
  rbind(df_inspections_nonreq)
```

```r
# 6: Count status occurrence in the first inspections
df_inspections %>%
  filter(inspection_number==1) %>%
  count(result)
```

```r
# 7: Count number of violations per inspection
df_inspections %>%
  filter(inspection_number==1) %>%
  count(n_fail)
```

```r
# 8: Geolocate inspections
gdf_inspections <- df_inspections %>%
  filter(!(is.na(longitude) | is.na(latitude))) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326, remove = FALSE)
```

```r
# 9: Filter establishments in Common Senses and cluster them
# load common senses geometry and label inspection within its limits
filepath_cs_geom <- "path/to/common_senses_geometry.shp"
gdf_common_senses <- read_sf(filepath_cs_geom) %>%
  rename(common_senses = id)

gdf_inspections <- gdf_inspections %>%
  st_join(gdf_common_senses)

gdf_inspections$common_senses[is.na(gdf_inspections$common_senses)] <- 0

# filter post pandemic inspections in common senses study area
gdf_cs_postpand <- gdf_cs %>%
  filter(common_senses == 1 & resultdttm >= ymd("2022-01-01"))

# obtain reprojected coordinates for the clustering
gdf_cs_postpand_NAD83 <- gdf_cs_postpand %>%
  st_transform(crs = 2249)# Transform to NAD83

df_cs_postpand_NAD83 <- gdf_cs_postpand_NAD83 %>%
  cbind(data.frame(st_coordinates(gdf_cs_postpand_NAD83$geometry))) %>%
  st_set_geometry(NULL)

# clusterize restaurants
df_clusters <- df_cs_postpand_NAD83 %>%
  filter(!(result%in%nonreq_results)) %>%
  filter(common_senses==1) %>%
  group_by(licenseno) %>%
  slice(1)

clusters <- dbscan(df_clusters[c("X", "Y")], eps = 250, MinPts = 3)

df_clusters$cluster <- clusters$cluster

gdf_cs_postpand <- gdf_cs_postpand %>%
  left_join(df_clusters[c("licenseno", "cluster")], by="licenseno")
```

```r
# 10: Aggregate data at cluster level and count violations per initial inspection
gdf_cs_postpand %>%
  filter(inspection_number == 1) %>%
  group_by(cluster, n_fail) %>%
  summarise(count = n(), .groups = 'drop')
```

Visualization

```r
# 11: fig 1.1 - Distribution of results in initial inspections
# create the auxiliar dataframe to plot
count_results_year <- df_inspections %>%
```

```
  filter(inspection_number == 1 & !(result%in%nonreq_results)) %>%
  mutate(result = ifelse(
    (result%in%c("HE_Fail", "HE_Pass", "HE_Filed")), result, "other"
    ))

count_results_year <- count_results_year %>%
  mutate(year = year(resultdttm)) %>%
  filter(year > 2006) %>%
  group_by(year, result) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  filter(!(is.na(year)))

# plot
ggplot(count_results_year) +
    geom_bar(
      stat = "identity",
      aes(x = year, y = count, fill = result),
      position = position_fill()
    ) +
  coord_flip()
```

```
# 12: fig 1.2 - Distribution of number of violations in initial inspections
# create the auxiliar dataframe to plot
df_inspections_fail <- df_inspections %>%
  filter(inspection_number==1 & result=="HE_Fail")

df_inspections_fail <- df_inspections_fail %>%
  mutate(total_fail_bins = ifelse(total_fail>=20, "20+", total_fail))

count_total_fail <- df_inspections_fail %>%
  count(total_fail_bins) %>%
  mutate(perc = round(n/sum(n), 3))

# plot
ggplot(df, aes(x = total_fail_bins, y = perc)) +
  geom_bar(stat="identity")
```

```
# 13: fig 2.1 - Distribution of first inspections and their cluster
# get basemap for Common Senses study area
common_senses_geom <- as.data.frame(
  unique(st_coordinates(gdf_common_senses$geometry[[1]]))
  )

bbox_common_senses <- c(
  "left" = min(common_senses_geom$X) - 0.001,
  "bottom" = min(common_senses_geom$Y) - 0.001,
  "right" = max(common_senses_geom$X) + 0.001,
  "top" = max(common_senses_geom$Y) + 0.001
)

basemap_common_senses <- get_stadiamap(bbox=bbox_common_senses,
                               maptype="stamen_terrain",
```

```
                              zoom=15)

# plot first inspection in their corresponding cluster
ggmap(basemap_common_senses) +
  geom_sf(
    data=filter(gdf_cs_postpand, inspection_number==1), inherit.aes=FALSE
    ) +
  geom_sf(data=gdf_common_senses, fill = NA, inherit.aes=FALSE)
```

```
# 14: fig 2.2 - Distribution of first inspections and their number of violations
ggmap(basemap_common_senses, darken = c(0.6, "white")) +
  geom_sf(
    data=filter(gdf_cs_postpand, inspection_number==1),
    aes(color = total_fail),
    inherit.aes=FALSE
    ) +
  geom_sf(data=gdf_common_senses, fill = NA, inherit.aes=FALSE)
```

```
# 15: fig. 3.1 - Boxplot
ggplot(
    data = filter(df_cs_postpand, inspection_number==1 & cluster!=0),
    aes(x=cluster, y=total_fail, fill=cluster)
    ) +
  geom_boxplot() +
  geom_jitter()
```