---

title: 'Occupational Health in U.S. Transit Agencies: Trends in OSHA-Reportable Illnesses and Injuries (2016–2023)'

format:

  html:

    df_print: "paged"

    fig_caption: yes

    toc: true

    toc_depth: 3

    number_sections: true

    self-contained: true

    css: styles.css

execute:

  echo: true

  cache: false

  echo.comments: false

  message: false

  warning: false

editor_options:

  chunk_output_type: console

This document was rendered on `r format(Sys.time(), '%B %d, %Y')`.

```{r setup, include=FALSE}

rm(list = ls(all = TRUE))

if (!is.null(sessionInfo()$otherPkgs)) {
```

```
  res <- suppressWarnings(
    lapply(paste('package:', names(sessionInfo()$otherPkgs), sep=""),
           detach, character.only=TRUE, unload=TRUE, force=TRUE))
}
```

```{r load.packages, include=FALSE}
if (!requireNamespace("BiocManager", quietly = TRUE)) {
 install.packages("BiocManager")
}

if (!require("limma")) {
 BiocManager::install("limma")
}

my_repo <- 'http://cran.r-project.org'
if (!require("pacman")) {
 install.packages("pacman", repos = my_repo)
}

pacman::p_load(
 dplyr, tidyr, readr, knitr, kableExtra, purrr, ggplot2, Hmisc, EnvStats, MKmisc, gridExtra, grid, patchwork, broom, modelr, stringdist, stringr
)

if (!require("gridExtra")) {
 pacman::p_load(gridExtra)
```

```
} else {
  message("gridExtra is installed and loaded successfully.")
}

if (!require("patchwork")) {
  pacman::p_load(patchwork)
} else {
  message("patchwork is installed and loaded successfully.")
}
source("global_functions.R")

library(dplyr)

install.packages("readxl")
library(readxl)
```

```{r load data}
folder_path <- "ITA Data Files"

file_list <- list.files(path = folder_path, pattern = "ITA.*\\.csv", full.names = TRUE)

for (file in file_list) {
  file_year <- gsub(".*ITA\\s|\\.csv", "", file)

  dataset_name <- paste0("ITA_", file_year)
```

```
  assign(dataset_name, read.csv(file))
}

ls(pattern = "ITA_")  # List all objects with names starting with "ITA_"
```

```{r import NTD Data}
years <- 2016:2023

datasets <- list()

for (year in years) {
  file_name <- paste0(year, "AgencyInfo.xlsx")
  datasets[[as.character(year)]] <- read_excel(file_name)
}

data_2016 <- datasets[["2016"]]
data_2017 <- datasets[["2017"]]
data_2018 <- datasets[["2018"]]
data_2019 <- datasets[["2019"]]
data_2020 <- datasets[["2020"]]
data_2021 <- datasets[["2021"]]
data_2022 <- datasets[["2022"]]
data_2023 <- datasets[["2023"]]
```

```{r NAICS Filtering}
library(dplyr)
```

#NAICS 4851 and 4852 classify businesses within the passenger transportation industry. NAICS 4851 focuses on Urban Transit Systems, which include establishments operating local transportation services such as buses, commuter trains, and subways within urban areas. These services typically follow fixed routes and schedules. In contrast, NAICS 4852 pertains to Interurban and Rural Bus Transportation, which involves passenger transportation between cities or rural areas, often covering longer distances. This category includes businesses offering scheduled routes as well as charter bus services.

```
filter_naics <- function(data) {
  data %>%
    filter(
      grepl("4851|4852", naics_code),
      !is.na(industry_description),
      str_trim(industry_description) != ""
    )
}

ITA_2016 <- filter_naics(ITA_2016)
ITA_2017 <- filter_naics(ITA_2017)
ITA_2018 <- filter_naics(ITA_2018)
ITA_2019 <- filter_naics(ITA_2019)
ITA_2020 <- filter_naics(ITA_2020)
ITA_2021 <- filter_naics(ITA_2021)
ITA_2022 <- filter_naics(ITA_2022)
ITA_2023 <- filter_naics(ITA_2023)
```

```{r Agency Name Matching Function}
```

```
update_company_name <- function(ita_data, ntd_data) {
  ita_data <- ita_data %>%
    mutate(
      company_name = ifelse(
        (is.na(company_name) | grepl("^\\d+$", company_name) | company_name == ""),
        ifelse(
          street_address %in% ntd_data$`Address Line 1`,
          ntd_data$`Agency Name`[match(street_address, ntd_data$`Address Line 1`)],
          ifelse(
            street_address %in% ntd_data$`Address Line 2`,
            ntd_data$`Agency Name`[match(street_address, ntd_data$`Address Line 2`)],
            company_name
          )
        ),
        company_name
      )
    )
  return(ita_data)
}

for (year in 2016:2023) {
  ita_var <- paste0("ITA_", year)
  ntd_var <- paste0("data_", year)

  if (exists(ita_var) && exists(ntd_var)) {
    ita_data <- get(ita_var)
```

```
    ntd_data <- get(ntd_var)

    updated_ita_data <- update_company_name(ita_data, ntd_data)

    assign(ita_var, updated_ita_data)
  }
}

ls(pattern = "ITA_")  # List all ITA datasets
```

```{r establishment to company name function}
fill_with_establishment_name <- function(ita_data) {
  ita_data <- ita_data %>%
    mutate(
      company_name = ifelse(
        (is.na(company_name) | company_name == "") &
        grepl("[a-zA-Z]", establishment_name) &
        !grepl("^\\d+$", establishment_name),
        establishment_name,
        company_name
      )
    )
  return(ita_data)
}

for (year in 2016:2023) {
```

```
  ita_var <- paste0("ITA_", year)

  if (exists(ita_var)) {
    ita_data <- get(ita_var)

    updated_ita_data <- fill_with_establishment_name(ita_data)

    assign(ita_var, updated_ita_data)
  }
}

ls(pattern = "ITA_")  # List all ITA datasets
```

```{r clean and count function}
remove_starting_numeric_company_name <- function(ita_data) {
  ita_data <- ita_data %>%
    mutate(
      company_name = ifelse(
        grepl("^\\d", company_name),
        NA,
        company_name
      )
    )
  return(ita_data)
}
```

```
for (year in 2016:2023) {
  ita_var <- paste0("ITA_", year)

  if (exists(ita_var)) {
    ita_data <- get(ita_var)

    updated_ita_data <- remove_starting_numeric_company_name(ita_data)

    assign(ita_var, updated_ita_data)
  }
}
```

```{r address outsourcing}
blank_company_names <- data.frame()

for (year in 2016:2023) {
  ita_var <- paste0("ITA_", year)

  if (exists(ita_var)) {
    ita_data <- get(ita_var)

    colnames(ita_data) <- tolower(gsub("\\s+", "_", colnames(ita_data)))

    blank_rows <- ita_data[is.na(ita_data$company_name) | ita_data$company_name == "", ]

    blank_rows <- blank_rows %>%
```

```
    select(company_name, street_address, city, state, zip_code)

  if (nrow(blank_rows) > 0) {
    blank_rows$year <- year

    blank_company_names <- rbind(blank_company_names, blank_rows)
  }
 }
}

write.csv(blank_company_names, "blank_company_names_filtered.csv", row.names = FALSE)

cat("The cleaned dataset with selected columns has been saved to 'blank_company_names_filtered.csv'\n")

blank_company_names <- blank_company_names %>%
 distinct()

write.csv(blank_company_names, "blank_company_names_filtered.csv", row.names = FALSE)

cat("The deduplicated dataset has been saved and overwrites 'blank_company_names_filtered.csv'\n")
```

```{r address outsource input}
searched_data <- read.csv("blank_company_searched.csv")
```

```
colnames(searched_data) <- tolower(gsub("\\s+", "_", colnames(searched_data)))

searched_data <- searched_data %>% select(street_address, company_name)

for (year in 2016:2023) {
  ita_var <- paste0("ITA_", year)

  if (exists(ita_var)) {
    ita_data <- get(ita_var)

    colnames(ita_data) <- tolower(gsub("\\s+", "_", colnames(ita_data)))

    ita_data <- ita_data %>%
      left_join(searched_data, by = "street_address") %>%
      mutate(
        company_name = ifelse(is.na(company_name.x) | company_name.x == "",
company_name.y, company_name.x)
      ) %>%
      select(-company_name.x, -company_name.y)

    assign(ita_var, ita_data)
  }
}

cat("All ITA datasets have been updated with company_name from
Blank_company_Searched.csv\n")
```

```

```{r final name solution}

clean_company_name <- function(ita_data) {

 ita_data <- ita_data %>%

  mutate(

   company_name = ifelse(tolower(company_name) == "null", "Unknown",
company_name),

   company_name = ifelse(is.na(company_name) | company_name == "",
establishment_id, company_name)

  )

 return(ita_data)

}


for (year in 2016:2023) {

 ita_var <- paste0("ITA_", year)


 if (exists(ita_var)) {

  ita_data <- get(ita_var)


  cleaned_ita_data <- clean_company_name(ita_data)


  assign(ita_var, cleaned_ita_data)

 }

}


cat("All ITA datasets have been updated: 'NULL' values replaced with 'Unknown' and blanks
filled with establishment_id.\n")
```
```

```

```{r final company name check}
remaining_blanks <- data.frame()

for (year in 2016:2023) {
  ita_var <- paste0("ITA_", year)

  if (exists(ita_var)) {
    ita_data <- get(ita_var)

    colnames(ita_data) <- tolower(gsub("\\s+", "_", colnames(ita_data)))

    blank_rows <- ita_data %>%
      filter(is.na(company_name) | company_name == "") %>%
      select(company_name, street_address, city, state, zip_code) %>%  # Keep relevant columns
      mutate(year = year)  # Add year column for reference

    if (nrow(blank_rows) > 0) {
      remaining_blanks <- rbind(remaining_blanks, blank_rows)
    }
  }
}

write.csv(remaining_blanks, "remaining_blank_company_names.csv", row.names = FALSE)
```
```

```
if (nrow(remaining_blanks) == 0) {
  cat("✅ All ITA datasets have been successfully updated. No blank company_name entries remain.\n")
} else {
  cat("⚠️ There are still", nrow(remaining_blanks), "rows with blank company_name. Saved to 'remaining_blank_company_names.csv' for review.\n")
}
```

```{r ITA_2016 combine}
ITA_Normalized_2016 <- ITA_2016 %>%
  mutate(
    normalized_name = str_to_lower(str_trim(company_name)),
    normalized_name = str_replace_all(normalized_name, "[[:punct:]]", "")
  )

similarity_matrix <- stringdistmatrix(
  ITA_Normalized_2016$normalized_name,
  ITA_Normalized_2016$normalized_name,
  method = "jw"
)

threshold <- 0.1

clusters <- cutree(hclust(as.dist(similarity_matrix)), h = threshold)

ITA_Normalized_2016 <- ITA_Normalized_2016 %>%
```

```
  mutate(cluster_id = clusters)


ITA_Merged_2016 <- ITA_Normalized_2016 %>%
  group_by(cluster_id, state) %>%
  summarise(
    company_name = paste(unique(company_name), collapse = "; "),
    id = paste(unique(id), collapse = ", "),
    street_address = paste(unique(street_address), collapse = "; "),
    city = paste(unique(city), collapse = "; "),
    state = first(state),  # Retain the state
    zip_code = paste(unique(zip_code), collapse = ", "),
    naics_code = first(na.omit(naics_code)),
    industry_description = first(na.omit(industry_description)),
    annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
    total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
    no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
    total_deaths = sum(total_deaths, na.rm = TRUE),
    total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
    total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
    total_other_cases = sum(total_other_cases, na.rm = TRUE),
    total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
    total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
    total_injuries = sum(total_injuries, na.rm = TRUE),
    total_poisonings = sum(total_poisonings, na.rm = TRUE),
    total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
    total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
```

```
  total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
  total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
  establishment_id = paste(unique(establishment_id), collapse = ", "),
  establishment_name = paste(unique(establishment_name), collapse = "; "),
  establishment_type = first(na.omit(establishment_type)),
  size = first(na.omit(size)),
  year_filing_for = first(na.omit(year_filing_for)),
  created_timestamp = first(na.omit(created_timestamp)),
  change_reason = first(na.omit(change_reason)),
  .groups = "drop"
 )

cat("✅ ITA_Merged_2016 has been successfully created while preserving ITA_2016.\n")
```

```{r ITA_2017&2018_combine}
merge_ITA_data <- function(ITA_data, year) {
 ITA_Normalized <- ITA_data %>%
  mutate(
   normalized_name = str_to_lower(str_trim(company_name)),
   normalized_name = str_replace_all(normalized_name, "[[:punct:]]", "")
  )

 similarity_matrix <- stringdistmatrix(
  ITA_Normalized$normalized_name,
  ITA_Normalized$normalized_name,
  method = "jw"
```

```
)

threshold <- 0.1

clusters <- cutree(hclust(as.dist(similarity_matrix)), h = threshold)

ITA_Normalized <- ITA_Normalized %>%
  mutate(cluster_id = clusters)

ITA_Merged <- ITA_Normalized %>%
  group_by(cluster_id, state) %>%
  summarise(
    company_name = paste(unique(company_name), collapse = "; "),
    id = paste(unique(id), collapse = ", "),
    street_address = paste(unique(street_address), collapse = "; "),
    city = paste(unique(city), collapse = "; "),
    state = first(state),  # Retain the state
    zip_code = paste(unique(zip_code), collapse = ", "),
    naics_code = first(na.omit(naics_code)),
    industry_description = first(na.omit(industry_description)),
    annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
    total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
    no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
    total_deaths = sum(total_deaths, na.rm = TRUE),
    total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
    total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
```

```
    total_other_cases = sum(total_other_cases, na.rm = TRUE),
    total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
    total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
    total_injuries = sum(total_injuries, na.rm = TRUE),
    total_poisonings = sum(total_poisonings, na.rm = TRUE),
    total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
    total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
    total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
    total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
    establishment_id = paste(unique(establishment_id), collapse = ", "),
    establishment_name = paste(unique(establishment_name), collapse = "; "),
    establishment_type = first(na.omit(establishment_type)),
    size = first(na.omit(size)),
    year_filing_for = first(na.omit(year_filing_for)),
    created_timestamp = first(na.omit(created_timestamp)),
    change_reason = first(na.omit(change_reason)),
    .groups = "drop"
  )

 assign(paste0("ITA_Merged_", year), ITA_Merged, envir = .GlobalEnv)
 cat(paste("✅ ITA_Merged_", year, "has been successfully created while preserving ITA_", year, ".\n", sep=""))
}

# Apply function to ITA_2017 and ITA_2018
merge_ITA_data(ITA_2017, "2017")
```

```
merge_ITA_data(ITA_2018, "2018")
```

```{r ITA 2019 Merging}
normalize_name <- function(name) {
  name <- tolower(trimws(name))
  name <- gsub("[[:punct:]]", "", name)
  return(name)
}

ITA_2019 <- ITA_2019 %>%
  mutate(
    normalized_name = sapply(company_name, normalize_name)
  )

ITA_2019 <- ITA_2019 %>%
  group_by(ein, state) %>%
  mutate(ein_group_id = cur_group_id()) %>%
  ungroup()

merged_data <- list()

for (group in unique(ITA_2019$ein_group_id)) {
  sub_data <- ITA_2019 %>% filter(ein_group_id == group)

  if (nrow(sub_data) > 1) {
```

```
  sim_matrix <- stringdistmatrix(sub_data$normalized_name,
sub_data$normalized_name, method = "jw")


  clusters <- cutree(hclust(as.dist(sim_matrix)), h = 0.1)
  sub_data <- sub_data %>% mutate(cluster_id = clusters)
 } else {
  sub_data <- sub_data %>% mutate(cluster_id = 1)
 }


 merged_sub <- sub_data %>%
  group_by(ein, cluster_id) %>%
  summarise(
   company_name = paste(unique(company_name), collapse = "; "),
   ein = first(ein),
   id = paste(unique(id), collapse = ", "),
   street_address = paste(unique(street_address), collapse = "; "),
   city = paste(unique(city), collapse = "; "),
   state = first(state),
   zip_code = paste(unique(zip_code), collapse = ", "),
   naics_code = first(na.omit(naics_code)),
   industry_description = first(na.omit(industry_description)),
   annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
   total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
   no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
   total_deaths = sum(total_deaths, na.rm = TRUE),
   total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
```

```
    total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
    total_other_cases = sum(total_other_cases, na.rm = TRUE),
    total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
    total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
    total_injuries = sum(total_injuries, na.rm = TRUE),
    total_poisonings = sum(total_poisonings, na.rm = TRUE),
    total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
    total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
    total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
    total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
    establishment_id = paste(unique(establishment_id), collapse = ", "),
    establishment_name = paste(unique(establishment_name), collapse = "; "),
    establishment_type = first(na.omit(establishment_type)),
    size = first(na.omit(size)),
    year_filing_for = first(na.omit(year_filing_for)),
    created_timestamp = first(na.omit(created_timestamp)),
    change_reason = first(na.omit(change_reason)),
    .groups = "drop"
  )

 merged_data[[length(merged_data) + 1]] <- merged_sub
}

ITA_Merged_2019 <- bind_rows(merged_data)

write.csv(ITA_Merged_2019, "ITA_Merged_2019.csv", row.names = FALSE)
```

```
cat("ITA_2019 merging complete. EIN first, then company_name similarity applied (90% threshold). EIN is now included in the final output.")
```

```{r ITA 2020 merging}
normalize_name <- function(name) {
  name <- tolower(trimws(name))  # Convert to lowercase and trim whitespace
  name <- gsub("[[:punct:]]", "", name)  # Remove punctuation
  return(name)
}

ITA_2020 <- ITA_2020 %>%
  mutate(
    normalized_name = sapply(company_name, normalize_name)
  )

ITA_2020 <- ITA_2020 %>%
  group_by(ein, state) %>%
  mutate(ein_group_id = cur_group_id()) %>%
  ungroup()

merged_data <- list()

for (group in unique(ITA_2020$ein_group_id)) {
  sub_data <- ITA_2020 %>% filter(ein_group_id == group)
```

```
 if (nrow(sub_data) > 1) {
  sim_matrix <- stringdistmatrix(sub_data$normalized_name,
sub_data$normalized_name, method = "jw")

  clusters <- cutree(hclust(as.dist(sim_matrix)), h = 0.1)
  sub_data <- sub_data %>% mutate(cluster_id = clusters)
 } else {
  sub_data <- sub_data %>% mutate(cluster_id = 1)
 }

 merged_sub <- sub_data %>%
  group_by(ein, cluster_id) %>%
  summarise(
   company_name = paste(unique(company_name), collapse = "; "),
   ein = first(ein),
   id = paste(unique(id), collapse = ", "),
   street_address = paste(unique(street_address), collapse = "; "),
   city = paste(unique(city), collapse = "; "),
   state = first(state),
   zip_code = paste(unique(zip_code), collapse = ", "),
   naics_code = first(na.omit(naics_code)),
   industry_description = first(na.omit(industry_description)),
   annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
   total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
   no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
   total_deaths = sum(total_deaths, na.rm = TRUE),
```

```
    total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
    total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
    total_other_cases = sum(total_other_cases, na.rm = TRUE),
    total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
    total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
    total_injuries = sum(total_injuries, na.rm = TRUE),
    total_poisonings = sum(total_poisonings, na.rm = TRUE),
    total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
    total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
    total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
    total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
    establishment_id = paste(unique(establishment_id), collapse = ", "),
    establishment_name = paste(unique(establishment_name), collapse = "; "),
    establishment_type = first(na.omit(establishment_type)),
    size = first(na.omit(size)),
    year_filing_for = first(na.omit(year_filing_for)),
    created_timestamp = first(na.omit(created_timestamp)),
    change_reason = first(na.omit(change_reason)),
    .groups = "drop"
  )

 merged_data[[length(merged_data) + 1]] <- merged_sub
}

ITA_Merged_2020 <- bind_rows(merged_data)
```

```
write.csv(ITA_Merged_2020, "ITA_Merged_2020.csv", row.names = FALSE)

cat("ITA_2020 merging complete. EIN first, then company_name similarity applied (90% threshold). EIN is now included in the final output.")
```

```{r ITA 2021 Merging}
normalize_name <- function(name) {
  name <- tolower(trimws(name))
  name <- gsub("[[:punct:]]", "", name)
  return(name)
}

ITA_2021 <- ITA_2021 %>%
  mutate(
    normalized_name = sapply(company_name, normalize_name)
  )

ITA_2021 <- ITA_2021 %>%
  group_by(ein, state) %>%
  mutate(ein_group_id = cur_group_id()) %>%
  ungroup()

merged_data <- list()

for (group in unique(ITA_2021$ein_group_id)) {
```

```
 sub_data <- ITA_2021 %>% filter(ein_group_id == group)


 if (nrow(sub_data) > 1) {
  sim_matrix <- stringdistmatrix(sub_data$normalized_name,
sub_data$normalized_name, method = "jw")


  clusters <- cutree(hclust(as.dist(sim_matrix)), h = 0.1)
  sub_data <- sub_data %>% mutate(cluster_id = clusters)
 } else {
  sub_data <- sub_data %>% mutate(cluster_id = 1)
 }


 merged_sub <- sub_data %>%
  group_by(ein, cluster_id) %>%
  summarise(
   company_name = paste(unique(company_name), collapse = "; "),
   ein = first(ein),
   id = paste(unique(id), collapse = ", "),
   street_address = paste(unique(street_address), collapse = "; "),
   city = paste(unique(city), collapse = "; "),
   state = first(state),
   zip_code = paste(unique(zip_code), collapse = ", "),
   naics_code = first(na.omit(naics_code)),
   industry_description = first(na.omit(industry_description)),
   annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
   total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
```

```
    no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
    total_deaths = sum(total_deaths, na.rm = TRUE),
    total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
    total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
    total_other_cases = sum(total_other_cases, na.rm = TRUE),
    total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
    total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
    total_injuries = sum(total_injuries, na.rm = TRUE),
    total_poisonings = sum(total_poisonings, na.rm = TRUE),
    total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
    total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
    total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
    total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
    establishment_id = paste(unique(establishment_id), collapse = ", "),
    establishment_name = paste(unique(establishment_name), collapse = "; "),
    establishment_type = first(na.omit(establishment_type)),
    size = first(na.omit(size)),
    year_filing_for = first(na.omit(year_filing_for)),
    created_timestamp = first(na.omit(created_timestamp)),
    change_reason = first(na.omit(change_reason)),
    .groups = "drop"
  )

 merged_data[[length(merged_data) + 1]] <- merged_sub
}
```

```
ITA_Merged_2021 <- bind_rows(merged_data)


write.csv(ITA_Merged_2021, "ITA_Merged_2021.csv", row.names = FALSE)


cat("ITA_2021 merging complete. EIN first, then company_name similarity applied (90% threshold). EIN is now included in the final output.")
```

```{r ITA 2022 Merging}
normalize_name <- function(name) {
  name <- tolower(trimws(name))
  name <- gsub("[[:punct:]]", "", name)
  return(name)
}

ITA_2022 <- ITA_2022 %>%
  mutate(
    normalized_name = sapply(company_name, normalize_name)
  )

ITA_2022 <- ITA_2022 %>%
  group_by(ein, state) %>%
  mutate(ein_group_id = cur_group_id()) %>%
  ungroup()
```

```
merged_data <- list()

for (group in unique(ITA_2022$ein_group_id)) {
  sub_data <- ITA_2022 %>% filter(ein_group_id == group)

  if (nrow(sub_data) > 1) {
    sim_matrix <- stringdistmatrix(sub_data$normalized_name,
sub_data$normalized_name, method = "jw")

    clusters <- cutree(hclust(as.dist(sim_matrix)), h = 0.1)
    sub_data <- sub_data %>% mutate(cluster_id = clusters)
  } else {
    sub_data <- sub_data %>% mutate(cluster_id = 1)
  }

  merged_sub <- sub_data %>%
    group_by(ein, cluster_id) %>%
    summarise(
      company_name = paste(unique(company_name), collapse = "; "),
      ein = first(ein),
      id = paste(unique(id), collapse = ", "),
      street_address = paste(unique(street_address), collapse = "; "),
      city = paste(unique(city), collapse = "; "),
      state = first(state),
      zip_code = paste(unique(zip_code), collapse = ", "),
      naics_code = first(na.omit(naics_code)),
```

```
  industry_description = first(na.omit(industry_description)),
  annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
  total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
  no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
  total_deaths = sum(total_deaths, na.rm = TRUE),
  total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
  total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
  total_other_cases = sum(total_other_cases, na.rm = TRUE),
  total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
  total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
  total_injuries = sum(total_injuries, na.rm = TRUE),
  total_poisonings = sum(total_poisonings, na.rm = TRUE),
  total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
  total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
  total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
  total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
  establishment_id = paste(unique(establishment_id), collapse = ", "),
  establishment_name = paste(unique(establishment_name), collapse = "; "),
  establishment_type = first(na.omit(establishment_type)),
  size = first(na.omit(size)),
  year_filing_for = first(na.omit(year_filing_for)),
  created_timestamp = first(na.omit(created_timestamp)),
  change_reason = first(na.omit(change_reason)),
  .groups = "drop"
)
```

```
  merged_data[[length(merged_data) + 1]] <- merged_sub
}

ITA_Merged_2022 <- bind_rows(merged_data)

write.csv(ITA_Merged_2022, "ITA_Merged_2022.csv", row.names = FALSE)

cat("ITA_2022 merging complete. EIN first, then company_name similarity applied (90% threshold). EIN is now included in the final output.")
```

```{r ITA 2023 Merging}
normalize_name <- function(name) {
  name <- tolower(trimws(name))
  name <- gsub("[[:punct:]]", "", name)
  return(name)
}

ITA_2023 <- ITA_2023 %>%
  mutate(
    normalized_name = sapply(company_name, normalize_name)
  )

ITA_2023 <- ITA_2023 %>%
  group_by(ein, state) %>%
  summarise(
    company_name = paste(unique(company_name), collapse = "; "),
```

```
ein = first(ein),
id = paste(unique(id), collapse = ", "),
street_address = paste(unique(street_address), collapse = "; "),
city = paste(unique(city), collapse = "; "),
state = first(state),
zip_code = paste(unique(zip_code), collapse = ", "),
naics_code = first(na.omit(naics_code)),
industry_description = first(na.omit(industry_description)),
annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
total_deaths = sum(total_deaths, na.rm = TRUE),
total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
total_other_cases = sum(total_other_cases, na.rm = TRUE),
total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
total_injuries = sum(total_injuries, na.rm = TRUE),
total_poisonings = sum(total_poisonings, na.rm = TRUE),
total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
establishment_id = paste(unique(establishment_id), collapse = ", "),
establishment_name = paste(unique(establishment_name), collapse = "; "),
establishment_type = first(na.omit(establishment_type)),
```

```
  size = first(na.omit(size)),
  year_filing_for = first(na.omit(year_filing_for)),
  created_timestamp = first(na.omit(created_timestamp)),
  change_reason = first(na.omit(change_reason)),
  .groups = "drop"
 )

merged_data <- list()

for (group in unique(ITA_2023$ein)) {
 sub_data <- ITA_2023 %>% filter(ein == group)

 if (nrow(sub_data) > 1) {
  sim_matrix <- stringdistmatrix(sub_data$company_name, sub_data$company_name,
method = "jw")

  clusters <- cutree(hclust(as.dist(sim_matrix)), h = 0.1)
  sub_data <- sub_data %>% mutate(cluster_id = clusters)
 } else {
  sub_data <- sub_data %>% mutate(cluster_id = 1)
 }

 merged_sub <- sub_data %>%
  group_by(ein, cluster_id) %>%
  summarise(
   company_name = paste(unique(company_name), collapse = "; "),
```

```
ein = first(ein),  # Retain EIN
id = paste(unique(id), collapse = ", "),
street_address = paste(unique(street_address), collapse = "; "),
city = paste(unique(city), collapse = "; "),
state = first(state),
zip_code = paste(unique(zip_code), collapse = ", "),
naics_code = first(na.omit(naics_code)),
industry_description = first(na.omit(industry_description)),
annual_average_employees = sum(annual_average_employees, na.rm = TRUE),
total_hours_worked = sum(total_hours_worked, na.rm = TRUE),
no_injuries_illnesses = sum(no_injuries_illnesses, na.rm = TRUE),
total_deaths = sum(total_deaths, na.rm = TRUE),
total_dafw_cases = sum(total_dafw_cases, na.rm = TRUE),
total_djtr_cases = sum(total_djtr_cases, na.rm = TRUE),
total_other_cases = sum(total_other_cases, na.rm = TRUE),
total_dafw_days = sum(total_dafw_days, na.rm = TRUE),
total_djtr_days = sum(total_djtr_days, na.rm = TRUE),
total_injuries = sum(total_injuries, na.rm = TRUE),
total_poisonings = sum(total_poisonings, na.rm = TRUE),
total_respiratory_conditions = sum(total_respiratory_conditions, na.rm = TRUE),
total_skin_disorders = sum(total_skin_disorders, na.rm = TRUE),
total_hearing_loss = sum(total_hearing_loss, na.rm = TRUE),
total_other_illnesses = sum(total_other_illnesses, na.rm = TRUE),
establishment_id = paste(unique(establishment_id), collapse = ", "),
establishment_name = paste(unique(establishment_name), collapse = "; "),
establishment_type = first(na.omit(establishment_type)),
```

```
    size = first(na.omit(size)),
    year_filing_for = first(na.omit(year_filing_for)),
    created_timestamp = first(na.omit(created_timestamp)),
    change_reason = first(na.omit(change_reason)),
    .groups = "drop"
  )

 merged_data[[length(merged_data) + 1]] <- merged_sub
}

ITA_Merged_2023 <- bind_rows(merged_data)

write.csv(ITA_Merged_2023, "ITA_Merged_2023.csv", row.names = FALSE)

cat("ITA_2023 merging complete. First merged by EIN, then applied company_name similarity clustering (90% threshold). EIN is now included in the final output.")
```

```{r FTE}
calculate_FTE <- function(data) {
 data %>%
  mutate(
    FTE = total_hours_worked / 2000,

    rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
    rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
    rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
```

```
    rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
    rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
    rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
    rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
    rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,

    rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
  ) %>%
  replace(is.na(.), 0)
}

for (year in 2016:2023) {
 ita_var <- paste0("ITA_Merged_", year)

 if (exists(ita_var)) {
  ita_data <- get(ita_var)

  updated_ita_data <- calculate_FTE(ita_data) %>%
   filter(total_hours_worked > 0)

  assign(ita_var, updated_ita_data)
 }
}

cat("✅ FTE and rate calculations per 100 FTE and per 10,000 FTE (for deaths) have been successfully applied to all years.\n")
```

```

```{r FTE rates plots}
rates_fte_2016_2023 <- bind_rows(
 ITA_Merged_2016 %>% mutate(year = 2016,
                            establishment_type = as.character(establishment_type),
                            ein = NA_character_),  # EIN was missing in early years, set to NA
 ITA_Merged_2017 %>% mutate(year = 2017,
                            establishment_type = as.character(establishment_type),
                            ein = NA_character_),
 ITA_Merged_2018 %>% mutate(year = 2018,
                            establishment_type = as.character(establishment_type),
                            ein = NA_character_),
 ITA_Merged_2019 %>% mutate(year = 2019,
                            establishment_type = as.character(establishment_type),
                            ein = as.character(ein)),  # Convert EIN to character
 ITA_Merged_2020 %>% mutate(year = 2020,
                            establishment_type = as.character(establishment_type),
                            ein = as.character(ein)),
 ITA_Merged_2021 %>% mutate(year = 2021,
                            establishment_type = as.character(establishment_type),
                            ein = as.character(ein)),
 ITA_Merged_2022 %>% mutate(year = 2022,
                            establishment_type = as.character(establishment_type),
                            ein = as.character(ein)),
 ITA_Merged_2023 %>% mutate(year = 2023,

```
                establishment_type = as.character(establishment_type),
                ein = as.character(ein))
)

rates_fte_2016_2023 <- rates_fte_2016_2023 %>%
  select(year, rate_per_100_FTE_hearing_loss, rate_per_100_FTE_poisonings,
       rate_per_100_FTE_respiratory, rate_per_100_FTE_skin_disorders,
       rate_per_100_FTE_dafw_cases, rate_per_100_FTE_injuries,
       rate_per_100_FTE_other_cases, rate_per_100_FTE_other_illnesses,
       rate_per_10k_FTE_deaths)

rates_fte_long <- rates_fte_2016_2023 %>%
  pivot_longer(cols = -year, names_to = "Illness_Type", values_to = "Rate")

rates_fte_long$Illness_Type <- recode(rates_fte_long$Illness_Type,
  "rate_per_100_FTE_hearing_loss" = "Hearing Loss",
  "rate_per_100_FTE_poisonings" = "Poisonings",
  "rate_per_100_FTE_respiratory" = "Respiratory Disorders",
  "rate_per_100_FTE_skin_disorders" = "Skin Disorders",
  "rate_per_100_FTE_dafw_cases" = "DAFW Cases",
  "rate_per_100_FTE_injuries" = "Injuries",
  "rate_per_100_FTE_other_cases" = "Other Cases",
  "rate_per_100_FTE_other_illnesses" = "Other Illnesses",
  "rate_per_10k_FTE_deaths" = "Deaths (per 10k FTE)"
)
```

```
ggplot(rates_fte_long, aes(x = year, y = Rate, color = Illness_Type)) +
  geom_line(size = 1.2) +
  geom_smooth(se = FALSE, method = "loess", linetype = "dashed", size = 0.8, alpha = 0.7) +
# Smooth trend
  facet_wrap(~Illness_Type, scales = "free_y") +
  scale_y_log10() +
  labs(title = "OSHA-Reportable Illness & Injury Rates per 100 FTE (Log Scale, 2016-2023)",
       x = "Year", y = "Rate per 100 FTE (Log Scale)", color = "Illness Type") +
  theme_minimal() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 45, hjust = 1))
```

```{r ITA 2016 Rates (FTE Only)}
fta_region_map <- c(
  "WA" = 10, "OR" = 10, "CA" = 9, "NV" = 9, "ID" = 10, "MT" = 8, "AK" = 10, "HI" = 9,
  "CO" = 8, "UT" = 8, "WY" = 8, "AZ" = 9, "NM" = 6, "TX" = 6, "OK" = 6,
  "ND" = 8, "SD" = 8, "MN" = 5, "WI" = 5, "IL" = 5, "IN" = 5, "MO" = 7, "MI" = 5, "OH" = 5,
  "KY" = 4, "WV" = 3, "VA" = 3, "PA" = 3, "NY" = 2, "NJ" = 2, "ME" = 1, "NH" = 1, "VT" = 1,
  "MA" = 1, "CT" = 1, "RI" = 1, "DE" = 3, "MD" = 3, "DC" = 3, "NC" = 4, "SC" = 4, "GA" = 4,
  "FL" = 4, "TN" = 4, "AL" = 4, "MS" = 4, "AR" = 6, "LA" = 6,
  "IA" = 7, "KS" = 7, "NE" = 7,
  "PR" = 4
)

ITA_Merged_2016 <- ITA_Merged_2016 %>%
```

```
  mutate(fta_region = fta_region_map[state])

rates_2016 <- ITA_Merged_2016 %>%
  filter(total_hours_worked > 0) %>%
  mutate(
    FTE = total_hours_worked / 2000,

    rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
    rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
    rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
    rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
    rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
    rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
    rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
    rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,

    rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
  ) %>%
  replace(is.na(.), 0)

summary_state_2016 <- rates_2016 %>%
  group_by(state) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2016 <- rates_2016 %>%
  group_by(fta_region) %>%
```

```
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2016 <- rates_2016 %>%
  group_by(size) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_establishment_type_2016 <- rates_2016 %>%
  group_by(establishment_type) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2016 <- rates_2016 %>%
  group_by(naics_code) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```{r ITA 2017 rates}
ITA_Merged_2017 <- ITA_Merged_2017 %>%
  mutate(fta_region = fta_region_map[state])

missing_states <- ITA_Merged_2017 %>%
  filter(is.na(fta_region)) %>%
  distinct(state)
print(missing_states)

rates_2017 <- ITA_Merged_2017 %>%
```

```
 filter(total_hours_worked > 0) %>%
 mutate(
  FTE = total_hours_worked / 2000,

  rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
  rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
  rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
  rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
  rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
  rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
  rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
  rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,

  rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
 ) %>%
 replace(is.na(.), 0)

summary_state_2017 <- rates_2017 %>%
 group_by(state) %>%
 summarise(across(starts_with("rate_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2017 <- rates_2017 %>%
 group_by(fta_region) %>%
 summarise(across(starts_with("rate_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2017 <- rates_2017 %>%
```

```
  group_by(size) %>%
  summarise(across(starts_with("rate_"), mean, na.rm = TRUE), .groups = "drop")

summary_establishment_type_2017 <- rates_2017 %>%
  group_by(establishment_type) %>%
  summarise(across(starts_with("rate_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2017 <- rates_2017 %>%
  group_by(naics_code) %>%
  summarise(across(starts_with("rate_"), mean, na.rm = TRUE), .groups = "drop")

```

```{r ITA 2018 Rates (FTE Only)}
ITA_Merged_2018 <- ITA_Merged_2018 %>%
  mutate(fta_region = fta_region_map[state])

missing_states <- ITA_Merged_2018 %>%
  filter(is.na(fta_region)) %>%
  distinct(state)
print(missing_states)

rates_2018 <- ITA_Merged_2018 %>%
  filter(total_hours_worked > 0) %>%
  mutate(
    FTE = total_hours_worked / 2000,
```

```
  rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
  rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
  rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
  rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
  rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
  rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
  rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
  rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,

  rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
 ) %>%
 replace(is.na(.), 0)


summary_state_2018 <- rates_2018 %>%
 group_by(state) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2018 <- rates_2018 %>%
 group_by(fta_region) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2018 <- rates_2018 %>%
 group_by(size) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```
summary_establishment_type_2018 <- rates_2018 %>%
  group_by(establishment_type) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2018 <- rates_2018 %>%
  group_by(naics_code) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```{r ITA 2019 Rates (FTE Only)}

ITA_Merged_2019 <- ITA_Merged_2019 %>%
  mutate(fta_region = fta_region_map[state])

missing_states <- ITA_Merged_2019 %>%
  filter(is.na(fta_region)) %>%
  distinct(state)
print(missing_states)



rates_2019 <- ITA_Merged_2019 %>%
  filter(total_hours_worked > 0) %>%
  mutate(
    FTE = total_hours_worked / 2000,
```

```
  rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
  rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
  rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
  rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
  rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
  rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
  rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
  rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,

  rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
) %>%
 replace(is.na(.), 0)



summary_state_2019 <- rates_2019 %>%
 group_by(state) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2019 <- rates_2019 %>%
 group_by(fta_region) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2019 <- rates_2019 %>%
 group_by(size) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```
summary_establishment_type_2019 <- rates_2019 %>%
  group_by(establishment_type) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2019 <- rates_2019 %>%
  group_by(naics_code) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```{r ITA 2020 Rates (FTE Only)}

ITA_Merged_2020 <- ITA_Merged_2020 %>%
  mutate(fta_region = fta_region_map[state])

missing_states <- ITA_Merged_2020 %>%
  filter(is.na(fta_region)) %>%
  distinct(state)
print(missing_states)



rates_2020 <- ITA_Merged_2020 %>%
  filter(total_hours_worked > 0) %>%
  mutate(
    FTE = total_hours_worked / 2000,

    rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
```

```
  rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
  rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
  rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
  rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
  rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
  rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
  rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,
  rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
 ) %>%
 replace(is.na(.), 0)


summary_state_2020 <- rates_2020 %>%
 group_by(state) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2020 <- rates_2020 %>%
 group_by(fta_region) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2020 <- rates_2020 %>%
 group_by(size) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_establishment_type_2020 <- rates_2020 %>%
 group_by(establishment_type) %>%
```

```
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2020 <- rates_2020 %>%
  group_by(naics_code) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```{r ITA 2021 Rates (FTE Only)}

ITA_Merged_2021 <- ITA_Merged_2021 %>%
  mutate(fta_region = fta_region_map[state])

missing_states <- ITA_Merged_2021 %>%
  filter(is.na(fta_region)) %>%
  distinct(state)
print(missing_states)

rates_2021 <- ITA_Merged_2021 %>%
  filter(total_hours_worked > 0) %>%
  mutate(
    FTE = total_hours_worked / 2000,

    rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
    rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
    rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
    rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
```

```
  rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
  rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
  rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
  rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,
  rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
) %>%
 replace(is.na(.), 0)

summary_state_2021 <- rates_2021 %>%
 group_by(state) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2021 <- rates_2021 %>%
 group_by(fta_region) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2021 <- rates_2021 %>%
 group_by(size) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_establishment_type_2021 <- rates_2021 %>%
 group_by(establishment_type) %>%
 summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2021 <- rates_2021 %>%
 group_by(naics_code) %>%
```

```
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```{r ITA 2022 Rates (FTE Only)}
ITA_Merged_2022 <- ITA_Merged_2022 %>%
  mutate(fta_region = fta_region_map[state])

missing_states <- ITA_Merged_2022 %>%
  filter(is.na(fta_region)) %>%
  distinct(state)
print(missing_states)

rates_2022 <- ITA_Merged_2022 %>%
  filter(total_hours_worked > 0) %>%
  mutate(
    FTE = total_hours_worked / 2000,

    rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
    rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
    rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
    rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
    rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
    rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
    rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
    rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,
    rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
```

```
  ) %>%
  replace(is.na(.), 0)

summary_state_2022 <- rates_2022 %>%
  group_by(state) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2022 <- rates_2022 %>%
  group_by(fta_region) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2022 <- rates_2022 %>%
  group_by(size) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_establishment_type_2022 <- rates_2022 %>%
  group_by(establishment_type) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2022 <- rates_2022 %>%
  group_by(naics_code) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")
```

```{r ITA 2023 Rates (FTE Only, Clean Size)}
```

```
ITA_Merged_2023 <- ITA_Merged_2023 %>%
  mutate(fta_region = fta_region_map[state])

missing_states <- ITA_Merged_2023 %>%
  filter(is.na(fta_region)) %>%
  distinct(state)
print(missing_states)



rates_2023 <- ITA_Merged_2023 %>%
  filter(total_hours_worked > 0) %>%
  mutate(
    FTE = total_hours_worked / 2000,

    rate_per_100_FTE_hearing_loss = (total_hearing_loss / FTE) * 100,
    rate_per_100_FTE_poisonings = (total_poisonings / FTE) * 100,
    rate_per_100_FTE_respiratory = (total_respiratory_conditions / FTE) * 100,
    rate_per_100_FTE_skin_disorders = (total_skin_disorders / FTE) * 100,
    rate_per_100_FTE_dafw_cases = (total_dafw_cases / FTE) * 100,
    rate_per_100_FTE_injuries = (total_injuries / FTE) * 100,
    rate_per_100_FTE_other_cases = (total_other_cases / FTE) * 100,
    rate_per_100_FTE_other_illnesses = (total_other_illnesses / FTE) * 100,
    rate_per_10k_FTE_deaths = (total_deaths / FTE) * 10000
  ) %>%
  replace(is.na(.), 0)
```

```
summary_state_2023 <- rates_2023 %>%
  group_by(state) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_fta_region_2023 <- rates_2023 %>%
  group_by(fta_region) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_size_2023 <- rates_2023 %>%
  group_by(size) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_establishment_type_2023 <- rates_2023 %>%
  group_by(establishment_type) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

summary_naics_2023 <- rates_2023 %>%
  group_by(naics_code) %>%
  summarise(across(starts_with("rate_per_"), mean, na.rm = TRUE), .groups = "drop")

```

```{r time series test 2016-2023 (FTE Only)}

time_series_naics <- bind_rows(
  summary_naics_2016 %>% mutate(year = 2016),
```

```
  summary_naics_2017 %>% mutate(year = 2017),
  summary_naics_2018 %>% mutate(year = 2018),
  summary_naics_2019 %>% mutate(year = 2019),
  summary_naics_2020 %>% mutate(year = 2020),
  summary_naics_2021 %>% mutate(year = 2021),
  summary_naics_2022 %>% mutate(year = 2022),
  summary_naics_2023 %>% mutate(year = 2023)
)

time_series_establishment <- bind_rows(
  summary_establishment_type_2016 %>% mutate(year = 2016, establishment_type = as.character(establishment_type)),
  summary_establishment_type_2017 %>% mutate(year = 2017, establishment_type = as.character(establishment_type)),
  summary_establishment_type_2018 %>% mutate(year = 2018, establishment_type = as.character(establishment_type)),
  summary_establishment_type_2019 %>% mutate(year = 2019, establishment_type = as.character(establishment_type)),
  summary_establishment_type_2020 %>% mutate(year = 2020, establishment_type = as.character(establishment_type)),
  summary_establishment_type_2021 %>% mutate(year = 2021, establishment_type = as.character(establishment_type)),
  summary_establishment_type_2022 %>% mutate(year = 2022, establishment_type = as.character(establishment_type)),
  summary_establishment_type_2023 %>% mutate(year = 2023, establishment_type = as.character(establishment_type))
)
```

```
time_series_fta <- bind_rows(
  summary_fta_region_2016 %>% mutate(year = 2016),
  summary_fta_region_2017 %>% mutate(year = 2017),
  summary_fta_region_2018 %>% mutate(year = 2018),
  summary_fta_region_2019 %>% mutate(year = 2019),
  summary_fta_region_2020 %>% mutate(year = 2020),
  summary_fta_region_2021 %>% mutate(year = 2021),
  summary_fta_region_2022 %>% mutate(year = 2022),
  summary_fta_region_2023 %>% mutate(year = 2023)
)

time_series_size <- bind_rows(
  summary_size_2016 %>% mutate(year = 2016, size = as.character(size)),
  summary_size_2017 %>% mutate(year = 2017, size = as.character(size)),
  summary_size_2018 %>% mutate(year = 2018, size = as.character(size)),
  summary_size_2019 %>% mutate(year = 2019, size = as.character(size)),
  summary_size_2020 %>% mutate(year = 2020, size = as.character(size)),
  summary_size_2021 %>% mutate(year = 2021, size = as.character(size)),
  summary_size_2022 %>% mutate(year = 2022, size = as.character(size)),
  summary_size_2023 %>% mutate(year = 2023, size = as.character(size))
)

time_series_naics_long <- time_series_naics %>%
  pivot_longer(cols = starts_with("rate_per_"), names_to = "illness_type", values_to = "rate")

time_series_establishment_long <- time_series_establishment %>%
```

```
  pivot_longer(cols = starts_with("rate_per_"), names_to = "illness_type", values_to = "rate")

time_series_fta_long <- time_series_fta %>%
  pivot_longer(cols = starts_with("rate_per_"), names_to = "illness_type", values_to = "rate")

time_series_size_long <- time_series_size %>%
  pivot_longer(cols = starts_with("rate_per_"), names_to = "illness_type", values_to = "rate")

ggplot(time_series_naics_long, aes(x = year, y = rate, color = factor(naics_code))) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  facet_wrap(~illness_type, scales = "free_y") +
  labs(title = "FTE-Adjusted Illness & Injury Rates by NAICS Code (2016–2023)",
       x = "Year", y = "Rate per 100 FTE (or per 10k for deaths)", color = "NAICS Code") +
  theme_minimal() +
  theme(legend.position = "bottom")

ggplot(time_series_establishment_long, aes(x = year, y = rate, color =
factor(establishment_type))) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  facet_wrap(~illness_type, scales = "free_y") +
  labs(title = "FTE-Adjusted Illness & Injury Rates by Establishment Type (2016–2023)",
       x = "Year", y = "Rate per 100 FTE (or per 10k for deaths)", color = "Establishment Type") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
ggplot(time_series_fta_long, aes(x = year, y = rate, color = factor(fta_region))) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  facet_wrap(~illness_type, scales = "free_y") +
  labs(title = "FTE-Adjusted Illness & Injury Rates by FTA Region (2016–2023)",
       x = "Year", y = "Rate per 100 FTE (or per 10k for deaths)", color = "FTA Region") +
  theme_minimal() +
  theme(legend.position = "bottom")

ggplot(time_series_size_long, aes(x = year, y = rate, color = factor(size))) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  facet_wrap(~illness_type, scales = "free_y") +
  labs(title = "FTE-Adjusted Illness & Injury Rates by Establishment Size (2016–2023)",
       x = "Year", y = "Rate per 100 FTE (or per 10k for deaths)", color = "Establishment Size") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```{r time series test (FTE Only)}
rates_2016_2023 <- bind_rows(
  rates_2016 %>% mutate(year = 2016, establishment_type = as.character(establishment_type), ein = NA_character_),
  rates_2017 %>% mutate(year = 2017, establishment_type = as.character(establishment_type), ein = NA_character_),
```

```
  rates_2018 %>% mutate(year = 2018, establishment_type =
as.character(establishment_type), ein = NA_character_),

  rates_2019 %>% mutate(year = 2019, establishment_type =
as.character(establishment_type), ein = as.character(ein)),

  rates_2020 %>% mutate(year = 2020, establishment_type =
as.character(establishment_type), ein = as.character(ein)),

  rates_2021 %>% mutate(year = 2021, establishment_type =
as.character(establishment_type), ein = as.character(ein)),

  rates_2022 %>% mutate(year = 2022, establishment_type =
as.character(establishment_type), ein = as.character(ein)),

  rates_2023 %>%

    mutate(year = 2023, establishment_type = as.character(establishment_type), ein =
as.character(ein))

)


str(rates_2016_2023)

head(rates_2016_2023)
```

```{r time series creation (FTE Only)}

hearing_loss_ts <- ts(

  rates_2016_2023 %>%

    group_by(year) %>%

    summarise(rate = mean(rate_per_100_FTE_hearing_loss, na.rm = TRUE)) %>%

    pull(rate),

  start = 2016, end = 2023, frequency = 1

)


respiratory_ts <- ts(
```

```
  rates_2016_2023 %>%
    group_by(year) %>%
    summarise(rate = mean(rate_per_100_FTE_respiratory, na.rm = TRUE)) %>%
    pull(rate),
  start = 2016, end = 2023, frequency = 1
)

skin_disorders_ts <- ts(
  rates_2016_2023 %>%
    group_by(year) %>%
    summarise(rate = mean(rate_per_100_FTE_skin_disorders, na.rm = TRUE)) %>%
    pull(rate),
  start = 2016, end = 2023, frequency = 1
)

poisonings_ts <- ts(
  rates_2016_2023 %>%
    group_by(year) %>%
    summarise(rate = mean(rate_per_100_FTE_poisonings, na.rm = TRUE)) %>%
    pull(rate),
  start = 2016, end = 2023, frequency = 1
)

dafw_cases_ts <- ts(
  rates_2016_2023 %>%
    group_by(year) %>%
```

```
    summarise(rate = mean(rate_per_100_FTE_dafw_cases, na.rm = TRUE)) %>%
    pull(rate),
  start = 2016, end = 2023, frequency = 1
)

rate_deaths_ts <- ts(
  rates_2016_2023 %>%
    group_by(year) %>%
    summarise(rate = mean(rate_per_10k_FTE_deaths, na.rm = TRUE)) %>%
    pull(rate),
  start = 2016, end = 2023, frequency = 1
)

rate_injuries_ts <- ts(
  rates_2016_2023 %>%
    group_by(year) %>%
    summarise(rate = mean(rate_per_100_FTE_injuries, na.rm = TRUE)) %>%
    pull(rate),
  start = 2016, end = 2023, frequency = 1
)

rate_other_cases_ts <- ts(
  rates_2016_2023 %>%
    group_by(year) %>%
    summarise(rate = mean(rate_per_100_FTE_other_cases, na.rm = TRUE)) %>%
    pull(rate),
```

```
  start = 2016, end = 2023, frequency = 1
)

rate_other_illnesses_ts <- ts(
  rates_2016_2023 %>%
    group_by(year) %>%
    summarise(rate = mean(rate_per_100_FTE_other_illnesses, na.rm = TRUE)) %>%
    pull(rate),
  start = 2016, end = 2023, frequency = 1
)

print(hearing_loss_ts)
print(respiratory_ts)
print(skin_disorders_ts)
print(poisonings_ts)
print(dafw_cases_ts)
print(rate_deaths_ts)
print(rate_injuries_ts)
print(rate_other_cases_ts)
print(rate_other_illnesses_ts)

```

```{r Year-over-Year Percentage Change (FTE Only)}
rates_2016_2023 <- rates_2016_2023 %>%
  arrange(year, state, establishment_type, fta_region, naics_code) %>%
  group_by(state, establishment_type, fta_region, naics_code) %>%
```

```
  mutate(

    yoy_hearing_loss = ifelse(is.na(lag(rate_per_100_FTE_hearing_loss)) |
lag(rate_per_100_FTE_hearing_loss) == 0, NA,

                              (rate_per_100_FTE_hearing_loss - lag(rate_per_100_FTE_hearing_loss)) /
lag(rate_per_100_FTE_hearing_loss) * 100),


    yoy_respiratory = ifelse(is.na(lag(rate_per_100_FTE_respiratory)) |
lag(rate_per_100_FTE_respiratory) == 0, NA,

                             (rate_per_100_FTE_respiratory - lag(rate_per_100_FTE_respiratory)) /
lag(rate_per_100_FTE_respiratory) * 100),


    yoy_skin_disorders = ifelse(is.na(lag(rate_per_100_FTE_skin_disorders)) |
lag(rate_per_100_FTE_skin_disorders) == 0, NA,

                                (rate_per_100_FTE_skin_disorders - lag(rate_per_100_FTE_skin_disorders)) /
lag(rate_per_100_FTE_skin_disorders) * 100),


    yoy_poisonings = ifelse(is.na(lag(rate_per_100_FTE_poisonings)) |
lag(rate_per_100_FTE_poisonings) == 0, NA,

                            (rate_per_100_FTE_poisonings - lag(rate_per_100_FTE_poisonings)) /
lag(rate_per_100_FTE_poisonings) * 100),


    yoy_dafw_cases = ifelse(is.na(lag(rate_per_100_FTE_dafw_cases)) |
lag(rate_per_100_FTE_dafw_cases) == 0, NA,

                            (rate_per_100_FTE_dafw_cases - lag(rate_per_100_FTE_dafw_cases)) /
lag(rate_per_100_FTE_dafw_cases) * 100),


    yoy_rate_deaths = ifelse(is.na(lag(rate_per_10k_FTE_deaths)) |
lag(rate_per_10k_FTE_deaths) == 0, NA,

                             (rate_per_10k_FTE_deaths - lag(rate_per_10k_FTE_deaths)) /
lag(rate_per_10k_FTE_deaths) * 100),
```

```
  yoy_rate_injuries = ifelse(is.na(lag(rate_per_100_FTE_injuries)) |
lag(rate_per_100_FTE_injuries) == 0, NA,
                           (rate_per_100_FTE_injuries - lag(rate_per_100_FTE_injuries)) /
lag(rate_per_100_FTE_injuries) * 100),


  yoy_rate_other_cases = ifelse(is.na(lag(rate_per_100_FTE_other_cases)) |
lag(rate_per_100_FTE_other_cases) == 0, NA,
                              (rate_per_100_FTE_other_cases - lag(rate_per_100_FTE_other_cases)) /
lag(rate_per_100_FTE_other_cases) * 100),


  yoy_rate_other_illnesses = ifelse(is.na(lag(rate_per_100_FTE_other_illnesses)) |
lag(rate_per_100_FTE_other_illnesses) == 0, NA,
                                  (rate_per_100_FTE_other_illnesses -
lag(rate_per_100_FTE_other_illnesses)) / lag(rate_per_100_FTE_other_illnesses) * 100)
 ) %>%
 ungroup()
```

```{r Shapiro-Wilk (FTE YoY Safe)}
safe_shapiro <- function(x) {
 x <- x[!is.na(x)]
 if (length(unique(x)) > 1) {
  return(shapiro.test(x))
 } else {
  return("Not enough variability to run Shapiro-Wilk")
 }
}
```

```
safe_shapiro(rates_2016_2023$yoy_hearing_loss)
safe_shapiro(rates_2016_2023$yoy_respiratory)
safe_shapiro(rates_2016_2023$yoy_skin_disorders)
safe_shapiro(rates_2016_2023$yoy_poisonings)
safe_shapiro(rates_2016_2023$yoy_dafw_cases)
safe_shapiro(rates_2016_2023$yoy_rate_deaths)
safe_shapiro(rates_2016_2023$yoy_rate_injuries)
safe_shapiro(rates_2016_2023$yoy_rate_other_cases)
safe_shapiro(rates_2016_2023$yoy_rate_other_illnesses)
```

```{r Kruskal-Wallis (FTE YoY)}
kruskal_results <- list(
  hearing_loss = kruskal.test(yoy_hearing_loss ~ year, data = rates_2016_2023),
  respiratory = kruskal.test(yoy_respiratory ~ year, data = rates_2016_2023),
  skin_disorders = kruskal.test(yoy_skin_disorders ~ year, data = rates_2016_2023),
  poisonings = kruskal.test(yoy_poisonings ~ year, data = rates_2016_2023),
  dafw_cases = kruskal.test(yoy_dafw_cases ~ year, data = rates_2016_2023),
  rate_deaths = kruskal.test(yoy_rate_deaths ~ year, data = rates_2016_2023),
  rate_injuries = kruskal.test(yoy_rate_injuries ~ year, data = rates_2016_2023),
  rate_other_cases = kruskal.test(yoy_rate_other_cases ~ year, data = rates_2016_2023),
  rate_other_illnesses = kruskal.test(yoy_rate_other_illnesses ~ year, data =
rates_2016_2023)
)

lapply(kruskal_results, print)
```

```{r Mann-Kendall Trend Test (FTE Rates)}
if (!require("trend")) install.packages("trend", dependencies = TRUE)
library(trend)

mk_hearing_loss <- mk.test(hearing_loss_ts)
mk_respiratory <- mk.test(respiratory_ts)
mk_skin_disorders <- mk.test(skin_disorders_ts)
mk_poisonings <- mk.test(poisonings_ts)
mk_dafw_cases <- mk.test(dafw_cases_ts)
mk_rate_deaths <- mk.test(rate_deaths_ts)
mk_rate_injuries <- mk.test(rate_injuries_ts)
mk_rate_other_cases <- mk.test(rate_other_cases_ts)
mk_rate_other_illnesses <- mk.test(rate_other_illnesses_ts)

print("Mann-Kendall Test Results:")
print("-----------------------------------------------------")
print("Hearing Loss:")
print(mk_hearing_loss)

print("Respiratory Conditions:")
print(mk_respiratory)

print("Skin Disorders:")
print(mk_skin_disorders)

print("Poisonings:")
```

```
print(mk_poisonings)

print("DAFW Cases:")
print(mk_dafw_cases)

print("Rate of Deaths:")
print(mk_rate_deaths)

print("Rate of Injuries:")
print(mk_rate_injuries)

print("Rate of Other Cases:")
print(mk_rate_other_cases)

print("Rate of Other Illnesses:")
print(mk_rate_other_illnesses)
```

```{r Pre/Post Pandemic ANOVA (FTE Rates Safe)}
rates_2016_2023 <- rates_2016_2023 %>%
  mutate(
    year = as.numeric(as.character(year)),  # Ensure year is numeric
    period = case_when(
      year <= 2019 ~ "Pre-Pandemic",
      year %in% c(2020, 2021) ~ "Pandemic",
      year >= 2022 ~ "Post-Pandemic"
    ),
```

```
  period = factor(period, levels = c("Pre-Pandemic", "Pandemic", "Post-Pandemic"))
 )

anova_results <- list(
 hearing_loss = aov(rate_per_100_FTE_hearing_loss ~ period,
                    data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_hearing_loss))),

 respiratory = aov(rate_per_100_FTE_respiratory ~ period,
                   data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_respiratory))),

 skin_disorders = aov(rate_per_100_FTE_skin_disorders ~ period,
                      data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_skin_disorders))),

 poisonings = aov(rate_per_100_FTE_poisonings ~ period,
                  data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_poisonings))),

 dafw_cases = aov(rate_per_100_FTE_dafw_cases ~ period,
                  data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_dafw_cases))),

 rate_deaths = aov(rate_per_10k_FTE_deaths ~ period,
                   data = rates_2016_2023 %>% filter(is.finite(rate_per_10k_FTE_deaths))),

 rate_injuries = aov(rate_per_100_FTE_injuries ~ period,
                     data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_injuries))),

 rate_other_cases = aov(rate_per_100_FTE_other_cases ~ period,
```

```
                data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_other_cases))),

  rate_other_illnesses = aov(rate_per_100_FTE_other_illnesses ~ period,
                             data = rates_2016_2023 %>%
filter(is.finite(rate_per_100_FTE_other_illnesses)))
)

lapply(anova_results, summary)
```

```{r regression models}
lm_results <- list(
  hearing_loss = lm(rate_per_100_FTE_hearing_loss ~ size + fta_region +
establishment_type + naics_code + state,
                    data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_hearing_loss))),

  respiratory = lm(rate_per_100_FTE_respiratory ~ size + fta_region + establishment_type +
naics_code + state,
                   data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_respiratory))),

  skin_disorders = lm(rate_per_100_FTE_skin_disorders ~ size + fta_region +
establishment_type + naics_code + state,
                      data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_skin_disorders))),

  poisonings = lm(rate_per_100_FTE_poisonings ~ size + fta_region + establishment_type +
naics_code + state,
                  data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_poisonings))),
```

```
 dafw_cases = lm(rate_per_100_FTE_dafw_cases ~ size + fta_region + establishment_type + naics_code + state,
                 data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_dafw_cases))),

 rate_deaths = lm(rate_per_10k_FTE_deaths ~ size + fta_region + establishment_type + naics_code + state,
                  data = rates_2016_2023 %>% filter(is.finite(rate_per_10k_FTE_deaths))),

 rate_injuries = lm(rate_per_100_FTE_injuries ~ size + fta_region + establishment_type + naics_code + state,
                    data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_injuries))),

 rate_other_cases = lm(rate_per_100_FTE_other_cases ~ size + fta_region + establishment_type + naics_code + state,
                       data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_other_cases))),

 rate_other_illnesses = lm(rate_per_100_FTE_other_illnesses ~ size + fta_region + establishment_type + naics_code + state,
                           data = rates_2016_2023 %>%
filter(is.finite(rate_per_100_FTE_other_illnesses)))
)

lapply(lm_results, summary)
```

```{r Main Effects Regression}
lm_main_effects <- list(
 hearing_loss = lm(rate_per_100_FTE_hearing_loss ~ size + fta_region + establishment_type + naics_code,
```

```
        data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_hearing_loss))),

 respiratory = lm(rate_per_100_FTE_respiratory ~ size + fta_region + establishment_type +
naics_code,
        data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_respiratory))),

 skin_disorders = lm(rate_per_100_FTE_skin_disorders ~ size + fta_region +
establishment_type + naics_code,
        data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_skin_disorders))),

 poisonings = lm(rate_per_100_FTE_poisonings ~ size + fta_region + establishment_type +
naics_code,
        data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_poisonings))),

 dafw_cases = lm(rate_per_100_FTE_dafw_cases ~ size + fta_region + establishment_type
+ naics_code,
        data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_dafw_cases))),

 rate_deaths = lm(rate_per_10k_FTE_deaths ~ size + fta_region + establishment_type +
naics_code,
        data = rates_2016_2023 %>% filter(is.finite(rate_per_10k_FTE_deaths))),

 rate_injuries = lm(rate_per_100_FTE_injuries ~ size + fta_region + establishment_type +
naics_code,
        data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_injuries))),

 rate_other_cases = lm(rate_per_100_FTE_other_cases ~ size + fta_region +
establishment_type + naics_code,
```

```
                data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_other_cases))),

  rate_other_illnesses = lm(rate_per_100_FTE_other_illnesses ~ size + fta_region + establishment_type + naics_code,
                            data = rates_2016_2023 %>% filter(is.finite(rate_per_100_FTE_other_illnesses)))
)

lapply(lm_main_effects, summary)
```

```{r YoY Bar Chart}
library(tidyverse)
rates_long <- rates_2016_2023 %>%
  select(year, yoy_hearing_loss, yoy_respiratory, yoy_skin_disorders, yoy_poisonings, yoy_dafw_cases,
       yoy_rate_deaths, yoy_rate_injuries, yoy_rate_other_cases, yoy_rate_other_illnesses) %>%
  pivot_longer(-year, names_to = "Illness", values_to = "YoY_Change")

ggplot(rates_long, aes(x = factor(year), y = YoY_Change, fill = is.na(YoY_Change))) +
  geom_col(aes(y = ifelse(is.na(YoY_Change), 0, YoY_Change)), color = "black") +
  facet_wrap(~Illness, scales = "free_y") +
  scale_fill_manual(
    values = c("FALSE" = "#1f77b4", "TRUE" = "gray80"),
    labels = c("Available", "Missing"),
    name = "Data Status"
  ) +
```

```
  labs(
    title = "Year-over-Year Changes in Illness Rates",
    subtitle = "Gray bars indicate missing data or unavailable base rates",
    x = "Year", y = "YoY Change (%)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```{r figure 1}
#----- Figure 1: FTE-adjusted Rates by Year (Unlogged) -----

figure1_data <- rates_2016_2023 %>%
  mutate(year = as.numeric(as.character(year))) %>%
  group_by(year) %>%
  summarise(
    Hearing_Loss = mean(rate_per_100_FTE_hearing_loss, na.rm = TRUE),
    Respiratory_Disorders = mean(rate_per_100_FTE_respiratory, na.rm = TRUE),
    Skin_Disorders = mean(rate_per_100_FTE_skin_disorders, na.rm = TRUE),
    Poisonings = mean(rate_per_100_FTE_poisonings, na.rm = TRUE),
    DAFW_Cases = mean(rate_per_100_FTE_dafw_cases, na.rm = TRUE),
    Injuries = mean(rate_per_100_FTE_injuries, na.rm = TRUE),
    Other_Cases = mean(rate_per_100_FTE_other_cases, na.rm = TRUE),
    Other_Illnesses = mean(rate_per_100_FTE_other_illnesses, na.rm = TRUE),
    Deaths_per_10k_FTE = mean(rate_per_10k_FTE_deaths, na.rm = TRUE)
  ) %>%
  pivot_longer(cols = -year, names_to = "Outcome", values_to = "Rate")
```

```
missing_years <- figure1_data %>%
  filter(Rate == 0 | is.na(Rate)) %>%
  pull(year) %>%
  unique()

figure1_data$Outcome <- factor(figure1_data$Outcome, levels = c(
  "Hearing_Loss", "Respiratory_Disorders", "Skin_Disorders", "Poisonings",
  "DAFW_Cases", "Injuries", "Other_Cases", "Other_Illnesses", "Deaths_per_10k_FTE"
), labels = c(
  "Hearing Loss", "Respiratory Disorders", "Skin Disorders", "Poisonings",
  "DAFW Cases", "Injuries", "Other Cases", "Other Illnesses", "Deaths (per 10k FTE)"
))

figure1_plot <- ggplot(figure1_data, aes(x = year, y = Rate)) +
  geom_line(color = "steelblue", linewidth = 1.1) +
  geom_point(shape = 21, fill = "white", color = "steelblue", size = 2.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray50", linewidth = 0.4) +
  facet_wrap(~ Outcome, scales = "free_y", ncol = 3) +
  labs(
    subtitle = "FTE-adjusted rates by year (2016–2023).\nRed dashed lines indicate years with zero or missing data.",
    x = "Year",
    y = "Rate"
  ) +
  scale_x_continuous(breaks = 2016:2023) +
```

```
 theme_minimal(base_size = 12, base_family = "Arial") +
 theme(
  plot.title = element_text(size = 14, face = "bold", margin = margin(b = 5)),
  plot.subtitle = element_text(size = 11, margin = margin(b = 10)),
  strip.text = element_text(face = "bold"),
  axis.text.x = element_text(angle = 45, hjust = 1),
  strip.background = element_rect(fill = "gray90", color = "gray80"),
  legend.position = "none"
 )

if (length(missing_years) > 0) {
 figure1_plot <- figure1_plot +
  geom_vline(xintercept = missing_years, linetype = "dotted", color = "red", alpha = 0.5)
}

figure1_plot
```

```{r table 1}
table2_data <- rates_2016_2023 %>%
 mutate(year = as.numeric(as.character(year))) %>%
 group_by(year) %>%
 summarise(
  `Hearing Loss` = mean(rate_per_100_FTE_hearing_loss, na.rm = TRUE),
```

```
  `Respiratory Disorders` = mean(rate_per_100_FTE_respiratory, na.rm = TRUE),
  `Skin Disorders` = mean(rate_per_100_FTE_skin_disorders, na.rm = TRUE),
  `Poisonings` = mean(rate_per_100_FTE_poisonings, na.rm = TRUE),
  `DAFW Cases` = mean(rate_per_100_FTE_dafw_cases, na.rm = TRUE),
  `Injuries` = mean(rate_per_100_FTE_injuries, na.rm = TRUE),
  `Other Cases` = mean(rate_per_100_FTE_other_cases, na.rm = TRUE),
  `Other Illnesses` = mean(rate_per_100_FTE_other_illnesses, na.rm = TRUE),
  `Deaths (per 10k FTE)` = mean(rate_per_10k_FTE_deaths, na.rm = TRUE)
 ) %>%
 mutate(across(-year, ~ ifelse(.x == 0, "0.000",
                       ifelse(.x < 0.001, "<0.001",
                           sprintf("%.3f", .x)))))

table2_data %>%
 kbl(
  caption = "",
  align = "c",
  escape = FALSE,
  position = "left"
 ) %>%
 kable_classic(full_width = FALSE, html_font = "Arial") %>%
 add_header_above(header = c(" " = 1, "Rates per 100 FTE (Deaths per 10,000 FTE)" = 9),
bold = TRUE) %>%
 row_spec(0, bold = TRUE, color = "white", background = "#4F81BD",
        extra_css = "border-top: 1px solid black; border-bottom: 1px solid black;") %>%
 row_spec(nrow(table2_data), extra_css = "border-bottom: 1px solid black;") %>%
```

```
  column_spec(1, bold = TRUE, border_right = TRUE, border_left = TRUE) %>%
  column_spec(10, border_right = TRUE)
```

```{r table 2}
n_counts <- rates_2016_2023 %>%
  summarise(
    `Hearing Loss` = sum(!is.na(yoy_hearing_loss)),
    `Respiratory Disorders` = sum(!is.na(yoy_respiratory)),
    `Skin Disorders` = sum(!is.na(yoy_skin_disorders)),
    `Poisonings` = sum(!is.na(yoy_poisonings)),
    `DAFW Cases` = sum(!is.na(yoy_dafw_cases)),
    `Injuries` = sum(!is.na(yoy_rate_injuries)),
    `Other Cases` = sum(!is.na(yoy_rate_other_cases)),
    `Other Illnesses` = sum(!is.na(yoy_rate_other_illnesses)),
    `Deaths (per 10k FTE)` = sum(!is.na(yoy_rate_deaths))
  ) %>%
  pivot_longer(everything(), names_to = "Outcome", values_to = "n")

mk_summary <- tibble::tibble(
  Outcome = c("Hearing Loss", "Respiratory Disorders", "Skin Disorders", "Poisonings",
              "DAFW Cases", "Injuries", "Other Cases", "Other Illnesses", "Deaths (per 10k FTE)"),
  `Kendall's τ` = c(
    unname(mk_hearing_loss$estimates["tau"]),
    unname(mk_respiratory$estimates["tau"]),
```

```
    unname(mk_skin_disorders$estimates["tau"]),
    unname(mk_poisonings$estimates["tau"]),
    unname(mk_dafw_cases$estimates["tau"]),
    unname(mk_rate_injuries$estimates["tau"]),
    unname(mk_rate_other_cases$estimates["tau"]),
    unname(mk_rate_other_illnesses$estimates["tau"]),
    unname(mk_rate_deaths$estimates["tau"])
  ),
  `p-value` = c(
    mk_hearing_loss$p.value,
    mk_respiratory$p.value,
    mk_skin_disorders$p.value,
    mk_poisonings$p.value,
    mk_dafw_cases$p.value,
    mk_rate_injuries$p.value,
    mk_rate_other_cases$p.value,
    mk_rate_other_illnesses$p.value,
    mk_rate_deaths$p.value
  )
)

yoy_max <- rates_2016_2023 %>%
  summarise(
    `Hearing Loss` = max(yoy_hearing_loss, na.rm = TRUE),
    `Respiratory Disorders` = max(yoy_respiratory, na.rm = TRUE),
    `Skin Disorders` = max(yoy_skin_disorders, na.rm = TRUE),
```

```
    `Poisonings` = max(yoy_poisonings, na.rm = TRUE),
    `DAFW Cases` = max(yoy_dafw_cases, na.rm = TRUE),
    `Injuries` = max(yoy_rate_injuries, na.rm = TRUE),
    `Other Cases` = max(yoy_rate_other_cases, na.rm = TRUE),
    `Other Illnesses` = max(yoy_rate_other_illnesses, na.rm = TRUE),
    `Deaths (per 10k FTE)` = max(yoy_rate_deaths, na.rm = TRUE)
  ) %>%
  pivot_longer(everything(), names_to = "Outcome", values_to = "Max YoY Change (%)")

table2_data <- mk_summary %>%
  left_join(yoy_max, by = "Outcome") %>%
  left_join(n_counts, by = "Outcome") %>%
  mutate(across(where(is.numeric), ~ round(.x, 2))) %>%
  relocate(n, .after = Outcome)

table2_data %>%
  kbl(
    caption = "",
    align = "c",
    escape = FALSE,
    position = "left"
  ) %>%
  kable_classic(full_width = FALSE, html_font = "Arial") %>%
  add_header_above(
    header = c(" " = 1, "Observations" = 1, "Mann-Kendall Trend" = 2, "Largest YoY Increase
(%)" = 1),
```

```
  bold = TRUE
 ) %>%
 row_spec(0, bold = TRUE, color = "white", background = "#4F81BD",
          extra_css = "border-top: 1px solid black; border-bottom: 1px solid black;") %>%
 row_spec(nrow(table2_data), extra_css = "border-bottom: 1px solid black;") %>%
 column_spec(1, bold = TRUE, border_right = TRUE, border_left = TRUE) %>%
 column_spec(ncol(table2_data), border_right = TRUE) %>%
 add_footnote(
  "Note: Extreme YoY changes often reflect small baseline values in rare outcomes (e.g., 0.001 to 0.01 = +900%). Trends were tested using Mann-Kendall; none were statistically significant at α = 0.05.",
  notation = "symbol"
 )
```

```{r table 3}
kruskal_n <- rates_2016_2023 %>%
 summarise(
  `Hearing Loss` = sum(!is.na(yoy_hearing_loss)),
  `Respiratory Disorders` = sum(!is.na(yoy_respiratory)),
  `Skin Disorders` = sum(!is.na(yoy_skin_disorders)),
  `Poisonings` = sum(!is.na(yoy_poisonings)),
  `DAFW Cases` = sum(!is.na(yoy_dafw_cases)),
  `Injuries` = sum(!is.na(yoy_rate_injuries)),
  `Other Cases` = sum(!is.na(yoy_rate_other_cases)),
  `Other Illnesses` = sum(!is.na(yoy_rate_other_illnesses)),
  `Deaths (per 10k FTE)` = sum(!is.na(yoy_rate_deaths))
 ) %>%
```

```
  pivot_longer(everything(), names_to = "Outcome", values_to = "n")

table3_data <- tibble::tibble(
  Outcome = c("Hearing Loss", "Respiratory Disorders", "Skin Disorders", "Poisonings",
              "DAFW Cases", "Injuries", "Other Cases", "Other Illnesses", "Deaths (per 10k FTE)"),
  `Kruskal-Wallis p-value` = c(
    kruskal_results$hearing_loss$p.value,
    kruskal_results$respiratory$p.value,
    kruskal_results$skin_disorders$p.value,
    kruskal_results$poisonings$p.value,
    kruskal_results$dafw_cases$p.value,
    kruskal_results$rate_injuries$p.value,
    kruskal_results$rate_other_cases$p.value,
    kruskal_results$rate_other_illnesses$p.value,
    kruskal_results$rate_deaths$p.value
  ) %>% round(4)
) %>%
  left_join(kruskal_n, by = "Outcome") %>%
  relocate(n, .after = Outcome)

table3_data %>%
  kbl(caption = "",
      align = "c", escape = FALSE, position = "left") %>%
  kable_classic(full_width = FALSE, html_font = "Arial") %>%
  add_header_above(
    header = c(" " = 1, "Observations" = 1, "Kruskal-Wallis Test Results" = 1),
```

```
  bold = TRUE
 ) %>%
 row_spec(0, bold = TRUE, color = "white", background = "#4F81BD",
          extra_css = "border-top: 1px solid black; border-bottom: 1px solid black;") %>%
 row_spec(nrow(table3_data), extra_css = "border-bottom: 1px solid black;") %>%
 column_spec(1, bold = TRUE, border_right = TRUE, border_left = TRUE) %>%
 column_spec(ncol(table3_data), border_right = TRUE)
```

```{r table 4}
anova_n <- rates_2016_2023 %>%
 summarise(
   `Hearing Loss` = sum(is.finite(rate_per_100_FTE_hearing_loss)),
   `Respiratory Disorders` = sum(is.finite(rate_per_100_FTE_respiratory)),
   `Skin Disorders` = sum(is.finite(rate_per_100_FTE_skin_disorders)),
   `Poisonings` = sum(is.finite(rate_per_100_FTE_poisonings)),
   `DAFW Cases` = sum(is.finite(rate_per_100_FTE_dafw_cases)),
   `Injuries` = sum(is.finite(rate_per_100_FTE_injuries)),
   `Other Cases` = sum(is.finite(rate_per_100_FTE_other_cases)),
   `Other Illnesses` = sum(is.finite(rate_per_100_FTE_other_illnesses)),
   `Deaths (per 10k FTE)` = sum(is.finite(rate_per_10k_FTE_deaths))
 ) %>%
 pivot_longer(everything(), names_to = "Outcome", values_to = "n")

table4_data <- tibble::tibble(
 Outcome = c("Hearing Loss", "Respiratory Disorders", "Skin Disorders", "Poisonings",
             "DAFW Cases", "Injuries", "Other Cases", "Other Illnesses", "Deaths (per 10k FTE)"),
```

```
  `ANOVA p` = c(
   summary(anova_results$hearing_loss)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$respiratory)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$skin_disorders)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$poisonings)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$dafw_cases)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$rate_injuries)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$rate_other_cases)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$rate_other_illnesses)[[1]]["Pr(>F)"][1,1],
   summary(anova_results$rate_deaths)[[1]]["Pr(>F)"][1,1]
  ) %>% round(4)
) %>%
  left_join(anova_n, by = "Outcome") %>%
  relocate(n, .after = Outcome)

table4_data %>%
  kbl(caption = "",
      align = "c", escape = FALSE, position = "left") %>%
  kable_classic(full_width = FALSE, html_font = "Arial") %>%
  add_header_above(header = c(" " = 1, "Observations" = 1, "ANOVA by Pandemic Period" =
1), bold = TRUE) %>%
  row_spec(0, bold = TRUE, color = "white", background = "#4F81BD",
           extra_css = "border-top: 1px solid black; border-bottom: 1px solid black;") %>%
  row_spec(nrow(table4_data), extra_css = "border-bottom: 1px solid black;") %>%
  column_spec(1, bold = TRUE, border_right = TRUE, border_left = TRUE) %>%
  column_spec(ncol(table4_data), border_right = TRUE)
```

```
```{r table 5}
simplify_predictors <- function(variables) {
  simplified <- unique(case_when(
    grepl("^state", variables) ~ "state",
    grepl("^fta_region", variables) ~ "fta_region",
    grepl("^naics_code", variables) ~ "naics_code",
    grepl("^establishment_type", variables) ~ "establishment_type",
    grepl("^size", variables) ~ "size",
    TRUE ~ variables
  ))
  paste(sort(simplified), collapse = "; ")
}

extract_model_info <- function(model) {
  summary_model <- summary(model)
  coefs <- summary_model$coefficients
  adj_r2 <- summary_model$adj.r.squared
  sig_vars <- rownames(coefs)[which(coefs[, 4] < 0.05)]
  n_obs <- length(model$fitted.values)
  list(
    adj_r2 = adj_r2,
    sig_vars = simplify_predictors(sig_vars),
    n = n_obs
  )
```
```

```
}

table5_data <- tibble::tibble(
  Outcome = names(lm_results),
  Model = lm_results
) %>%
  rowwise() %>%
  mutate(
    `Adjusted R²` = round(extract_model_info(Model)$adj_r2, 3),
    `Significant Predictors (p < 0.05)` = extract_model_info(Model)$sig_vars,
    n = extract_model_info(Model)$n
  ) %>%
  ungroup() %>%
  mutate(Outcome = case_when(
    Outcome == "hearing_loss" ~ "Hearing Loss",
    Outcome == "respiratory" ~ "Respiratory Disorders",
    Outcome == "skin_disorders" ~ "Skin Disorders",
    Outcome == "poisonings" ~ "Poisonings",
    Outcome == "dafw_cases" ~ "DAFW Cases",
    Outcome == "rate_deaths" ~ "Deaths (per 10k FTE)",
    Outcome == "rate_injuries" ~ "Injuries",
    Outcome == "rate_other_cases" ~ "Other Cases",
    Outcome == "rate_other_illnesses" ~ "Other Illnesses",
    TRUE ~ Outcome
  )) %>%
  select(Outcome, n, `Adjusted R²`, `Significant Predictors (p < 0.05)`)
```

```
table5_data %>%
  kbl(
    caption = "",
    align = "c",
    escape = FALSE,
    position = "left"
  ) %>%
  kable_classic(full_width = FALSE, html_font = "Arial") %>%
  add_header_above(
    header = c(" " = 1, " " = 1, "Model Performance" = 1, "Key Predictors" = 1),
    bold = TRUE
  ) %>%
  row_spec(0, bold = TRUE, color = "white", background = "#4F81BD",
           extra_css = "border-top: 1px solid black; border-bottom: 1px solid black;") %>%
  row_spec(nrow(table5_data), extra_css = "border-bottom: 1px solid black;") %>%
  column_spec(1, bold = TRUE, border_left = TRUE) %>%
  column_spec(ncol(table5_data), border_right = TRUE)
```

```{r table 6}
find_outlier_years <- function(df, outcome_col, year_col = "year", threshold = 2) {
  outcome_data <- df %>%
    group_by(!!sym(year_col)) %>%
    summarise(mean_rate = mean(.data[[outcome_col]], na.rm = TRUE)) %>%
```

```
  mutate(z = scale(mean_rate)) %>%
  filter(abs(z) > threshold) %>%
  pull(!!sym(year_col))

 if (length(outcome_data) == 0) return("—") else return(paste(outcome_data, collapse = ",
"))
}

table6_data <- tibble::tibble(
 Outcome = c(
  "Hearing Loss", "Respiratory Disorders", "Skin Disorders", "Poisonings",
  "DAFW Cases", "Injuries", "Other Cases", "Other Illnesses", "Deaths (per 10k FTE)"
 ),
 `Outlier Years` = c(
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_hearing_loss"),
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_respiratory"),
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_skin_disorders"),
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_poisonings"),
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_dafw_cases"),
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_injuries"),
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_other_cases"),
  find_outlier_years(rates_2016_2023, "rate_per_100_FTE_other_illnesses"),
  find_outlier_years(rates_2016_2023, "rate_per_10k_FTE_deaths")
 )
)
```

```
table6_data %>%
  kbl(
    caption = "",
    align = "lc",
    escape = FALSE,
    position = "left"
  ) %>%
  kable_classic(full_width = FALSE, html_font = "Arial") %>%
  column_spec(1, bold = TRUE, border_right = TRUE, border_left = TRUE) %>%
  column_spec(2, border_right = TRUE) %>%
  row_spec(0, bold = TRUE, color = "white", background = "#4F81BD",
           extra_css = "border-top: 1px solid black; border-bottom: 1px solid black;") %>%
  row_spec(nrow(table6_data), extra_css = "border-bottom: 1px solid black;")


```
```{r figure 2}
rates_long <- rates_2016_2023 %>%
  select(year, yoy_hearing_loss, yoy_respiratory, yoy_skin_disorders, yoy_poisonings, yoy_dafw_cases,
         yoy_rate_deaths, yoy_rate_injuries, yoy_rate_other_cases, yoy_rate_other_illnesses) %>%
  pivot_longer(-year, names_to = "Illness", values_to = "YoY_Change") %>%
  mutate(
    Illness = recode(Illness,
      "yoy_hearing_loss" = "Hearing Loss",
      "yoy_respiratory" = "Respiratory Disorders",
      "yoy_skin_disorders" = "Skin Disorders",
```

```
    "yoy_poisonings" = "Poisonings",
    "yoy_dafw_cases" = "DAFW Cases",
    "yoy_rate_deaths" = "Deaths (per 10k FTE)",
    "yoy_rate_injuries" = "Injuries",
    "yoy_rate_other_cases" = "Other Cases",
    "yoy_rate_other_illnesses" = "Other Illnesses"
  ),
  Missing = is.na(YoY_Change),
  YoY_Change = replace_na(YoY_Change, 0)
)

ggplot(rates_long, aes(x = factor(year), y = YoY_Change, fill = Missing)) +
  geom_col(color = "NA", width = 0.7, show.legend = FALSE) +
  facet_wrap(~ Illness, scales = "free_y", ncol = 3) +
  scale_fill_manual(values = c("FALSE" = "#4682B4", "TRUE" = "gray80")) +
  labs(
    subtitle = "Note: Bars for years with missing or ineligible prior-year data were removed during preprocessing.\nY-axis scales vary by outcome to improve readability",
    x = "Year", y = "Percent Change (%)"
  ) +
  theme_minimal(base_family = "Arial") +
  theme(
    strip.text = element_text(face = "bold", size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid.major.y = element_line(color = "gray80"),
    panel.grid.minor.y = element_blank(),
```

```
    panel.border = element_rect(color = "gray60", fill = NA, linewidth = 0.5),
    plot.title = element_text(face = "bold", size = 14),
    plot.subtitle = element_text(size = 10)
  )

```

```{r data combine code table}
years <- 2016:2023

clustering_summary <- data.frame(
  Year = integer(),
  Original_Rows = integer(),
  Merged_Rows = integer(),
  Row_Difference = integer(),
  Percent_Retained = numeric(),
  stringsAsFactors = FALSE
)

for (year in years) {
  original_name <- paste0("ITA_", year)
  merged_name <- paste0("ITA_Merged_", year)

  if (exists(original_name) && exists(merged_name)) {
    original_rows <- nrow(get(original_name))
    merged_rows <- nrow(get(merged_name))
```

```
    row_diff <- original_rows - merged_rows
    percent_retained <- (merged_rows / original_rows) * 100

    clustering_summary <- rbind(clustering_summary, data.frame(
      Year = year,
      Original_Rows = original_rows,
      Merged_Rows = merged_rows,
      Row_Difference = row_diff,
      Percent_Retained = round(percent_retained, 2)
    ))
  }
}



colnames(clustering_summary) <- c(
  "Year", "Original Rows", "Merged Rows", "Row Difference", "Percent Combined"
)

clustering_summary %>%
  kbl(
    caption = "",
    align = "c",
    escape = FALSE,
    position = "left"
  ) %>%
  kable_classic(full_width = FALSE, html_font = "Arial") %>%
```

```
  column_spec(1, bold = TRUE, border_right = TRUE, border_left = TRUE) %>%
  column_spec(ncol(clustering_summary), border_right = TRUE) %>%
  row_spec(0, bold = TRUE, color = "white", background = "#4F81BD",
           extra_css = "border-top: 1px solid black; border-bottom: 1px solid black;") %>%
  row_spec(nrow(clustering_summary), extra_css = "border-bottom: 1px solid black;")
```