Road accidents in New York

Luca Renz & Alvaro Cervan

30/08/2024

Contents

1 Data Cleaning and Merging

```
## packages

# install.packages("lubridate")
library(lubridate) # used to round to the next hour

 #install.packages("dplyr")
library(dplyr)

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

# VEHICLES DF - CLEANING
vehicles_df <- read.csv("data/vehicles.csv", header=TRUE)

## columns to be removed due to high NaN ratio.
columns_to_remove <- c(
  "CROSS.STREET.NAME",
  "ON.STREET.NAME",
  "OFF.STREET.NAME",
  "CONTRIBUTING.FACTOR.VEHICLE.3",
  "CONTRIBUTING.FACTOR.VEHICLE.4",
  "CONTRIBUTING.FACTOR.VEHICLE.5",
  "VEHICLE.TYPE.CODE.3",
  "VEHICLE.TYPE.CODE.4",
  "VEHICLE.TYPE.CODE.5",
  "COLLISION_ID"
)

min_cutoff_date <- as.Date("2016-01-01")
```

```r
max_cutoff_date <- as.Date("2022-01-01")

## Create broader categories for main causes
conditions <- c("Aggressive Driving/Road Rage", "Pavement Slippery", "Following Too Closely",
                "Unspecified", "", "Passing Too Closely", "Driver Inexperience",
                "Passing or Lane Usage Improper", "Turning Improperly", "Unsafe Lane Changing",
                "Unsafe Speed", "Reaction to Uninvolved Vehicle", "Steering Failure",
                "Traffic Control Disregarded", "Other Vehicular", "Driver Inattention/Distraction",
                "Oversized Vehicle", "Pedestrian/Bicyclist/Other Pedestrian Error/Confusion",
                "Alcohol Involvement", "View Obstructed/Limited", "Failure to Yield Right-of-Way",
                "Illnes", "Lost Consciousness", "Brakes Defective", "Backing Unsafely", "Glare",
                "Passenger Distraction", "Fell Asleep", "Obstruction/Debris", "Tinted Windows",
                "Animals Action", "Drugs (illegal)", "Pavement Defective", "Other Lighting Defects",
                "Outside Car Distraction", "Driverless/Runaway Vehicle", "Tire Failure/Inadequate",
                "Fatigued/Drowsy", "Headlights Defective", "Accelerator Defective",
                "Failure to Keep Right", "Physical Disability", "Eating or Drinking",
                "Cell Phone (hands-free)", "Lane Marking Improper/Inadequate",
                "Cell Phone (hand-Held)", "Using On Board Navigation Device", "Other Electronic Devic
                "Traffic Control Device Improper/Non-Working", "Tow Hitch Defective",
                "Windshield Inadequate", "Vehicle Vandalism", "Shoulders Defective/Improper",
                "Prescription Medication", "Listening/Using Headphones", "Texting", "80",
                "Reaction to Other Uninvolved Vehicle", "1", "Drugs (Illegal)", "Illness",
                "Cell Phone (hand-held)")

## mapping for conditions
categorize_condition <- function(condition) {
  if (grepl("Driving|Following|Passing|Inexperience|Improper|Changing|Speed|Distraction|Alcohol|Drugs
    return("Human Error")
  } else if (grepl("Brakes|Steering|Tire|Headlights|Accelerator|Windshield|Tow Hitch|Defective|Failur
    return("Mechanical Error")
  } else if (grepl("Pavement|Glare|Obstruction|Weather|Animals|View|Control|Shoulders", condition, ig
    return("Environmental Conditions")
  } else if (grepl("Illness|Lost Consciousness|Physical Disability", condition, ignore.case = TRUE)) 
    return("Medical Condition")
  } else if (condition == "" || grepl("Unspecified|Other", condition, ignore.case = TRUE)) {
    return("Other/Unspecified")
  } else {
    return("Other/Unspecified")
  }
}

vehicles_df <- vehicles_df %>%
  mutate(CRASH.DATE = as.Date(CRASH.DATE, format = "%m/%d/%Y")) %>%  # Convert CRASH.DATE to Date for
  select(-all_of(columns_to_remove)) %>%
  filter(CRASH.DATE >= min_cutoff_date) %>%
  filter(CRASH.DATE < max_cutoff_date) %>%
  filter(!is.na(LATITUDE) & !is.na(LONGITUDE)) %>%
  filter(!is.na(CRASH.TIME)) %>%
  ##create new column with broader category for accident
  mutate(Category = sapply(CONTRIBUTING.FACTOR.VEHICLE.1, categorize_condition)) %>%
  # Combine date and time into one string
  mutate(CRASH.DATETIME = as.POSIXct(paste(CRASH.DATE, CRASH.TIME), format = "%Y-%m-%d %H:%M")) %>%
  # Round down to the current hour
```

```
    mutate(CRASH.DATETIME = floor_date(CRASH.DATETIME, "hour")) %>%
    # enriching datetime format to weekdays, month, quarter, year
    mutate(
      Weekday = wday(CRASH.DATETIME, label = TRUE, abbr = FALSE),
      Month = month(CRASH.DATETIME, label = TRUE, abbr = FALSE),
      Quarter = quarter(CRASH.DATETIME),
      Year = year(CRASH.DATETIME) ,
      Day = day(CRASH.DATETIME),
      Hour = hour(CRASH.DATETIME),
      TimeOfDay = case_when(
        Hour >= 6 & Hour < 12  ~ "Morning",
        Hour >= 12 & Hour < 17 ~ "Afternoon",
        Hour >= 17 & Hour < 21 ~ "Evening",
        Hour >= 21 ~ "Night",
        Hour < 6 ~ "Night",
        TRUE ~ "Unknown",
      )
    ) %>%
    filter(!is.na(CRASH.TIME)) %>%
    mutate(IS.DEADLY.ACCIDENT = (NUMBER.OF.PERSONS.KILLED + NUMBER.OF.PEDESTRIANS.KILLED + NUMBER.OF.CY
           IS.INJURED.ACCIDENT = (NUMBER.OF.PERSONS.INJURED + NUMBER.OF.PEDESTRIANS.INJURED + NUMBER.OF

  # WEATHER DF - CLEANING
  weather_df <- read_excel("data/weather.xlsx")
  weather_df <- weather_df %>%
    rename("temperature_celsius" = "temperature_2m (Â°C)", "winddirection_10m_degrees" = "winddirection_
    mutate(time = as.POSIXct(time, format = "%Y-%m-%dT%H:%M")) %>% # Convert the TIME column to correct
    filter(!is.na(time)) %>% #filter rows where there is no time
    filter(rowSums(is.na(.)) / ncol(weather_df) < 0.8) # Filter rows where 80% or more columns are NaN/

  # MERGING DATAFRAMES
  merged_df <- left_join(vehicles_df, weather_df, by = c("CRASH.DATETIME" = "time"))

  write.csv(merged_df,"merged_all_years.csv")

  split_datasets <- split(merged_df, merged_df$Year)

  # Save each year to a separate file
  lapply(names(split_datasets), function(year) {
    filename <- paste0("merged_data_", year, ".csv")
    write.csv(split_datasets[[year]], file = filename, row.names = FALSE)
  })
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
```

```
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
```

1.1 Road accidents in New York

1.1.1 Total Accidents in New York

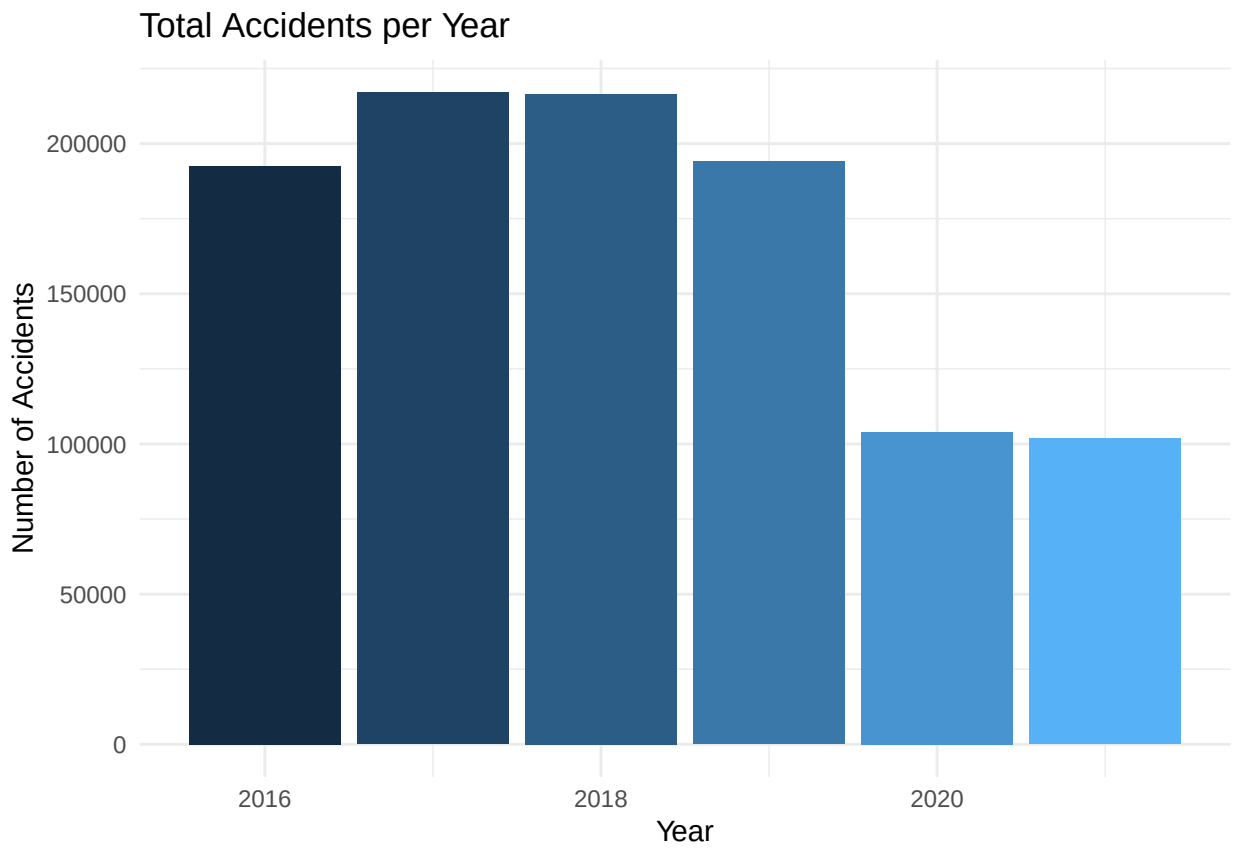
```r
# Loading necessary libraries
library(ggplot2)
library(dplyr)

# Load the data from a CSV file
data <- read.csv("data/merged_all_years.csv", header = TRUE)

# Convert the CRASH.DATE into a Date object
data$CRASH.DATE <- as.Date(data$CRASH.DATE, "%d/%m/%Y")

# Aggregating data by Year
yearly_accidents <- data %>%
  group_by(Year) %>%
  summarise(Total_Accidents = n(), .groups = 'drop')

# Plotting the data
ggplot(yearly_accidents, aes(x = Year, y = Total_Accidents, fill = Year)) +
  geom_bar(stat = "identity") +  # Set a single color for the bars
  labs(title = "Total Accidents per Year",
       x = "Year",
       y = "Number of Accidents") + theme_minimal() + theme(legend.position = "none")
```




```r
library(dplyr)
library(ggplot2)

# Load the data from the CSV file
data <- read.csv("data/merged_data_2021.csv", header = TRUE)

# Filter for deadly accidents
deadly_accidents <- data %>%
  filter(IS.DEADLY.ACCIDENT == TRUE)


# Define a function to map VEHICLE.TYPE.CODE.1 to a more general VEHICLE.TYPE
map_vehicle_type <- function(vehicle_code) {
  vehicle_code <- tolower(vehicle_code)  # Convert to lowercase for consistent matching

  if (grepl("ambulance|police|nypd|emergency|rescue|nyc ambula|ambu", vehicle_code)) {
    return("Emergency Vehicle")
  } else if (grepl("bus|school|shuttle|coach", vehicle_code)) {
    return("Bus")
  } else if (grepl("sedan|4dr|2dr|passenger|car", vehicle_code)) {
    return("Car")
  } else if (grepl("van|camper|rv|minivan|mini van|suburban|vanette|uhaul|work van|van/truck|cargo van|p
    return("Van")
  } else if (grepl("tractor|agriculture|chassis|farm|combine|excavator|tract|bulldozer|backhoe|crane|mi
    return("Construction Machinery")
  } else if (grepl("truck|pick|pkup|box|flat|tow|dump|stake|tanker|trailer|semi|tractor|garbage|delivery
```

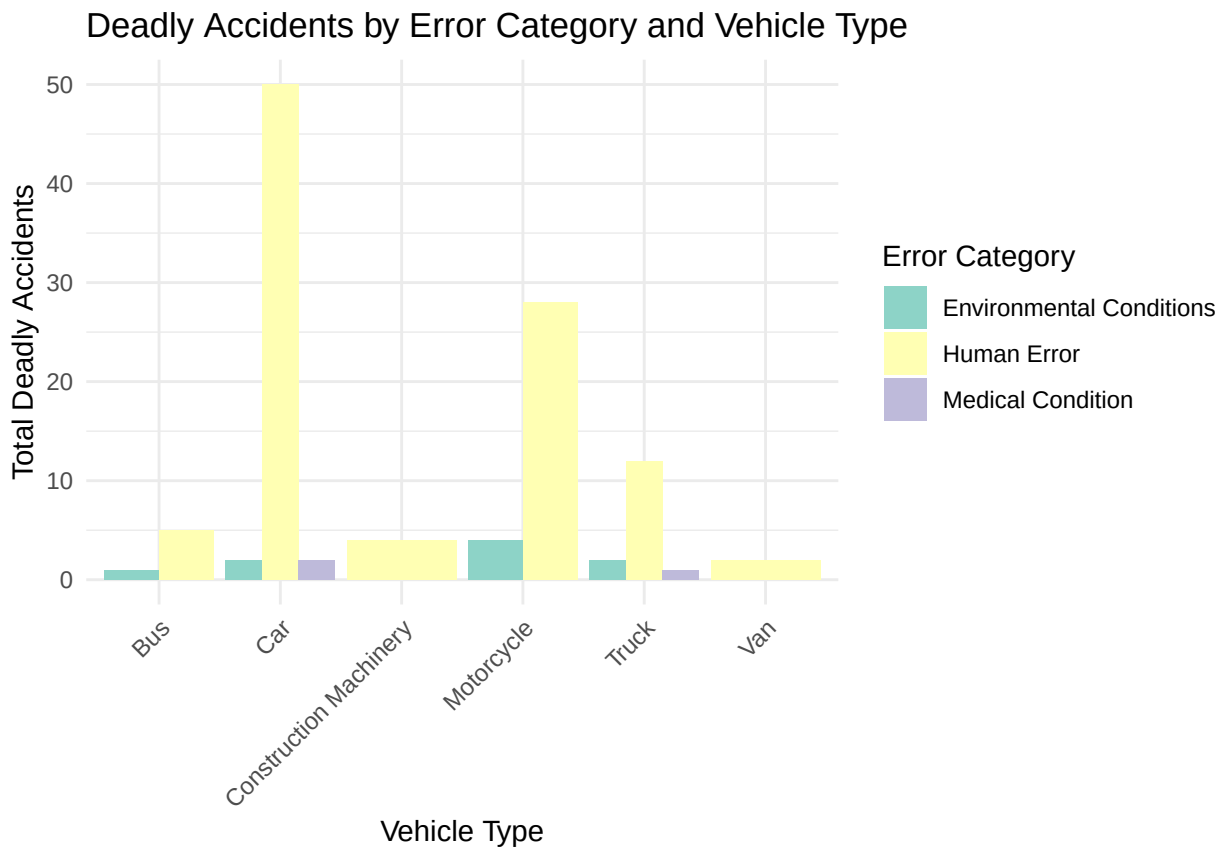
```
    return("Truck")
  } else if (grepl("motorcycle|bike|scooter|moped|e-bike|e-scooter|dirt bike|minibike|motor|vespa", vehi
    return("Motorcycle")
  } else if (grepl("bicycle|bike|pedicab", vehicle_code)) {
    return("Bicycle")
  } else if (grepl("unknown|other|unk", vehicle_code)) {
    return("Unknown")
  } else {
    return("Unknown")
  }
}

# Apply the mapping function to create the VEHICLE.TYPE column
deadly_accidents <- deadly_accidents %>%
  mutate(VEHICLE.TYPE = sapply(VEHICLE.TYPE.CODE.1, map_vehicle_type))

filtered_deadly_accidents <- deadly_accidents %>%
  filter(VEHICLE.TYPE != "Unknown" & Category != "Other/Unspecified")

# Count the number of deadly accidents by category and vehicle type
error_counts <- filtered_deadly_accidents %>%
  group_by(Category, VEHICLE.TYPE) %>%
  summarise(Total_Deadly_Accidents = n(), .groups = 'drop')

# Create the plot
ggplot(error_counts, aes(x = VEHICLE.TYPE, y = Total_Deadly_Accidents, fill = Category)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Deadly Accidents by Error Category and Vehicle Type",
       x = "Vehicle Type",
       y = "Total Deadly Accidents",
       fill = "Error Category") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  # Rotate x-axis labels for better readabi
  scale_fill_brewer(palette = "Set3")  # Apply a color palette
```



1.2 Chapter of Choice (Map (Maybe html/interactive))

```
#install.packages("sf")
#install.packages("terra")
#install.packages("spData")
#install.packages("spDataLarge", repos = "https://nowosad.r-universe.dev")
#library(sf)            # classes and functions for vector data
#> Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 8.2.1; sf_use_s2() is TRUE
#library(terra)         # classes and functions for raster data
#> Terra version 1.4.12
#library(spData)        # sample spatial data
#library(spDataLarge)   # large sample spatial data
```

```
#plot new york
#ny <- st_read(system.file("shape/ny.shp", package="sf"))
#plot(ny["NAME"])
```