testing_project

2024-01-22

Uploading all data

```
# Set the working directory
setwd("C:/Users/jans7/OneDrive - Marquette University/SP24/COSC 6510 - Data Intelligence/Crime_Data_fol

# list of all csv files in folder
csv_files <- list.files(pattern = "\\.csv$")

# reading each CSV file into a data frame and store them in a list
data_list <- lapply(csv_files, read.csv)
```

What data to Keep

First want to see what columns are the same across datasets

```
# Find common columns
common_columns <- names(data_list[[1]])

for (i in seq_along(data_list)[-1]) {
  common_columns <- intersect(common_columns, names(data_list[[i]]))
}

# Find columns not in common
non_common_columns <- setdiff(unique(unlist(lapply(data_list, names))), common_columns)

cat("Common columns across all files:\n")
```

```
## Common columns across all files:
```

```
print(common_columns)
```

```
##  [1] "Report_No"     "Reported_Date" "From_Date"     "To_Date"
##  [5] "Offense"       "IBRS"          "Description"   "Beat"
##  [9] "Address"       "City"          "Zip.Code"      "Rep_Dist"
## [13] "Area"          "DVFlag"        "Involvement"   "Race"
## [17] "Sex"           "Age"
```

```r
cat("\nColumns not in common across all files:\n")
```

```
##
## Columns not in common across all files:
```

```r
print(non_common_columns)
```

```
##  [1] "Reported.Time"      "From.Time"          "To.Time"
##  [4] "Invl_No"            "Firearm.Used.Flag"  "Latitude"
##  [7] "Longitude"          "Location.1"         "Reported_Time"
## [10] "From_Time"          "To_Time"            "Location"
## [13] "Age_Range"          "Fire.Arm.Used.Flag"
```

I should look at what is contained in location vs location.1 files then figure out how many have lat long and how that different than location

I think reported data, from date, tp date, offense,address, city, zip.code, rep_dist and area are all important

Don't think I need IBRS, beat, DV_flag, involvement, race, sex, age

I am thinking that I could do somehting kinda cool is figure out what percentage have lat long, and what have zipcode, and can make a dictionary - however, after some research "The average land area of a zip code is around 90 square miles," - avoid using zipcodes for geospatial analysis https://towardsdatascience.com/stop-using-zip-codes-for-geospatial-analysis-ceacb6e80c38

Data Wrangling - Lat & Long

Lets look at a random sample of data from each data frame

```r
for (i in seq_along(data_list)) {
  cat("Sample of 5 rows from", csv_files[i], ":\n")
  #print(head(data_list[[i]]))
  cat("\n")
}
```

```
## Sample of 5 rows from KCPD_Crime_Data_2015_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2016_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2017_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2018_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2019_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2020_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2021_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2022_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2023_20240202.csv :
##
## Sample of 5 rows from KCPD_Crime_Data_2024_20240408.csv :
```

Zipcode 99999 means that they didnt have an address location.1 seams to have KC district and lat long, while location can be the whole address together, Then some locations are the geom point

Defintely don't Need Report_No, From_Date, From.Time, To_Date, To.Time, IBRS, Beat, DVFlag, Invl_No, Involvement, Race, Sex, Age

Recognized that 2021-24 have the same type of data in "Location" (this would be training on 3 years) While 2016-20 have a different style of the location data where it is that address the n(the data point). Then 2015 has a lat and long column

Extracting the () from geom point data in the 2016-20 records as well as 2021-24 with two different techniques

FUNCTIONS!!!!!

```
extractLatLonFromPoint <- function(file_index, data_list) {
  # Extract dataframe based on file_index
  df <- data_list[[file_index]]

  # Initialize empty vectors for latitude and longitude
  latitude <- numeric(nrow(df))
  longitude <- numeric(nrow(df))

  # Loop through each row of the data frame
  for (i in 1:nrow(df)) {
    # Extract latitude and longitude from the "POINT" string format
    match <- regmatches(df$Location[i],
                        regexec("POINT \\((-?[0-9.]+) (-?[0-9.]+)\\)", df$Location[i]))

    # Check if a match was found
    if (!is.na(match[[1]][1])) {
      latitude[i] <- as.numeric(match[[1]][3])
      longitude[i] <- as.numeric(match[[1]][2])
    } else {
      latitude[i] <- NA
      longitude[i] <- NA
    }
  }

  # Add latitude and longitude as new columns to the data frame
  df$Latitude <- latitude
  df$Longitude <- longitude

  # Return the modified dataframe
  return(df)
}

extractLatLonFromLongExpre <- function(file_index, data_list) {
  # Extract dataframe based on file_index
  data <- data_list[[file_index]]

  # Initialize empty vectors for latitude and longitude
  latitude <- numeric(nrow(data))
  longitude <- numeric(nrow(data))

  # Loop through each row of the data
  for (i in 1:nrow(data)) {
```

```
    # Suppress the warning for coercion to NA
    suppressWarnings({
      # Extract latitude and longitude using regex
      match <- regmatches(data$Location[i],
                          regexpr("\\((-?\\d+\\.\\d+), (-?\\d+\\.\\d+)\\)", data$Location[i]))

      # Check if a match was found
      if (length(match) > 0) {
        latitude[i] <- as.numeric(unlist(strsplit(gsub("[\\(\\)]", "", match), ", ")))[1]
        longitude[i] <- as.numeric(unlist(strsplit(gsub("[\\(\\)]", "", match), ", ")))[2]
      } else {
        latitude[i] <- NA
        longitude[i] <- NA
      }
    })
  }

  # Add latitude and longitude to the original data frame
  data$Latitude <- latitude
  data$Longitude <- longitude

  # Return the modified dataframe
  return(data)
}
```

```
cleandata <- list()

# Loop through each element in data_list and copy it to cleandata
for (i in seq_along(data_list)) {
  cleandata[[i]] <- data_list[[i]]
}

for (i in 1:10) {
  print(i)
  if (i == 1) {
    # Store the first dataset directly without modification
    cleandata[[i]] <- data_list[[i]]
  } else if (i >= 2 && i <= 6) {
    # Apply extract_lat_lon_regex for datasets 2 to 6
    cleandata[[i]] <- extractLatLonFromLongExpre(i, data_list)
  } else if (i >= 7 && i <= 10){
    # Apply extract_lat_lon for datasets 7 to 10
    cleandata[[i]] <- extractLatLonFromPoint(i, data_list)
  }
  else {
    print('Error')
  }
}
```

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

```
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

handling missing lat and long

All behind paywalls? Unless figure out Google API. Did not figure this out

Combining Data

I want the all crime data files to be in the same data file. Therefore we need all the same columns and column types

Lets reprint common and not common columns

```
# Find common columns
common_columns <- names(cleandata[[1]])

for (i in seq_along(cleandata)) {
  common_columns <- intersect(common_columns, names(cleandata[[i]]))
}

# Find columns not in common
non_common_columns <- setdiff(unique(unlist(lapply(cleandata, names))), common_columns)

cat("Common columns across all files:\n")
```

```
## Common columns across all files:
```

```
print(common_columns)
```

```
##  [1] "Report_No"     "Reported_Date" "From_Date"     "To_Date"
##  [5] "Offense"       "IBRS"          "Description"   "Beat"
##  [9] "Address"       "City"          "Zip.Code"      "Rep_Dist"
## [13] "Area"          "DVFlag"        "Involvement"   "Race"
## [17] "Sex"           "Age"           "Latitude"      "Longitude"
```

```
cat("\nColumns not in common across all files:\n")
```

```
##
## Columns not in common across all files:
```

```
print(non_common_columns)
```

```
##  [1] "Reported.Time"      "From.Time"          "To.Time"
##  [4] "Invl_No"            "Firearm.Used.Flag"  "Location.1"
##  [7] "Reported_Time"      "From_Time"          "To_Time"
## [10] "Location"           "Age_Range"          "Fire.Arm.Used.Flag"
```

```r
# Items to remove from common_columns
# because I dont really care to have these in final dataset
items_to_remove <- c("Report_No", "IBRS", "Beat", "DVFlag", "Involvement",
                     "Race", "Sex", "Offense", "From_Date", "To_Date")

# Remove items from common_columns
common_columns <- setdiff(common_columns, items_to_remove)


print(common_columns)
```

```
##  [1] "Reported_Date" "Description"   "Address"       "City"
##  [5] "Zip.Code"      "Rep_Dist"      "Area"          "Age"
##  [9] "Latitude"      "Longitude"
```

Have to clean zip code data to be the same type

```r
for (i in seq_along(cleandata)) {
  # Convert "Zip.Code" column to integers
  cleandata[[i]]$Zip.Code <- as.integer(cleandata[[i]]$Zip.Code)
}
```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

Combining the Data

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Initialize combined_data with the first data frame
combined_data <- cleandata[[1]][common_columns]

# Loop through the remaining data frames and combine them
# by adding on the rows
for (i in 2:length(cleandata)) {
  combined_data <- rbind(combined_data, cleandata[[i]][common_columns])
}

head(combined_data)
```

```
##   Reported_Date          Description                     Address        City
## 1    03/06/2015       Misc Violation    BROADWAY and WESTPORT RD KANSAS CITY
## 2    09/21/2015 Aggravated Assault (
## 3    09/21/2015       Family Offense
## 4    09/08/2015           Auto Theft PROSPECT AV and E TRUMAN RD KANSAS CITY
## 5    05/19/2015 Possession/Sale/Dist    VICTOR ST and WALROND AV KANSAS CITY
## 6    08/31/2015 Non Aggravated Assau       PASEO and E TRUMAN RD KANSAS CITY
##   Zip.Code Rep_Dist Area Age Latitude Longitude
## 1    64131   PJ3229  CPD  NA  38.9767  -94.5767
## 2    99999                NA   0.0000    0.0000
## 3    99999                NA   0.0000    0.0000
## 4    64126   PJ7474  EPD  NA  39.0947  -94.5516
## 5    64128   PJ2340  EPD  NA  39.0735  -94.5461
## 6    61109   PJ1326  CPD  29  42.2167  -89.0251
```

Cleaning Data

Looking at Combined Data 'NA's and potentially cleaning issues

```
# Find character columns in combined_data
character_cols <- sapply(combined_data, is.character)

# Replace empty strings with NA for character columns
# the replace 0 values in Zip.Code column with NA
combined_data[character_cols] <- lapply(combined_data[character_cols], na_if, "")
combined_data$Zip.Code[combined_data$Zip.Code == 0] <- NA

# cleaning up obvious entering mistakes

# Replace Age values above 100 with NA
combined_data$Age[combined_data$Age > 100] <- NA
# Replace 0 values in Latitude column with NA and corresponding Longitude values
combined_data$Longitude[combined_data$Latitude == 0] <- NA
combined_data$Latitude[combined_data$Latitude == 0] <- NA

# throwing a lat and long box around KC and
# if coordinates fall outside of the box then remove

# Replace Latitude values less than 38 and greater than
# or equal to 40 with NA and corresponding Longitude values
combined_data$Longitude[combined_data$Latitude < 38 | combined_data$Latitude >= 40] <- NA
combined_data$Latitude[combined_data$Latitude < 38 | combined_data$Latitude >= 40] <- NA

# Replace Longitude values greater than -94
# or less than -95 with NA and corresponding Latitude values
combined_data$Latitude[combined_data$Longitude > -94 | combined_data$Longitude < -95] <- NA
combined_data$Longitude[combined_data$Longitude > -94 | combined_data$Longitude < -95] <- NA

# Verify changes
summary(combined_data)
```

```
##  Reported_Date      Description          Address              City
```

```
##  Length:1039901     Length:1039901     Length:1039901     Length:1039901
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Zip.Code           Rep_Dist             Area                Age
##  Min.   :     5301   Length:1039901     Length:1039901     Min.   : 17.0
##  1st Qu.:    64112   Class :character   Class :character   1st Qu.: 26.0
##  Median :    64127   Mode  :character   Mode  :character   Median : 35.0
##  Mean   :    67238                                         Mean   : 37.7
##  3rd Qu.:    64133                                         3rd Qu.: 47.0
##  Max.   :641303016                                         Max.   :100.0
##  NA's   :44630                                             NA's   :372484
##     Latitude        Longitude
##  Min.   :38.65    Min.   :-94.94
##  1st Qu.:39.02    1st Qu.:-94.58
##  Median :39.07    Median :-94.56
##  Mean   :39.07    Mean   :-94.55
##  3rd Qu.:39.11    3rd Qu.:-94.52
##  Max.   :39.89    Max.   :-94.07
##  NA's   :141839   NA's   :141839
```

```r
# Calculate percentage of NA values for each column
na_percentages <- colMeans(is.na(combined_data)) * 100

# Print the percentages
print(na_percentages)
```

```
## Reported_Date   Description       Address          City      Zip.Code
##    0.00000000    7.75977713    0.01240503    0.01702085    4.29175470
##      Rep_Dist          Area           Age      Latitude     Longitude
##   10.54773483    0.53293535   35.81917894   13.63966378   13.63966378
```

Now need to fix how date is being recorded.

Fixing the Date Issue

```r
# Convert all dates in the Date column to Date class
combined_data$Date <- as.Date(combined_data$Reported_Date, format = "%m/%d/%Y")

# verify the changes
print(combined_data[head(nrow(combined_data), 10), ])
```

```
##         Reported_Date         Description           Address        City
## 1039901    04/06/2024 Motor Vehicle Theft 00  W PERSHING RD KANSAS CITY
##         Zip.Code Rep_Dist Area Age Latitude Longitude       Date
## 1039901    64108   PJ1831  CPD  39 39.08459 -94.58673 2024-04-06
```

Noticing some dates are not withing the records of which I pulled so removing those because they are most likely errors

```
dates_outside_range <- combined_data$Date < as.Date("2015-01-01") |
                       combined_data$Date > as.Date("2024-12-31")

# Count the number of dates outside the range
num_dates_outside_range <- sum(dates_outside_range)

print(paste("Number of dates outside the range of 2015 to 2024:", num_dates_outside_range))
```

```
## [1] "Number of dates outside the range of 2015 to 2024: 128"
```

```
dates_outside_range_values <- combined_data$Date[which(dates_outside_range)]
rows_outside_range <- which(dates_outside_range)

crime_data_clean_date <- combined_data[!dates_outside_range, ]
```

Cleaning Complete Save File

```
summary(crime_data_clean_date)
```

```
##  Reported_Date      Description          Address              City
##  Length:1039773     Length:1039773     Length:1039773     Length:1039773
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Zip.Code           Rep_Dist             Area                Age
##  Min.   :     5301   Length:1039773     Length:1039773     Min.   : 17.0
##  1st Qu.:    64112   Class :character   Class :character   1st Qu.: 26.0
##  Median :    64127   Mode  :character   Mode  :character   Median : 35.0
##  Mean   :    67239                                         Mean   : 37.7
##  3rd Qu.:    64133                                         3rd Qu.: 47.0
##  Max.   :641303016                                         Max.   :100.0
##  NA's   :44615                                             NA's   :372410
##     Latitude        Longitude           Date
##  Min.   :38.65    Min.   :-94.94   Min.   :2015-01-01
##  1st Qu.:39.02    1st Qu.:-94.58   1st Qu.:2017-01-29
##  Median :39.07    Median :-94.56   Median :2019-01-27
##  Mean   :39.07    Mean   :-94.55   Mean   :2019-05-19
##  3rd Qu.:39.11    3rd Qu.:-94.52   3rd Qu.:2021-09-25
##  Max.   :39.89    Max.   :-94.07   Max.   :2024-04-07
##  NA's   :141833   NA's   :141833
```

```
# Calculate percentage of NA values for each column
na_percentages <- colMeans(is.na(crime_data_clean_date)) * 100
```

```r
# Print the percentages
print(na_percentages)
```

```
## Reported_Date   Description       Address          City      Zip.Code
##    0.00000000    7.75880889    0.01240655    0.01702295    4.29084040
##      Rep_Dist          Area           Age      Latitude     Longitude
##   10.54624423    0.53300095   35.81647148   13.64076582   13.64076582
##          Date
##    0.00000000
```

```r
write.csv(crime_data_clean_date, "crimedata_clean.csv")
```