IMT 573: Problem Set 3 - Working with Data Part 2

Srushti Chaukhande

Due: Tuesday, October 22, 2024 by 10:00AM PT

Collaborators:

Instructions: Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problem_set3.Rmd` file from Canvas. Open `problem_set3.Rmd` in RStudio and supply your solutions to the assignment by editing `problem_set3.Rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment. Collaboration shouldn't be confused with group project work (where each person does a part of the project). Working on problem sets should be your individual contribution.
3. Be sure to include well-documented (e.g. commented) code chucks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do not need four different visualizations of the same pattern.
4. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment. In particular, note that Stack Overflow is licenses as Creative Commons (CC-BY-SA). This means you have to attribute any code you refer from SO.
5. Partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. But please **DO NOT** submit pages and pages of hard-to-read code and attempts that is impossible to grade. That is, avoid redundancy. Remember that one of the key goals of a data scientist is to produce coherent reports that others can easily follow. Students are *strongly* encouraged to attempt each question and to document their reasoning process even if they cannot find the correct answer. If you would like to include R code to show this process, but it does not run without errors you can do so with the `eval=FALSE` option as follows:

```
a + b # these object dont' exist
# if you run this on its own it with give an error
```

6. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the knitted PDF file to `ps1_ourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.
7. Collaboration is often fun and useful, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.

Instructions

Revisiting COVID-19 Data

We are working with COVID-19 Data (which was also explored in Problem Set 2).

This dataset asks you to assemble and manipulate a COVID-19 dataset, and use it for a few illustrative figures. It replicates many real-worlds problems, including mismatching variable coding and differing variable names. We expect you to use dplyr-framework but you are welcome to use something else. Many questions also include hints and suggestions, these are designed to help you in case you do not have a good idea about how to proceed. But if you know better then you are welcome to follow other routes.

Most of the data is downloaded from John Hopkins university COVID-19 data project. The main variables in the monthly files are

FIPS US only. Federal Information Processing Standards code that uniquely identifies counties within the USA.

Admin2 County name. US only.

Province_State Province, state or dependency name.

Country_Region Country, region or sovereignty name. The names of locations included on the Website correspond with the official designations used by the U.S. Department of State. Confirmed Counts include confirmed and probable (where reported).

Deaths Counts include confirmed and probable (where reported).

This data is supplemented with information from the US Department of State, and Wikipedia. The data is on canvas: here.

Note: The following code will load the required initial information and make necessary modifications to help solve the problems in this problem set. Please ensure the unzipped folder 'covid' is placed in your current working directory.

```
# Load African countries list
africa <- read.delim("/Users/srushti/Downloads/covid/countries-africa.csv.bz2")
africanCountries <- africa$country

# Adjust the names of specific African countries for matching the COVID dataset
africanCountries[africanCountries == "Democratic Republic of the Congo"] <-
  "Congo (Kinshasa)"
africanCountries[africanCountries == "Republic of the Congo"] <- "Congo (Brazzaville)"
africanCountries[africanCountries == "Ivory Coast"] <- "Cote d'Ivoire"
```

1.1 Load and merge all datasets (20pt)

Now we load and merge all dataset for the complete covid-era. But before we get there, we need one more step of preparation: get month out of the file name.

1. (5pt) The file name is written as "covid-global_–.csv.bz2", and date always "01" in these files. Extract the date part from the first file name as Date object.

   Hint: check out as.Date and it's format argument. Format accepts the file name as a pattern, just you have to replace month with %m, date with %d, and year with %Y. You may also use gsub or other string replacement functions to remove the non-date part of the file name. See more the help file for strptime for time format patterns.

2. (15pt) Now it is time to merge all the data files into one. I recommend to proceed along these lines:

   - Create an empty final dataset
   - Loop over all the files. Inside the loop:
     - load the file
     - extract African countries only, and preserve only the number of deaths (we do not use other variables in this assignment). But note that variable may differ across different dataset!
     - extract year and month from the file name and add it to the extracted data. Above you extracted the date, check out lubridate::year and lubridate::month for how to extract year and month from a date.
     - merge the new dataset to the final dataset. Ensure you do not mess up the countries! Question: how should you merge these datasets?

   Hint: I got a dataset with 986 rows when I did all this.

Solution 1.1

1. **Solution:**

Insert Response

```
# Get first file in the list of files
files <- list.files(path = "/Users/srushti/Downloads/covid/", pattern =
                      "covid-global_.*\\.csv\\.bz2$", full.names = TRUE)
first_file <- files[1]
first_file <- basename(first_file)
date_string <- gsub("covid-global_|.csv.bz2", "", first_file)
print(date_string)
```

```
## [1] "01-01-2021"
```

```
# Convert the extracted string to a Date object
date_object <- as.Date(date_string, format = "%d-%m-%Y")
# Print the result
print(date_object)
```

```
## [1] "2021-01-01"
```

2. **Solution:** ###### Insert Response

```
final_dataset <- data.frame()
for (file in files) {
  date_string <- gsub("covid-global_|.csv.bz2", "", basename(file))
  date_object <- as.Date(date_string, format = "%m-%d-%Y")
  year_value <- year(date_object)
  month_value <- month(date_object)
  covid_dataset <- read.csv(file, sep = "\t")
  covid_dataset <- covid_dataset %>%
  rename_with(~ gsub("Country_Region|Country/Region|Country.Region", "Country_Region"
                     , .x, ignore.case = TRUE)) %>%
  rename_with(~ gsub("deaths|death", "Deaths", .x, ignore.case = TRUE))
```

```r
  # Filter for African countries and select relevant columns
  african_data <- covid_dataset %>%
    filter(Country_Region %in% africanCountries) %>%
    select(Country_Region, Deaths) %>%
    mutate(year = year_value, month = month_value)

  # Merge with final dataset
  final_dataset <- bind_rows(final_dataset, african_data)

}
# View the final dataset row number
nrow(final_dataset)
```

```
## [1] 986
```

1.2 Display time series (20pt)

Finally, let's see how has the number of COVID-19 deaths developed over time in Africa.

1. (4pt) Extract the population size from the dataset of African countries. Ensure the result is a valid number, you need to do some math with it next.
2. (2pt) For each country, compute the death rate: number of deaths per 1M population. Note: you have to merge the population data with the covid death data you compiled above.
3. (3pt) Which 10 countries have the largest death rate? (As of the latest date in the data, Oct 1st, 2021).
4. (4pt) Make a plot where you show how the death rate has grown in these 10 countries over time. Ensure the plot is appropriately labelled and uses appropriate plot type, colors, and other visual details.
5. (5pt) Let us also look at the number of monthly deaths: how much has the death rate grown from one month to another in these 10 countries? Compute the number of new monthly deaths (per 1M population) and display on a similar plot.
6. (2pt) Which country out of these 10 experienced the highest peak in the new monthly deaths? When was that? How many COVID "waves" can you see on the plot?

Solution 1.2

1. **Solution:**

Insert Response

```r
# Load the African countries dataset
african_countries <- read.csv("/Users/srushti/Downloads/covid/countries-africa.csv",
                              sep = "\t")

# Convert the population column to numeric
african_countries <- african_countries %>%
  mutate(population = as.numeric(gsub(",", "", population)))
```

2. **Solution:**

Insert Response

```
# Merge datasets on 'country'
merged_data <- final_dataset %>% left_join(african_countries, c("Country_Region"
                                                                = "country"))

# Calculate death rate per 1M population
merged_data <- merged_data %>% mutate(death_rate_per_1M = (Deaths / population) * 1e6)
```

3. **Solution:**

Insert Response

```
oct_data <- merged_data %>%
  filter(year.x == 2021 & month == 10)

top_10_countries <- oct_data %>%
  arrange(desc(death_rate_per_1M)) %>%
  slice_head(n = 10) %>%
  select(Country_Region, death_rate_per_1M)

print(top_10_countries)
```

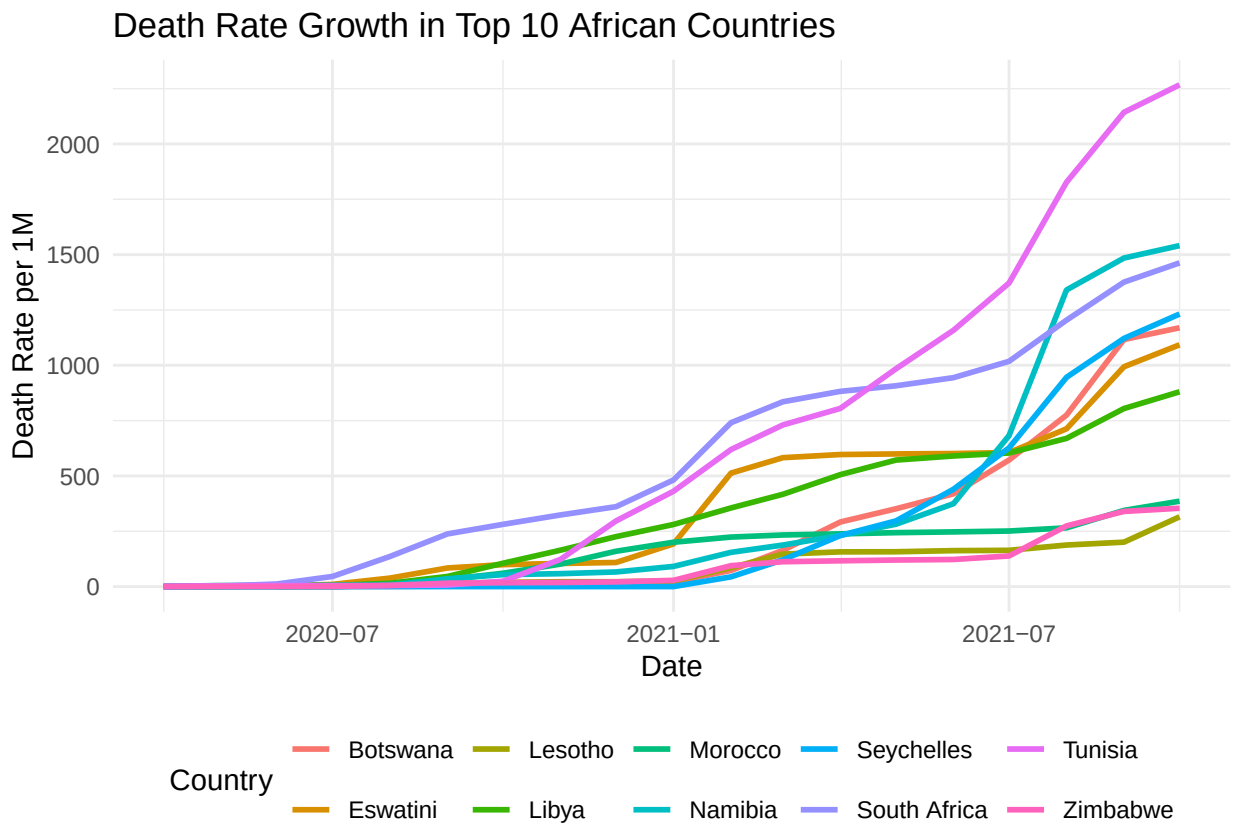
```
##    Country_Region death_rate_per_1M
## 1         Tunisia         2267.2820
## 2         Namibia         1540.7550
## 3    South Africa         1462.8027
## 4      Seychelles         1231.5136
## 5        Botswana         1169.4382
## 6        Eswatini         1092.5740
## 7           Libya          880.3069
## 8         Morocco          385.8540
## 9        Zimbabwe          354.0246
## 10        Lesotho          315.3645
```

4. **Solution:**

Insert Response

```
# Plot the death rate over time
top_10 <- top_10_countries$Country_Region
top_10_plot <- merged_data %>% filter(Country_Region %in% top_10)
top_10_plot_ <- top_10_plot %>%
  mutate(date = as.Date(paste(year.x, month, "01", sep = "-")))

# Plot the death rate over time for the top 10 countries
ggplot(top_10_plot_, aes(x = date, y = death_rate_per_1M, color = Country_Region)) +
  geom_line(size = 1) +
  labs(title = "Death Rate Growth in Top 10 African Countries",
       x = "Date",
       y = "Death Rate per 1M",
       color = "Country") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



5. **Solution:**

Insert Response

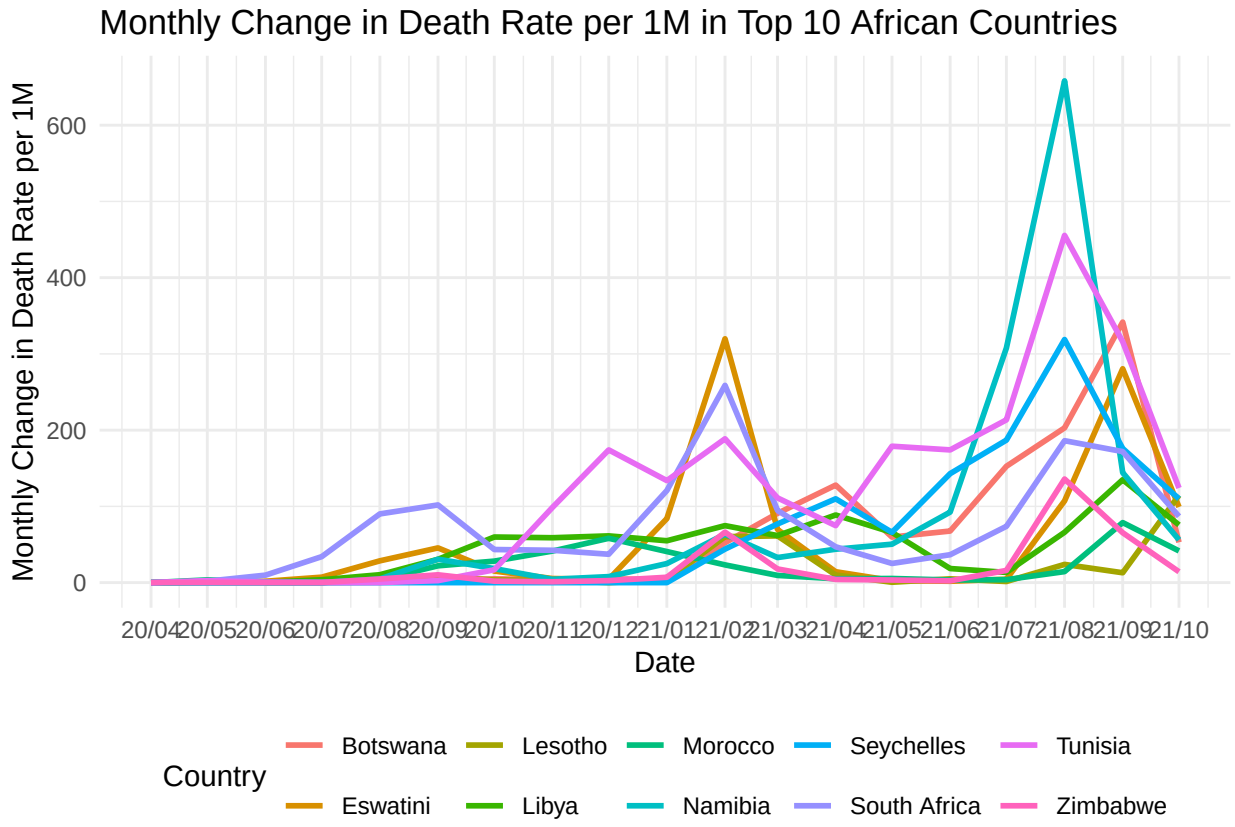
```
# Ensure data is sorted by country and date
top_10_plot <- top_10_plot %>%
  arrange(Country_Region, year.x, month)

# Calculate monthly change in death rate per 1M
top_10_plot <- top_10_plot %>%
  group_by(Country_Region) %>%
  mutate(monthly_death_rate_change = death_rate_per_1M - lag(death_rate_per_1M, default = first(death_ra

# Convert year and month to a date format for plotting
top_10_data <- top_10_plot %>%
  mutate(date = as.Date(paste(year.x, month, "01", sep = "-")))

# Plot the monthly change in death rate over time for the top 10 countries
ggplot(top_10_data, aes(x = date, y = monthly_death_rate_change, color = Country_Region)) +
  geom_line(size = 1) +
  labs(title = "Monthly Change in Death Rate per 1M in Top 10 African Countries",
       x = "Date",
       y = "Monthly Change in Death Rate per 1M",
       color = "Country") +
  scale_x_date(date_labels = "%y/%m", date_breaks = "1 month") +
```

```
  theme_minimal() +
  theme(legend.position = "bottom")
```



6. **Solution:**

Insert Response

1. **Namibia** has the highest peak in the new monthly deaths.
2. **Two** waves can be seen on the plot. Once around February 2021 and August 2021.

2. Census Data

Joining Census Data to Police Reports In this problem set, we will be joining disparate sets of data - namely: Seattle police crime data, information on Seattle police beats, and education attainment from the US Census. Our goal is to build a dataset where we can examine questions around crimes in Seattle and the educational attainment of people living in the areas in which the crime occurred; this requires data to be combined from these two individual sources.

As a general rule, be sure to keep copies of the original dataset(s) as you work through cleaning.

(a) (5 pts) Importing and Inspecting Crime Data Load the Seattle crime data from the provided `crime_data.csv` data file - Canvas file link. We will call this dataset the "Crime Dataset." Perform a basic inspection of the Crime Dataset and discuss what you find.

Solution:

Insert Response

```
crime_dataset <- read.csv("/Users/srushti/Downloads/crime_data.csv")

#Inspect the dataset and understand it's structure
head(crime_dataset)
```

```
##   Report.Number Occurred.Date Occurred.Time Reported.Date Reported.Time
## 1     1.975e+12    12/16/1975           900    12/16/1975          1500
## 2     1.976e+12    01/01/1976             1    01/31/1976          2359
## 3     1.979e+12    01/28/1979          1600    02/09/1979          1430
## 4     1.981e+13    08/22/1981          2029    08/22/1981          2030
## 5     1.981e+12    02/14/1981          2000    02/15/1981           435
## 6     1.988e+13    09/29/1988           155    09/29/1988           155
##      Crime.Subcategory  Primary.Offense.Description  Precinct Sector Beat
## 1 BURGLARY-RESIDENTIAL           BURGLARY-FORCE-RES     SOUTH      R   R3
## 2    SEX OFFENSE-OTHER    SEXOFF-INDECENT LIBERTIES   UNKNOWN
## 3            CAR PROWL               THEFT-CARPROWL      EAST      G   G2
## 4             HOMICIDE HOMICIDE-PREMEDITATED-WEAPON     SOUTH      S   S2
## 5 BURGLARY-RESIDENTIAL           BURGLARY-FORCE-RES SOUTHWEST      W   W3
## 6  MOTOR VEHICLE THEFT               VEH-THEFT-AUTO      WEST      M   M2
##                     Neighborhood
## 1           LAKEWOOD/SEWARD PARK
## 2                        UNKNOWN
## 3       CENTRAL AREA/SQUIRE PARK
## 4                BRIGHTON/DUNLAP
## 5 ROXHILL/WESTWOOD/ARBOR HEIGHTS
## 6                    SLU/CASCADE
```

```
summary(crime_dataset)
```

```
##  Report.Number       Occurred.Date      Occurred.Time  Reported.Date
##  Min.   :2.008e+08   Length:523591      Min.   :   0   Length:523591
##  1st Qu.:2.008e+13   Class :character   1st Qu.: 900   Class :character
##  Median :2.012e+13   Mode  :character   Median :1500   Mode  :character
##  Mean   :1.635e+13                      Mean   :1359
##  3rd Qu.:2.016e+13                      3rd Qu.:1920
##  Max.   :2.019e+13                      Max.   :2359
##                                         NA's   :2
##  Reported.Time  Crime.Subcategory  Primary.Offense.Description
##  Min.   :   0   Length:523591      Length:523591
##  1st Qu.: 950   Class :character   Class :character
##  Median :1407   Mode  :character   Mode  :character
##  Mean   :1353
##  3rd Qu.:1817
##  Max.   :2359
##  NA's   :2
##    Precinct            Sector              Beat           Neighborhood
##  Length:523591      Length:523591      Length:523591      Length:523591
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
##
##
##
##
```

```
#str(crime_dataset)

#Check for Missing Values
colSums(is.na(crime_dataset))
```

```
##               Report.Number               Occurred.Date
##                           0                           0
##               Occurred.Time               Reported.Date
##                           2                           0
##               Reported.Time           Crime.Subcategory
##                           2                           0
## Primary.Offense.Description                    Precinct
##                           0                           0
##                      Sector                        Beat
##                           0                           0
##                Neighborhood
##                           0
```

```
#Check number of unique crimes
num_unique_offenses <- crime_dataset %>%
  pull(Primary.Offense.Description) %>%
  unique() %>%
  length()
print(num_unique_offenses)
```

```
## [1] 144
```

(b) (5 pts) Looking at Years That Crimes Were Committed Let's start by looking at the years in which crimes were committed. What is the earliest year in the dataset? Are there any distinct trends with the annual number of crimes committed in the dataset?

Subset the data to only include crimes that were committed after 2011. Going forward, we will use this data subset.

Solution:

```
library(lubridate)
crime_data <- read_csv("/Users/srushti/Downloads/crime_data.csv",
                       na = c("", "NA"), show_col_types = FALSE)

crime_data <- crime_data %>%
  mutate(`Occurred Date` = mdy(`Occurred Date`))

min_year_row <- crime_data %>%
  arrange(`Occurred Date`) %>%
  filter(!is.na(`Occurred Date`))

print(min_year_row[1,])
```

```
## # A tibble: 1 x 11
##   `Report Number` `Occurred Date` `Occurred Time` `Reported Date`
##             <dbl> <date>                    <dbl> <chr>
## 1         2.01e13 1908-12-13                 2114 12/13/2008
## # i 7 more variables: `Reported Time` <dbl>, `Crime Subcategory` <chr>,
## #   `Primary Offense Description` <chr>, Precinct <chr>, Sector <chr>,
## #   Beat <chr>, Neighborhood <chr>
```

```r
#Subset data after 2011
filtered_crime_data <- crime_data %>% filter(`Occurred Date` > "2011-12-31")
nrow(filtered_crime_data)
```

```
## [1] 350053
```

Earliest year in the dataset is **1908**

(c) (5 pts) Looking at Frequency of Beats What is a Police Beat? How frequently are the beats in the Crime Dataset listed? Are there any anomolies with how frequently some of the beats are listed? Are there missing beats?

Solution:

A Police Beat refers to a specific geographic area that a police officer is assigned to patrol. It is the basic unit of territory for law enforcement agencies and is used to organize patrol operations. Officers assigned to a beat are responsible for responding to incidents, conducting routine patrols, and engaging with the community within that area.

```r
# Count the frequency of each beat
beat_frequency <- filtered_crime_data %>%
  count(Beat) %>%
  arrange(desc(n))

# Display the frequency of beats
print(beat_frequency)
```

```
## # A tibble: 60 x 2
##    Beat      n
##    <chr> <int>
##  1 K3    11611
##  2 M2    10210
##  3 E2    10200
##  4 U1    10157
##  5 L2    10049
##  6 M1     9883
##  7 M3     9723
##  8 B2     9253
##  9 Q3     9249
## 10 U3     9019
## # i 50 more rows
```

```r
#Display Missing beats
missing_beats <- filtered_crime_data %>% filter(is.na(Beat)) %>%
  summarise(missing_count = n())
print(missing_beats)
```

```
## # A tibble: 1 x 1
##   missing_count
##           <int>
## 1          2054
```

(d) (4 pts) Importing Police Beat Data and Filtering on Frequency Load the data on Seattle police beats provided in `police_beat_and_precinct_centerpoints.csv` - Canvas file link. We will call this dataset the "Beats Dataset."

Does the Crime Dataset include police beats that are not present in the Beats Dataset? If so, how many and with what frequency do they occur? Would you say that these comprise a large number of the observations in the Crime Dataset or are they rather infrequent? Do you think removing them would drastically alter the scope of the Crime Dataset?

Let's remove all instances in the Crime Dataset that have beats which occur fewer than 10 times across the Crime Dataset. Also remove any observations with missing beats. After only keeping years of interest and filtering based on frequency of the beat, how many observations do we now have in the Crime Dataset?

Solution:

Insert Response

```
beats_dataset <- read_csv(
  "/Users/srushti/Downloads/police_beat_and_precinct_centerpoints.csv"
                           , na = c("", "NA"), show_col_types = FALSE)

beat_ds_frequency <- beats_dataset %>%
  count(Name) %>%
  arrange(desc(n))

#Identify beats in Crime Dataset not present in Beats Dataset
unique_beats_crime <- filtered_crime_data %>%
  filter(!(Beat %in% beats_dataset$Name)) %>%
  count(Beat) %>%
  arrange(desc(n))

print(unique_beats_crime)
```

```
## # A tibble: 7 x 2
##   Beat      n
##   <chr> <int>
## 1 <NA>   2054
## 2 DET       7
## 3 S         4
## 4 CTY       1
## 5 K         1
## 6 SS        1
## 7 WS        1
```

```
# Count the frequency of each beat in the Crime Dataset
beat_counts <- filtered_crime_data %>%
  count(Beat) %>%
  filter(n >= 10)
```

```r
# Filter the Crime Dataset to include only beats that occur 10 or
# more times and are not missing
crime_data_row <- filtered_crime_data %>%
  filter(Beat %in% beat_counts$Beat & !is.na(Beat))

# Display the number of observations in the filtered dataset
num_observations <- nrow(crime_data_row)
print(num_observations)
```

```
## [1] 347980
```

I think these are infrequent and removing them won't alter the scope of the document by large.

(e) (6 pts) Importing and Inspecting Police Beat Data To join the Beat Dataset to census data, we must have census tract information. Use the `tigirs` package to extract the 15-digit census tract for each police beat using the corresponding latitude and longitude. Do this using each of the police beats listed in the Beats Dataset. Do not use a for-loop for this but instead rely on R functions (e.g. the 'apply' family of functions). Add a column to the Beat Dataset that contains the 15-digit census tract for the each beat. (HINT: you may find `tigris`'s `call_geolocator_latlon` function useful)

We will eventually join the Beats Dataset to the Crime Dataset. We could have joined the two and then found the census tracts for each beat. Would there have been a particular advantage/disadvantage to doing this join first and then finding census tracts? If so, what is it? (NOTE: you do not need to write any code to answer this)

Solution:

Insert Response

```r
census_tract <- read_csv("/Users/srushti/Downloads/census_edu_data.csv",
                         na = c("", "NA"), show_col_types = FALSE)

#use function from tigris package and mapply
beats_dataset$census_tract <- mapply(call_geolocator_latlon, beats_dataset$Latitude,
                                     beats_dataset$Longitude)
```

(f) (6 pts) Extracting FIPS Codes Once we have the 15-digit census codes, we will break down the code based on information of interest. You can find more information on what these 15 digits represent here: https://transition.fcc.gov/form477/Geo/more_about_census_blocks.pdf.

Solution:

Insert Response

Extracting 11 digit FIPS code involves : *State, County and Tract codes and leaving out the block code* States and the territories are identified by a 2-digit code. *Counties within states are identified by a 3-digit code.* Tracts within counties are identified 6-digit code. *Blocks within tracts are identified by a 4-digit code.* We can extract 11 digits using substr function

(g) (6 pts) Extracting 11-digit Codes The census data uses an 11-digit code that consists of the state, county, and tract code. It does not include the block code. To join the census data to the Beats Dataset, we must have this code for each of the beats. Extract the 11-digit code for each of the beats in the Beats Dataset. The 11 digits consist of the 2 state digits, 3 county digits, and 6 tract digits. Add a column with the 11-digit code for each beat.

Solution:

Insert Response

```
beats_data <- beats_dataset %>%
  mutate(census_11 = substr(census_tract, 1, 11))

# Display the updated dataset
head(beats_data)
```

```
## # A tibble: 6 x 6
##   Name  `Location 1`                    Latitude Longitude census_tract census_11
##   <chr> <chr>                              <dbl>     <dbl> <chr>        <chr>
## 1 B1    (47.7097756394592, -122.37099~     47.7     -122. 53033001400~ 53033001~
## 2 B2    (47.6790521901374, -122.39174~     47.7     -122. 53033003202~ 53033003~
## 3 B3    (47.6812920482227, -122.36423~     47.7     -122. 53033002900~ 53033002~
## 4 C1    (47.6342500180223, -122.31568~     47.6     -122. 53033006500~ 53033006~
## 5 C2    (47.6192385752996, -122.31355~     47.6     -122. 53033007502~ 53033007~
## 6 C3    (47.6300792887474, -122.29208~     47.6     -122. 53033006300~ 53033006~
```

(h) (5 pts) Extracting 11-digit Codes From Census Now, we will examine census data provided om `census_edu_data.csv` - Canvas file link. The data includes counts of education attainment across different census tracts. Note how this data is in a 'wide' format and how it can be converted to a 'long' format. For now, we will work with it as is.

The census data contains a `GEO.id` column. Among other things, this variable encodes the 11-digit code that we had extracted above for each of the police beats. Specifically, when we look at the characters after the characters "US" for values of `GEO.id`, we see encodings for state, county, and tract, which should align with the beats we had above. Extract the 11-digit code from the `GEO.id` column. Add a column to the census data with the 11-digit code for each census observation.

Solution:

Insert Response

```
# Extract the 11-digit code from 'GEO.id' by removing the 'US' prefix and add
# a new column
census_tract <- census_tract %>%
  mutate(census_11 = substr(GEO.id, 10, 20))

# Display the first few rows to verify the new column
head(census_tract)
```

```
## # A tibble: 6 x 29
##   GEO.id           GEO.id2 `GEO.display-label` total no_schooling nursery_school
##   <chr>              <dbl> <chr>               <dbl>        <dbl>          <dbl>
## 1 1400000US530330~ 5.30e10 Census Tract 1, Ki~  5708           82              0
## 2 1400000US530330~ 5.30e10 Census Tract 2, Ki~  6079          115              0
## 3 1400000US530330~ 5.30e10 Census Tract 3, Ki~  2152           49              0
## 4 1400000US530330~ 5.30e10 Census Tract 4.01,~  5084           60              0
## 5 1400000US530330~ 5.30e10 Census Tract 4.02,~  4498           60              0
## 6 1400000US530330~ 5.30e10 Census Tract 5, Ki~  2333            6              9
## # i 23 more variables: kindergarten <dbl>, `1st_grade` <dbl>,
## #   `2nd_grade` <dbl>, `3rd_grade` <dbl>, `4th_grade` <dbl>, `5th_grade` <dbl>,
```

```
## #   `6th_grade` <dbl>, `7th_grade` <dbl>, `8th_grade` <dbl>, `9th_grade` <dbl>,
## #   `10th_grade` <dbl>, `11th_grade` <dbl>, `12th_grade_no_diploma` <dbl>,
## #   high_school_diploma <dbl>, ged_or_alternative_credential <dbl>,
## #   some_college_less_than_1_year <dbl>,
## #   some_college_1_or_more_years_no_degree <dbl>, associates_degree <dbl>, ...
```

(i) (10 pts) Join Datasets Join the census data with the Beat Dataset using the 11-digit codes as keys. Be sure that you do not lose any of the police beats when doing this join (i.e. your output dataframe should have the same number of rows as the cleaned Beats Dataset - use the correct join). Are there any police beats that do not have any associated census data? If so, how many?

Then, join the Crime Dataset to our joined beat/census data. We can do this using the police beat name. Again, be sure you do not lose any observations from the Crime Dataset. What is the final dimensions of the joined dataset?

Solution:

Insert Response

```
#join census data with beats dataset
beats_data_joined <- beats_data %>% left_join(census_tract, by="census_11")
head(beats_data_joined)
```

```
## # A tibble: 6 x 34
##   Name  `Location 1`   Latitude Longitude census_tract census_11 GEO.id  GEO.id2
##   <chr> <chr>             <dbl>     <dbl> <chr>        <chr>     <chr>     <dbl>
## 1 B1    (47.709775639~     47.7     -122. 53033001400~ 53033001~ 14000~  5.30e10
## 2 B2    (47.679052190~     47.7     -122. 53033003202~ 53033003~ <NA>   NA
## 3 B3    (47.681292048~     47.7     -122. 53033002900~ 53033002~ 14000~  5.30e10
## 4 C1    (47.634250018~     47.6     -122. 53033006500~ 53033006~ 14000~  5.30e10
## 5 C2    (47.619238575~     47.6     -122. 53033007502~ 53033007~ <NA>   NA
## 6 C3    (47.630079288~     47.6     -122. 53033006300~ 53033006~ 14000~  5.30e10
## # i 26 more variables: `GEO.display-label` <chr>, total <dbl>,
## #   no_schooling <dbl>, nursery_school <dbl>, kindergarten <dbl>,
## #   `1st_grade` <dbl>, `2nd_grade` <dbl>, `3rd_grade` <dbl>, `4th_grade` <dbl>,
## #   `5th_grade` <dbl>, `6th_grade` <dbl>, `7th_grade` <dbl>, `8th_grade` <dbl>,
## #   `9th_grade` <dbl>, `10th_grade` <dbl>, `11th_grade` <dbl>,
## #   `12th_grade_no_diploma` <dbl>, high_school_diploma <dbl>,
## #   ged_or_alternative_credential <dbl>, ...
```

```
# police beats that do not have any associated
missing_beats_data <- beats_data_joined %>% filter(is.na(GEO.id))
nrow(missing_beats_data)
```

```
## [1] 24
```

```
#join the Crime Dataset to new beats data
crime_data_joined <- crime_data_row %>% left_join(beats_data_joined, c("Beat"="Name"))

#Final dimension of dataset
print("Dimensions of crime dataset : ")
```

```
## [1] "Dimensions of crime dataset : "
```

```r
nrow(crime_data_joined)
```

```
## [1] 347980
```

```r
ncol(crime_data_joined)
```

```
## [1] 44
```

```r
#saving dataset for future use
write.csv(crime_data_joined, "/Users/srushti/Downloads/crime_data_joined.csv", row.names = FALSE)
```

Once everything is joined, save the final dataset for future use.

References

- Mitra, T. (2024, October 8). Working with data part I: Data integration [Lecture slides]. IMT 573A - Data Science 1 - Theoretical Foundations. University of Washington.
- Mitra, T. (2024, October 17). Working with data part II: Data integration [Lecture slides]. IMT 573A - Data Science 1 - Theoretical Foundations. University of Washington.