IMT 573: Problem Set 3 - Working With Data II

Vighnesh Misal

Due: Tuesday, October 22, 2019

Collaborators: Ashish Anand, Gurleen Gujral

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Server.

1. Download the `problemset3.Rmd` file from Canvas or save a copy to your local directory on RStudio Server. Open `problemset3.Rmd` in RStudio and supply your solutions to the assignment by editing `problemset3.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.
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. Collaboration on problem sets is fun and useful, and we encourage it, 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.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. Remember partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. 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
```

7. 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 `ps3_YourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library('dplyr')
```

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
library('censusr')
```

```
## Warning: package 'censusr' was built under R version 3.6.1
```

```
library('stringr')
```

```
## Warning: package 'stringr' was built under R version 3.6.1
```

```
library('tidyverse')
```

```
## Warning: package 'tidyverse' was built under R version 3.6.1
```

```
## Warning: package 'purrr' was built under R version 3.6.1
```

Problem 1: 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 ultimate 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.

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

(a) Importing and Inspecting Crime Data

Load the Seattle crime data (crime_data.csv). You can find more information on the data here: (https://data.seattle.gov/Public-Safety/Crime-Data/4fs7-3vj5). This dataset is constantly refreshed online so we will be using the csv file for consistency. We will henceforth call this dataset the "Crime Dataset." Perform a basic inspection of the Crime Dataset and discuss what you find.

```
crime_data <- read.csv("Crime_Data.csv")
crime_df <- tbl_df(crime_data)
nrow(crime_df)
```

```
## [1] 523591
```

```
colnames(crime_df)
```

```
##  [1] "Report.Number"               "Occurred.Date"
##  [3] "Occurred.Time"               "Reported.Date"
##  [5] "Reported.Time"               "Crime.Subcategory"
##  [7] "Primary.Offense.Description" "Precinct"
##  [9] "Sector"                      "Beat"
## [11] "Neighborhood"
```

```
str(crime_df)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    523591 obs. of  11 variables:
##  $ Report.Number              : num  1.98e+12 1.98e+12 1.98e+12 1.98e+13 1.98e+12 ...
##  $ Occurred.Date              : Factor w/ 4621 levels "","01/01/1973",..: 4418 4 385 2989 605 3448 35
##  $ Occurred.Time              : int  900 1 1600 2029 2000 155 2213 0 1130 NA ...
##  $ Reported.Date              : Factor w/ 4299 levels "01/01/1999","01/01/2006",..: 4111 371 481 2773
##  $ Reported.Time              : int  1500 2359 1430 2030 435 155 2213 844 1700 NA ...
##  $ Crime.Subcategory          : Factor w/ 31 levels "","AGGRAVATED ASSAULT",..: 7 25 9 14 7 17 14 26
##  $ Primary.Offense.Description: Factor w/ 144 levels "ADULT-VULNERABLE-FINANCIAL",..: 19 113 124 44 1
##  $ Precinct                   : Factor w/ 7 levels "","EAST","NORTH",..: 4 6 2 4 5 7 4 5 4 6 ...
##  $ Sector                     : Factor w/ 24 levels "","6804","9512",..: 18 1 9 19 23 14 18 8 16 1 .
##  $ Beat                       : Factor w/ 65 levels "","B1","B2","B3",..: 51 1 21 54 63 37 50 17 43 1
##  $ Neighborhood               : Factor w/ 59 levels "ALASKA JUNCTION",..: 29 58 9 7 50 52 11 24 53 58
```

```
head(crime_df)
```

```
## # A tibble: 6 x 11
##   Report.Number Occurred.Date Occurred.Time Reported.Date Reported.Time
##           <dbl> <fct>                <int> <fct>                 <int>
## 1       1.98e12 12/16/1975             900 12/16/1975             1500
## 2       1.98e12 01/01/1976               1 01/31/1976             2359
## 3       1.98e12 01/28/1979            1600 02/09/1979             1430
## 4       1.98e13 08/22/1981            2029 08/22/1981             2030
## 5       1.98e12 02/14/1981            2000 02/15/1981              435
## 6       1.99e13 09/29/1988             155 09/29/1988              155
## # ... with 6 more variables: Crime.Subcategory <fct>,
## #   Primary.Offense.Description <fct>, Precinct <fct>, Sector <fct>,
## #   Beat <fct>, Neighborhood <fct>
```

```
tail(crime_df)
```

```
## # A tibble: 6 x 11
##   Report.Number Occurred.Date Occurred.Time Reported.Date Reported.Time
##           <dbl> <fct>                <int> <fct>                 <int>
## 1 2019000099916 03/20/2019            1330 03/20/2019             1654
## 2 2019000099944 03/20/2019            1713 03/20/2019             1713
## 3 2019000099946 03/20/2019             730 03/20/2019             1721
## 4 2019000099949 03/20/2019            1724 03/20/2019             1724
## 5 2019000099974 03/20/2019            1750 03/20/2019             1904
## 6 2019000099993 03/19/2019            1800 03/20/2019             2237
## # ... with 6 more variables: Crime.Subcategory <fct>,
## #   Primary.Offense.Description <fct>, Precinct <fct>, Sector <fct>,
## #   Beat <fct>, Neighborhood <fct>
```

The data has 523591 records

It has 11 columns:

Report.Number (numeric) : Uniquely identify each incident

Occurred.Data (factor) : Date that the crime occurred

Occurred.Time (factor) : Time that the crime occurred

Reported.Date (factor) : Date that the crime was reported

Reported.Time (int) : Time that the crime was reported

Crime.Subcategory (factor) : The type of crime that was committed

Primary.Offense.Description (factor) : Description of offense

Precint : Code to identify precint

Sector : Code to identify the sector

Beat : Code to identify the beat

Neighbourhood : Identifies the neighbourhood where the crime occured.

The dates and time are not in the corrct date or time format which will prove to be a problem for analysis and need to be converted.

(b) 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?

```
crime_df$Occurred.Date = as.Date(crime_df$Occurred.Date, format = '%m/%d/%Y')

crime_df <- crime_df %>%
  mutate(Occurred.Date.Year = lubridate::year(Occurred.Date))

min(crime_df$Occurred.Date.Year, na.rm = TRUE)
```

```
## [1] 1908
```

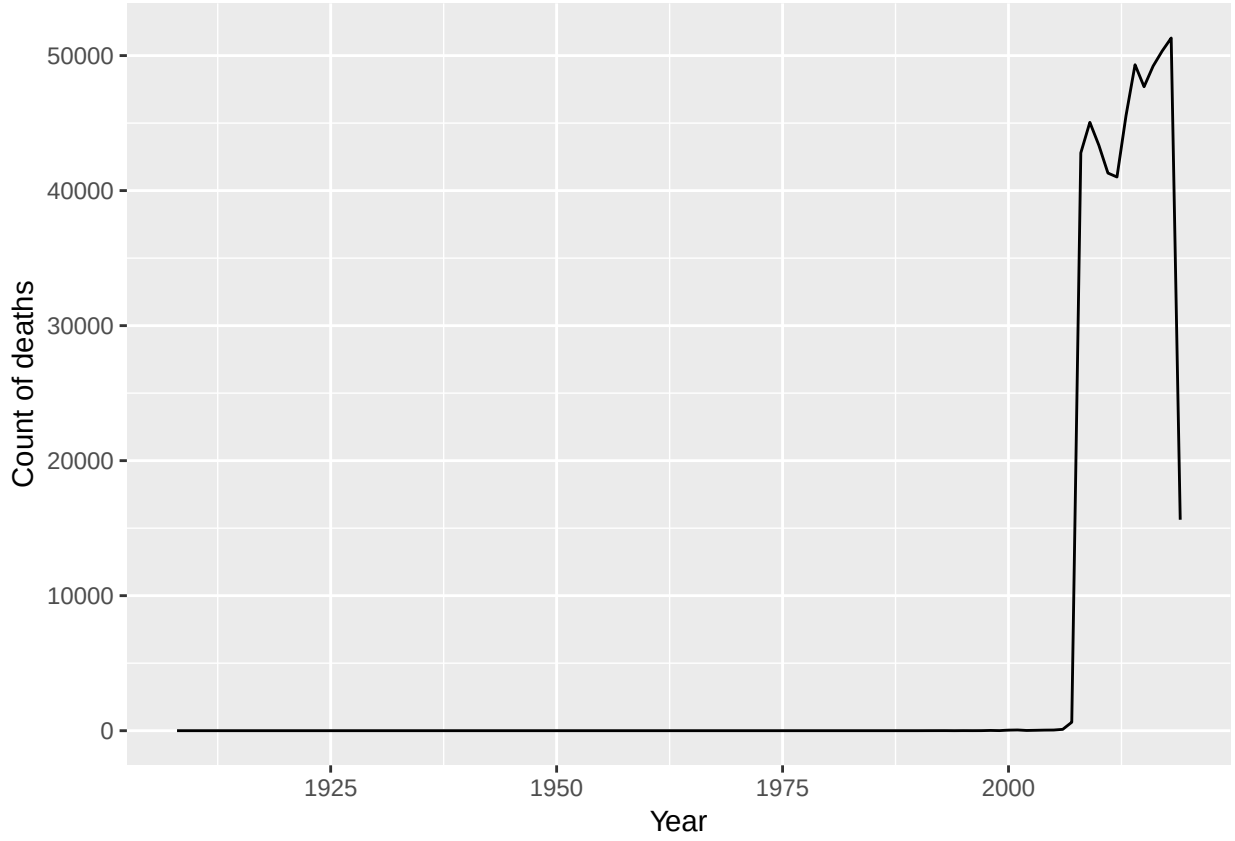
The earliest year in which a crime was reported is 1908

```
crime_year<- crime_df %>%
  group_by(Occurred.Date.Year) %>%
```

```
  summarise(total = n()) %>%
  arrange(desc(total))


ggplot(crime_year, mapping = aes(x = Occurred.Date.Year, y = total, group = 1)) + geom_line() + xlab('Y
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



The numner of crimes show a suden spike in the number of crimes for and after the year 2006. THere is a drop in crimes for the last year present, but this may be due to not having data for the complete year.

Let's subset the data to only include crimes that were committed after 2011 (remember good practices of data provenance!). Going forward, we will use this data subset.

```
filter_data <- crime_df %>%
  filter(Occurred.Date.Year > '2011')
```

(c) Looking at Frequency of Beats

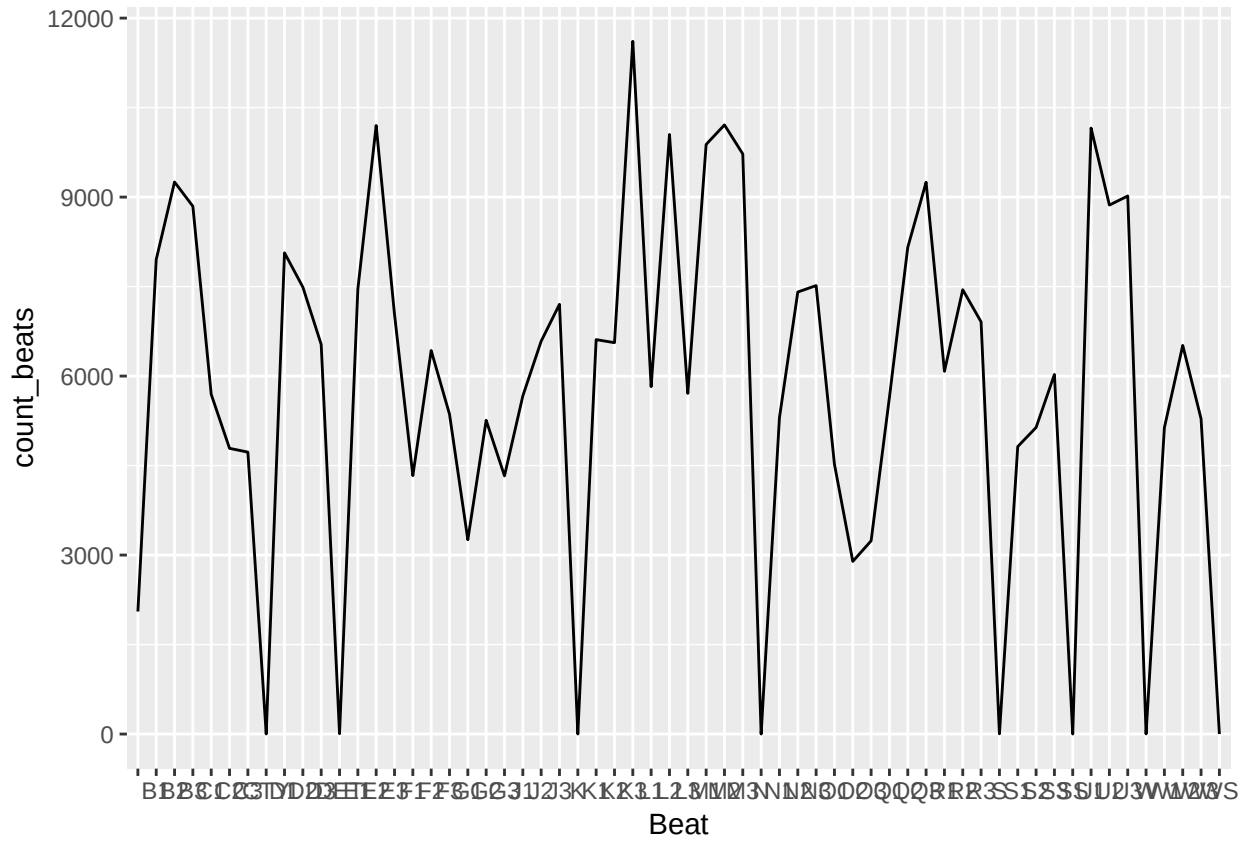
How frequently are the beats in the Crime Dataset listed? Are there any anomolies with how frequently some of the beats are listed? Are there missing beats?

```
count_beat <- filter_data %>%
  group_by(Beat)%>%
  summarise(count_beats = n())
count_beat
```

```
## # A tibble: 60 x 2
##    Beat  count_beats
```

```
##    <fct>       <int>
##  1 ""           2054
##  2 B1           7954
##  3 B2           9253
##  4 B3           8846
##  5 C1           5694
##  6 C2           4789
##  7 C3           4726
##  8 CTY             1
##  9 D1           8066
## 10 D2           7491
## # ... with 50 more rows
```

```
ggplot(count_beat, aes(x=Beat, y=count_beats, group = 1)) + geom_line()
```




```
sum(filter_data$Beat == '')
```

```
## [1] 2054
```

We can observe from the frequency table of beat counts that beats K3, M2, E2 are much more frequent in the dataset.

We can also observe from the beats frequency table that 2054 records have missing beats values

(d) Importing Police Beat Data and Filtering on Frequency

Load the data on Seattle police beats (police_beat_and_precinct_centerpoints.csv). You can find additional information on the data here: (https://data.seattle.gov/Land-Base/Police-Beat-and-Precinct-Centerpoints/4khs-fz35) and more information on what a police beat is here: https://en.wikipedia.org/wiki/Beat_(police). We will henceforth call this dataset the "Beats Dataset."

```
beat_data <- read.csv('Police_Beat_and_Precinct_Centerpoints.csv')
beats_dataset <- tbl_df(beat_data)
```

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?

```
missing_from_Beats_dataset <- subset(beats_dataset, !(beats_dataset$Name %in% filter_data$Beat))
missing_from_Crime_dataset <- subset(count_beat, !(count_beat$Beat %in% beats_dataset$Name))

missing_from_Beats_dataset$Name
```

```
## [1] CITYWIDE E        SE       SW
## 57 Levels: B1 B2 B3 C1 C2 C3 CITYWIDE D1 D2 D3 E E1 E2 E3 F1 F2 F3 ... W3
```

```
missing_from_Crime_dataset$Beat
```

```
## [1]     CTY DET K   S   SS  WS
## 65 Levels:  B1 B2 B3 C1 C2 C3 CS CTY D1 D2 D3 DET E1 E2 E3 F1 F2 F3 ... X9
```

```
missing_from_Crime_dataset
```

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

There are 8 distince missing beat values in the crime dataset that aren't present in the beats dataset.

There are also 2054 records that have no values recorded in the Beat column.

The missing beat form a very small percentage of the total number of records and can thus be disregarded from further analysis.

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?

```
count_beat1 <- count_beat %>%
  filter(Beat != '') %>%
  filter(count_beats >= 10)

joined <- inner_join(beats_dataset,filter_data, by = c("Name" = "Beat"))
```

```
## Warning: Column `Name`/`Beat` joining factors with different levels,
## coercing to character vector
```

```
cleaned_crime_dataset <- subset(joined, (joined$Name %in% count_beat1$Beat))
nrow(cleaned_crime_dataset)
```

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

There are now 347980 records present in the dataset.

(e) Importing and Inspecting Police Beat Data

To join the Beat Dataset to census data, we must have census tract information.

First, let's remove the beats in the Beats Dataset that are not listed in the (cleaned) Crime Dataset.

Then, let's use the censusr 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 censusr's call_geolocator_latlon function useful)

```
coords <- data.frame(lat=beats_dataset$Latitude,lon=beats_dataset$Longitude)

beats_dataset %>% mutate(Census_Tract=apply(coords, 1, function(row) call_geolocator_latlon(row['lat'],
```

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)

It was particularly difficult to that since beats table has count values and performaing a join causes these values to be repeated in the joined data frame which might lead to potentially incorrect results and perception of what the data is about by a user who is working directly with the new joined data frame.

(f) 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.

First, create a column that contains the state code for each beat in the Beats Dataset. Then create a column that contains the county code for each beat. Find the FIPS codes for WA State and King County (the county of Seattle) online. Are the extracted state and county codes what you would expect them to be? Why or why not?

```
beats_dataset <- beats_dataset %>%
  mutate(state_code = substr(Census_Tract, 1, 2),
         country_code = substr(Census_Tract, 3,5))
```

The Washington State Code is 53 and the King County code is 53033. The extraced codes are exactly what I expect them to be. The 15 digit FIPS code identifies the State, County, Tract and Block. The extracted substrings from the FIPS code correspond with the ones for Washington State and King's County.

(g) 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.

```
beats_dataset <- beats_dataset %>%
  mutate(Code_combined = substr(Census_Tract, 1, 11))
```

(h) Extracting 11-digit Codes From Census

Now, we will examine census data (census_edu_data.csv). 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.

```
census_data <- read.csv('census_edu_data.csv')
census_data_df <- tbl_df(census_data)
```

```
census_data_df<- census_data_df %>%
  mutate(digit_code_11=str_sub(census_data_df$GEO.id ,-11,-1))
```

(i) 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?

```
beats_census_joined <- inner_join(beats_dataset,census_data_df, by = c("Code_combined" = "digit_code_11

beats_census_joined
```

```
## # A tibble: 57 x 36
##    Name  Location.1 Latitude Longitude Census_Tract state_code country_code
##    <fct> <fct>         <dbl>     <dbl> <chr>        <chr>      <chr>
##  1 B1    (47.70977~     47.7     -122. 53033001400~ 53         033
##  2 B2    (47.67905~     47.7     -122. 53033003200~ 53         033
##  3 B3    (47.68129~     47.7     -122. 53033002900~ 53         033
##  4 C1    (47.63425~     47.6     -122. 53033006500~ 53         033
##  5 C2    (47.61923~     47.6     -122. 53033007500~ 53         033
##  6 C3    (47.63007~     47.6     -122. 53033006300~ 53         033
##  7 CITY~ (47.62100~     47.6     -122. 53033007300~ 53         033
##  8 D1    (47.62744~     47.6     -122. 53033006700~ 53         033
##  9 D2    (47.62565~     47.6     -122. 53033006600~ 53         033
## 10 D3    (47.61034~     47.6     -122. 53033008300~ 53         033
## # ... with 47 more rows, and 29 more variables: Code_combined <chr>,
## #   GEO.id <fct>, GEO.id2 <dbl>, GEO.display.label <fct>, total <int>,
## #   no_schooling <int>, nursery_school <int>, kindergarten <int>,
## #   X1st_grade <int>, X2nd_grade <int>, X3rd_grade <int>,
## #   X4th_grade <int>, X5th_grade <int>, X6th_grade <int>,
## #   X7th_grade <int>, X8th_grade <int>, X9th_grade <int>,
## #   X10th_grade <int>, X11th_grade <int>, X12th_grade_no_diploma <int>,
## #   high_school_diploma <int>, ged_or_alternative_credential <int>,
## #   some_college_less_than_1_year <int>,
## #   some_college_1_or_more_years_no_degree <int>, associates_degree <int>,
## #   bachelors_degree <int>, masters_degree <int>,
## #   professional_school_degree <int>, doctorate_degree <int>
```

No. THere are no police beats that lack census data as we got the same number of rows that we got from performig the previous join. Hence there are no beats that don't have any associated census data.

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?

```
final <- inner_join(filter_data,beats_census_joined,by = c("Beat"="Name"))
```

```
## Warning: Column `Beat`/`Name` joining factors with different levels,
## coercing to character vector
```

```r
nrow(final)
```

```
## [1] 347984
```

The final data frame has 347980 observations.

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