IMT 573: Problem Set 4

Working with Data: Part II

Ali Qazi

Due: July 17th, 2022

Collaborators: Akeel Qazi, Anthony Mercado

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 Cloud.

1. Download the `04_ps_workingdatatwo.Rmd` file from Canvas or save a copy to your local directory on RStudio Cloud. Supply your solutions to the assignment by editing `04_ps_workingdatatwo.Rmd`.
2. Replace the "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 no 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 don't 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`, download and rename the knitted PDF file to `ps4_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(tidyverse)
library(censusr)
```

```
library(dplyr)
library(stringr)
library(tigris) # for geolocator
library(tidycensus)
```

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 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 (remember data provenance!).

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

```
inspect_data = read.csv("crime_data.csv")
```

Looking at the raw data, I can see the report number, date, and time. In addition the data set includes the occurred time, date, neighborhood, the type of crime, and more.

(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?

```
ds_year <- group_by(inspect_data, year = substr(inspect_data$Occurred.Date,7,10))
year <- summarise(ds_year, count = n())
year
```

```
## # A tibble: 46 x 2
##    year   count
##    <chr>  <int>
##  1 ""         2
##  2 "1908"     1
##  3 "1964"     1
##  4 "1973"     1
##  5 "1974"     1
##  6 "1975"     2
##  7 "1976"     2
##  8 "1977"     1
##  9 "1978"     1
## 10 "1979"     2
## # ... with 36 more rows
```

The earliest year in the data set is 1908. During the first few years of the data set, the number of crimes are low. There is a huge rise in 2008. This may be due to better data collection methods.

```
# Using data set from 2008 and later to practice data provenance
ds <- filter(ds_year, year >= 2008)
```

(c) 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?

```
ds_beat <- group_by(ds, Name = Beat)
ds_beat <- summarise(ds_beat, count= n())

ds_beat
```

```
## # A tibble: 65 x 2
##    Name  count
##    <chr> <int>
##  1 ""     3213
##  2 "B1"  11131
##  3 "B2"  13759
##  4 "B3"  13034
##  5 "C1"   8271
##  6 "C2"   6866
##  7 "C3"   7424
##  8 "CS"      1
##  9 "CTY"     2
## 10 "D1"  13202
## # ... with 55 more rows
```

Beat is the territory that a police officer patrols. There are many beats that are more frequent compared to others. There is a missing beat in the first row. Some beat that are not frequent and appear once while some appear over 13,000 times.

(d) Importing Police Beat Data and Filtering on Frequency Load the data on Seattle police beats provided in `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). We will call this dataset the "Beats Dataset."

```
inspect_raw_beat_ds = read.csv("police_beat_and_precinct_centerpoints.csv")
```

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?

```
beat_leftjoin <- left_join(ds_beat, inspect_raw_beat_ds, by = "Name")

beat_missing <- filter(beat_leftjoin, is.na(Latitude) == TRUE)

select(beat_missing, Name, count)
```

```
## # A tibble: 12 x 2
##    Name   count
##    <chr>  <int>
```

```
##  1 ""      3213
##  2 "CS"       1
##  3 "CTY"      2
##  4 "DET"      9
##  5 "H1"       1
##  6 "INV"      2
##  7 "K"        1
##  8 "LAPT"     1
##  9 "S"        9
## 10 "SS"       2
## 11 "WS"       1
## 12 "X9"       3
```

The crime dataset include more police beats than presented in the beats dataset. There are 3213 missing beats just like the ds_beats data in the first row as well. The beats in this dataset are far less frequent. Since the frequency of each beat is very small I do not 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?

```
beat_ten <- filter(beat_leftjoin, count <= 10)
ds_cleaned <- filter(ds, Beat != "CTY",Beat != "DET", Beat != "INV",
Beat != "K",Beat != "N",Beat != "S",Beat != "SS",Beat != "W",Beat != "WS",Beat != "")
ds_cleaned
```

```
## # A tibble: 519,305 x 12
## # Groups:   year [12]
##    Report.Number Occurred.Date Occurred.Time Reported.Date Reported.Time
##            <dbl> <chr>                <int> <chr>                 <int>
##  1       2.01e13 03/17/2008            1000 03/17/2008             2245
##  2       2.01e12 01/08/2008             800 01/08/2008             1925
##  3       2.01e13 03/17/2008            2322 03/17/2008             2327
##  4       2.01e13 03/17/2008            2030 03/17/2008             2338
##  5       2.01e13 03/17/2008            2339 03/17/2008             2339
##  6       2.01e13 03/18/2008              41 03/18/2008               41
##  7       2.01e12 01/08/2008            1915 01/08/2008             1930
##  8       2.01e12 01/08/2008            1800 01/08/2008             1925
##  9       2.01e13 03/18/2008             200 03/18/2008              204
## 10       2.01e13 03/18/2008             205 03/18/2008              205
## # ... with 519,295 more rows, and 7 more variables: Crime.Subcategory <chr>,
## #   Primary.Offense.Description <chr>, Precinct <chr>, Sector <chr>,
## #   Beat <chr>, Neighborhood <chr>, year <chr>
```

There are now 519,305 observations (after 2008).

(e) Importing and Inspecting Police Beat Data To join the Beat Dataset to census data, we must have census tract information. 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)

```r
# remove beats not in crime data set
rbeat <- left_join(inspect_raw_beat_ds, beat_ten, by = "Name")
rbeat <- filter(rbeat, is.na(rbeat$count) == TRUE)
rbeat <- select(rbeat, Name, Location = Location.1.x, Latitude = Latitude.x, Longitude =Longitude.x)

func <- function(Lat,Lon){return(call_geolocator_latlon(Lat,Lon))}

beat_ct <- mutate(rbeat, census_tract = mapply(func, rbeat$Latitude,rbeat$Longitude))

beat_ct
```

```
##        Name                              Location Latitude Longitude
## 1        B1 (47.7097756394592, -122.370990523069) 47.70978 -122.3710
## 2        B2 (47.6790521901374, -122.391748391741) 47.67905 -122.3918
## 3        B3 (47.6812920482227, -122.364236159741) 47.68129 -122.3642
## 4        C1 (47.6342500180223, -122.315684762418) 47.63425 -122.3157
## 5        C2 (47.6192385752996, -122.313557430551) 47.61924 -122.3136
## 6        C3 (47.6300792887474, -122.292087128251) 47.63008 -122.2921
## 7  CITYWIDE (47.6210041048652, -122.332993498998) 47.62100 -122.3330
## 8        D1 (47.6274421308028, -122.345705781837) 47.62744 -122.3457
## 9        D2 (47.6256548876049, -122.331370005506) 47.62565 -122.3314
## 10       D3 (47.6103493249325, -122.328653706199) 47.61035 -122.3286
## 11        E (47.6201542748144, -122.304782602556) 47.62015 -122.3048
## 12       E1 (47.6203486882073, -122.324419823241) 47.62035 -122.3244
## 13       E2  (47.6118432671102, -122.32016086571) 47.61184 -122.3202
## 14       E3  (47.603162336406, -122.319319689671) 47.60316 -122.3193
## 15       F1 (47.5484146593035, -122.354809670155) 47.54841 -122.3548
## 16       F2 (47.5254502461741, -122.365817548329) 47.52545 -122.3658
## 17       F3 (47.5261052985115, -122.336388313318) 47.52611 -122.3364
## 18       G1 (47.6091373306494, -122.307899616793) 47.60914 -122.3079
## 19       G2 (47.5958952989518, -122.306633195511) 47.59590 -122.3066
## 20       G3 (47.6031821881675, -122.292398835358) 47.60318 -122.2924
## 21       J1  (47.676809900774, -122.337899655521) 47.67681 -122.3379
## 22       J2 (47.6613374516723, -122.363818988307) 47.66134 -122.3638
## 23       J3 (47.6563781774877, -122.336468775341) 47.65638 -122.3365
## 24       K1 (47.6077552981764, -122.334107460638) 47.60776 -122.3341
## 25       K2 (47.5998930290529, -122.326813620856) 47.59989 -122.3268
## 26       K3 (47.5903972078525, -122.333545010682) 47.59040 -122.3336
## 27       L1 (47.7265488817709, -122.302631931191) 47.72655 -122.3026
## 28       L2 (47.7095588837442, -122.303661007867) 47.70956 -122.3037
## 29       L3 (47.6808531540255, -122.277032733938) 47.68085 -122.2770
## 30       M1 (47.6157584422587, -122.350867935301) 47.61576 -122.3509
## 31       M2 (47.6146150193586, -122.340275405136) 47.61462 -122.3403
## 32       M3 (47.6077571617787, -122.340896390036) 47.60776 -122.3409
## 33       N1 (47.7226875390406, -122.340459039106) 47.72269 -122.3405
## 34       N2  (47.698470493249, -122.351867710243) 47.69847 -122.3519
## 35       N3 (47.7045005246442, -122.329961214037) 47.70450 -122.3300
## 36       O1 (47.5822859359213, -122.311799603309) 47.58229 -122.3118
## 37       O2 (47.5656855826482, -122.330941962362) 47.56569 -122.3309
## 38       O3 (47.5345836385751, -122.303020266287) 47.53458 -122.3030
## 39       Q1  (47.650261230265, -122.400003042555) 47.65026 -122.4000
## 40       Q2 (47.6428529450151, -122.362673076853) 47.64285 -122.3627
## 41       Q3 (47.6269804063179, -122.362807276708) 47.62698 -122.3628
```

```
## 42       R1 (47.5758114569194, -122.288707022144) 47.57581 -122.2887
## 43       R2  (47.562285343514, -122.304240734006) 47.56229 -122.3042
## 44       R3 (47.5527951110333, -122.268210782218) 47.55280 -122.2682
## 45       S1 (47.5439339496481, -122.286476209963) 47.54393 -122.2865
## 46       S2 (47.5263519484816, -122.274095175041) 47.52635 -122.2741
## 47       S3 (47.5093533353672, -122.259542630385) 47.50935 -122.2595
## 48       SE (47.5476766838051, -122.284789228904) 47.54768 -122.2848
## 49       SW (47.5478566154038, -122.361787408364) 47.54786 -122.3618
## 50       U1 (47.6848677676269, -122.309913082907) 47.68487 -122.3099
## 51       U2  (47.6585545300635, -122.30659481859) 47.65855 -122.3066
## 52       U3 (47.6660083487855, -122.312204733721) 47.66601 -122.3122
## 53       W1 (47.5788164080083, -122.378814011668) 47.57882 -122.3788
## 54       W2 (47.5607068301888, -122.386946475037) 47.56071 -122.3869
## 55       W3 (47.5255479889804, -122.384581696918) 47.52555 -122.3846
##       census_tract
## 1  530330014004000
## 2  530330032021003
## 3  530330029003016
## 4  530330065001015
## 5  530330075022001
## 6  530330063002008
## 7  530330073032000
## 8  530330067023005
## 9  530330066001024
## 10 530330083001003
## 11 530330076002008
## 12 530330074061003
## 13 530330075031010
## 14 530330086002008
## 15 530330108001006
## 16 530330114012005
## 17 530330113001013
## 18 530330087001011
## 19 530330090002011
## 20 530330078001032
## 21 530330046001004
## 22 530330048004017
## 23 530330054021000
## 24 530330081021013
## 25 530330092001007
## 26 530330093002014
## 27 530330002022000
## 28 530330011001013
## 29 530330039002001
## 30 530330080041001
## 31 530330072023012
## 32 530330081011008
## 33 530330006021015
## 34 530330017012001
## 35 530330012013006
## 36 530330094003018
## 37 530330093003097
## 38 530330109001016
## 39 530330057002005
```

```
## 40 530330059023009
## 41 530330070011013
## 42 530330095003028
## 43 530330100011021
## 44 530330102004012
## 45 530330110012003
## 46 530330118013007
## 47 530330119011009
## 48 530330103013013
## 49 530330108002003
## 50 530330026001015
## 51 530330053032015
## 52 530330044021006
## 53 530330098012011
## 54 530330105021014
## 55 530330116011009
```

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)

The speed will be slower.

(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_ds <- mutate(beat_ct, state_code = substr(beat_ct$census_tract,1,2),
                   county_code = substr(beat_ct$census_tract,3,5))

beats_ds
```

```
##        Name                              Location Latitude Longitude
## 1        B1 (47.7097756394592, -122.370990523069) 47.70978 -122.3710
## 2        B2 (47.6790521901374, -122.391748391741) 47.67905 -122.3918
## 3        B3 (47.6812920482227, -122.364236159741) 47.68129 -122.3642
## 4        C1 (47.6342500180223, -122.315684762418) 47.63425 -122.3157
## 5        C2 (47.6192385752996, -122.313557430551) 47.61924 -122.3136
## 6        C3 (47.6300792887474, -122.292087128251) 47.63008 -122.2921
## 7  CITYWIDE (47.6210041048652, -122.332993498998) 47.62100 -122.3330
## 8        D1 (47.6274421308028, -122.345705781837) 47.62744 -122.3457
## 9        D2 (47.6256548876049, -122.331370005506) 47.62565 -122.3314
## 10       D3 (47.6103493249325, -122.328653706199) 47.61035 -122.3286
## 11        E (47.6201542748144, -122.304782602556) 47.62015 -122.3048
## 12       E1 (47.6203486882073, -122.324419823241) 47.62035 -122.3244
## 13       E2  (47.6118432671102, -122.32016086571) 47.61184 -122.3202
## 14       E3  (47.603162336406, -122.319319689671) 47.60316 -122.3193
## 15       F1 (47.5484146593035, -122.354809670155) 47.54841 -122.3548
## 16       F2 (47.5254502461741, -122.365817548329) 47.52545 -122.3658
```

```
## 17       F3 (47.5261052985115, -122.336388313318) 47.52611 -122.3364
## 18       G1 (47.6091373306494, -122.307899616793) 47.60914 -122.3079
## 19       G2 (47.5958952989518, -122.306633195511) 47.59590 -122.3066
## 20       G3 (47.6031821881675, -122.292398835358) 47.60318 -122.2924
## 21       J1  (47.676809900774, -122.337899655521) 47.67681 -122.3379
## 22       J2 (47.6613374516723, -122.363818988307) 47.66134 -122.3638
## 23       J3 (47.6563781774877, -122.336468775341) 47.65638 -122.3365
## 24       K1 (47.6077552981764, -122.334107460638) 47.60776 -122.3341
## 25       K2 (47.5998930290529, -122.326813620856) 47.59989 -122.3268
## 26       K3 (47.5903972078525, -122.333545010682) 47.59040 -122.3336
## 27       L1 (47.7265488817709, -122.302631931191) 47.72655 -122.3026
## 28       L2 (47.7095588837442, -122.303661007867) 47.70956 -122.3037
## 29       L3 (47.6808531540255, -122.277032733938) 47.68085 -122.2770
## 30       M1 (47.6157584422587, -122.350867935301) 47.61576 -122.3509
## 31       M2 (47.6146150193586, -122.340275405136) 47.61462 -122.3403
## 32       M3 (47.6077571617787, -122.340896390036) 47.60776 -122.3409
## 33       N1 (47.7226875390406, -122.340459039106) 47.72269 -122.3405
## 34       N2  (47.698470493249, -122.351867710243) 47.69847 -122.3519
## 35       N3 (47.7045005246442, -122.329961214037) 47.70450 -122.3300
## 36       O1 (47.5822859359213, -122.311799603309) 47.58229 -122.3118
## 37       O2 (47.5656855826482, -122.330941962362) 47.56569 -122.3309
## 38       O3 (47.5345836385751, -122.303020266287) 47.53458 -122.3030
## 39       Q1  (47.650261230265, -122.400003042555) 47.65026 -122.4000
## 40       Q2 (47.6428529450151, -122.362673076853) 47.64285 -122.3627
## 41       Q3 (47.6269804063179, -122.362807276708) 47.62698 -122.3628
## 42       R1 (47.5758114569194, -122.288707022144) 47.57581 -122.2887
## 43       R2  (47.562285343514, -122.304240734006) 47.56229 -122.3042
## 44       R3 (47.5527951110333, -122.268210782218) 47.55280 -122.2682
## 45       S1 (47.5439339496481, -122.286476209963) 47.54393 -122.2865
## 46       S2 (47.5263519484816, -122.274095175041) 47.52635 -122.2741
## 47       S3 (47.5093533353672, -122.259542630385) 47.50935 -122.2595
## 48       SE (47.5476766838051, -122.284789228904) 47.54768 -122.2848
## 49       SW (47.5478566154038, -122.361787408364) 47.54786 -122.3618
## 50       U1 (47.6848677676269, -122.309913082907) 47.68487 -122.3099
## 51       U2  (47.6585545300635, -122.30659481859) 47.65855 -122.3066
## 52       U3 (47.6660083487855, -122.312204733721) 47.66601 -122.3122
## 53       W1 (47.5788164080083, -122.378814011668) 47.57882 -122.3788
## 54       W2 (47.5607068301888, -122.386946475037) 47.56071 -122.3869
## 55       W3 (47.5255479889804, -122.384581696918) 47.52555 -122.3846
##       census_tract state_code county_code
## 1  530330014004000         53         033
## 2  530330032021003         53         033
## 3  530330029003016         53         033
## 4  530330065001015         53         033
## 5  530330075022001         53         033
## 6  530330063002008         53         033
## 7  530330073032000         53         033
## 8  530330067023005         53         033
## 9  530330066001024         53         033
## 10 530330083001003         53         033
## 11 530330076002008         53         033
## 12 530330074061003         53         033
## 13 530330075031010         53         033
## 14 530330086002008         53         033
```

```
## 15 530330108001006         53         033
## 16 530330114012005         53         033
## 17 530330113001013         53         033
## 18 530330087001011         53         033
## 19 530330090002011         53         033
## 20 530330078001032         53         033
## 21 530330046001004         53         033
## 22 530330048004017         53         033
## 23 530330054021000         53         033
## 24 530330081021013         53         033
## 25 530330092001007         53         033
## 26 530330093002014         53         033
## 27 530330002022000         53         033
## 28 530330011001013         53         033
## 29 530330039002001         53         033
## 30 530330080041001         53         033
## 31 530330072023012         53         033
## 32 530330081011008         53         033
## 33 530330006021015         53         033
## 34 530330017012001         53         033
## 35 530330012013006         53         033
## 36 530330094003018         53         033
## 37 530330093003097         53         033
## 38 530330109001016         53         033
## 39 530330057002005         53         033
## 40 530330059023009         53         033
## 41 530330070011013         53         033
## 42 530330095003028         53         033
## 43 530330100011021         53         033
## 44 530330102004012         53         033
## 45 530330110012003         53         033
## 46 530330118013007         53         033
## 47 530330119011009         53         033
## 48 530330103013013         53         033
## 49 530330108002003         53         033
## 50 530330026001015         53         033
## 51 530330053032015         53         033
## 52 530330044021006         53         033
## 53 530330098012011         53         033
## 54 530330105021014         53         033
## 55 530330116011009         53         033
```

Yes, It is what I expected as FIPS codes for WA is 53 and for King County 033.

(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_ds <- mutate(beats_ds, digital_code_11 = substr(beat_ct$census_tract,1,11))
beats_ds
```

```
##        Name                              Location Latitude Longitude
```

```
## 1        B1 (47.7097756394592, -122.370990523069) 47.70978 -122.3710
## 2        B2 (47.6790521901374, -122.391748391741) 47.67905 -122.3918
## 3        B3 (47.6812920482227, -122.364236159741) 47.68129 -122.3642
## 4        C1 (47.6342500180223, -122.315684762418) 47.63425 -122.3157
## 5        C2 (47.6192385752996, -122.313557430551) 47.61924 -122.3136
## 6        C3 (47.6300792887474, -122.292087128251) 47.63008 -122.2921
## 7  CITYWIDE (47.6210041048652, -122.332993498998) 47.62100 -122.3330
## 8        D1 (47.6274421308028, -122.345705781837) 47.62744 -122.3457
## 9        D2 (47.6256548876049, -122.331370005506) 47.62565 -122.3314
## 10       D3 (47.6103493249325, -122.328653706199) 47.61035 -122.3286
## 11        E (47.6201542748144, -122.304782602556) 47.62015 -122.3048
## 12       E1 (47.6203486882073, -122.324419823241) 47.62035 -122.3244
## 13       E2  (47.6118432671102, -122.32016086571) 47.61184 -122.3202
## 14       E3  (47.603162336406, -122.319319689671) 47.60316 -122.3193
## 15       F1 (47.5484146593035, -122.354809670155) 47.54841 -122.3548
## 16       F2 (47.5254502461741, -122.365817548329) 47.52545 -122.3658
## 17       F3 (47.5261052985115, -122.336388313318) 47.52611 -122.3364
## 18       G1 (47.6091373306494, -122.307899616793) 47.60914 -122.3079
## 19       G2 (47.5958952989518, -122.306633195511) 47.59590 -122.3066
## 20       G3 (47.6031821881675, -122.292398835358) 47.60318 -122.2924
## 21       J1  (47.676809900774, -122.337899655521) 47.67681 -122.3379
## 22       J2 (47.6613374516723, -122.363818988307) 47.66134 -122.3638
## 23       J3 (47.6563781774877, -122.336468775341) 47.65638 -122.3365
## 24       K1 (47.6077552981764, -122.334107460638) 47.60776 -122.3341
## 25       K2 (47.5998930290529, -122.326813620856) 47.59989 -122.3268
## 26       K3 (47.5903972078525, -122.333545010682) 47.59040 -122.3336
## 27       L1 (47.7265488817709, -122.302631931191) 47.72655 -122.3026
## 28       L2 (47.7095588837442, -122.303661007867) 47.70956 -122.3037
## 29       L3 (47.6808531540255, -122.277032733938) 47.68085 -122.2770
## 30       M1 (47.6157584422587, -122.350867935301) 47.61576 -122.3509
## 31       M2 (47.6146150193586, -122.340275405136) 47.61462 -122.3403
## 32       M3 (47.6077571617787, -122.340896390036) 47.60776 -122.3409
## 33       N1 (47.7226875390406, -122.340459039106) 47.72269 -122.3405
## 34       N2  (47.698470493249, -122.351867710243) 47.69847 -122.3519
## 35       N3 (47.7045005246442, -122.329961214037) 47.70450 -122.3300
## 36       O1 (47.5822859359213, -122.311799603309) 47.58229 -122.3118
## 37       O2 (47.5656855826482, -122.330941962362) 47.56569 -122.3309
## 38       O3 (47.5345836385751, -122.303020266287) 47.53458 -122.3030
## 39       Q1  (47.650261230265, -122.400003042555) 47.65026 -122.4000
## 40       Q2 (47.6428529450151, -122.362673076853) 47.64285 -122.3627
## 41       Q3 (47.6269804063179, -122.362807276708) 47.62698 -122.3628
## 42       R1 (47.5758114569194, -122.288707022144) 47.57581 -122.2887
## 43       R2  (47.562285343514, -122.304240734006) 47.56229 -122.3042
## 44       R3 (47.5527951110333, -122.268210782218) 47.55280 -122.2682
## 45       S1 (47.5439339496481, -122.286476209963) 47.54393 -122.2865
## 46       S2 (47.5263519484816, -122.274095175041) 47.52635 -122.2741
## 47       S3 (47.5093533353672, -122.259542630385) 47.50935 -122.2595
## 48       SE (47.5476766838051, -122.284789228904) 47.54768 -122.2848
## 49       SW (47.5478566154038, -122.361787408364) 47.54786 -122.3618
## 50       U1 (47.6848677676269, -122.309913082907) 47.68487 -122.3099
## 51       U2  (47.6585545300635, -122.30659481859) 47.65855 -122.3066
## 52       U3 (47.6660083487855, -122.312204733721) 47.66601 -122.3122
## 53       W1 (47.5788164080083, -122.378814011668) 47.57882 -122.3788
## 54       W2 (47.5607068301888, -122.386946475037) 47.56071 -122.3869
```

```
## 55       W3 (47.5255479889804, -122.384581696918) 47.52555 -122.3846
##       census_tract state_code county_code digital_code_11
## 1  530330014004000         53         033     53033001400
## 2  530330032021003         53         033     53033003202
## 3  530330029003016         53         033     53033002900
## 4  530330065001015         53         033     53033006500
## 5  530330075022001         53         033     53033007502
## 6  530330063002008         53         033     53033006300
## 7  530330073032000         53         033     53033007303
## 8  530330067023005         53         033     53033006702
## 9  530330066001024         53         033     53033006600
## 10 530330083001003         53         033     53033008300
## 11 530330076002008         53         033     53033007600
## 12 530330074061003         53         033     53033007406
## 13 530330075031010         53         033     53033007503
## 14 530330086002008         53         033     53033008600
## 15 530330108001006         53         033     53033010800
## 16 530330114012005         53         033     53033011401
## 17 530330113001013         53         033     53033011300
## 18 530330087001011         53         033     53033008700
## 19 530330090002011         53         033     53033009000
## 20 530330078001032         53         033     53033007800
## 21 530330046001004         53         033     53033004600
## 22 530330048004017         53         033     53033004800
## 23 530330054021000         53         033     53033005402
## 24 530330081021013         53         033     53033008102
## 25 530330092001007         53         033     53033009200
## 26 530330093002014         53         033     53033009300
## 27 530330002022000         53         033     53033000202
## 28 530330011001013         53         033     53033001100
## 29 530330039002001         53         033     53033003900
## 30 530330080041001         53         033     53033008004
## 31 530330072023012         53         033     53033007202
## 32 530330081011008         53         033     53033008101
## 33 530330006021015         53         033     53033000602
## 34 530330017012001         53         033     53033001701
## 35 530330012013006         53         033     53033001201
## 36 530330094003018         53         033     53033009400
## 37 530330093003097         53         033     53033009300
## 38 530330109001016         53         033     53033010900
## 39 530330057002005         53         033     53033005700
## 40 530330059023009         53         033     53033005902
## 41 530330070011013         53         033     53033007001
## 42 530330095003028         53         033     53033009500
## 43 530330100011021         53         033     53033010001
## 44 530330102004012         53         033     53033010200
## 45 530330110012003         53         033     53033011001
## 46 530330118013007         53         033     53033011801
## 47 530330119011009         53         033     53033011901
## 48 530330103013013         53         033     53033010301
## 49 530330108002003         53         033     53033010800
## 50 530330026001015         53         033     53033002600
## 51 530330053032015         53         033     53033005303
## 52 530330044021006         53         033     53033004402
```

```
## 53 530330098012011          53          033      53033009801
## 54 530330105021014          53          033      53033010502
## 55 530330116011009          53          033      53033011601
```

(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.

```
inspect_raw_edu <- read.csv("census_edu_data.csv")

edu11_digital_code <- mutate(inspect_raw_edu,
digital_code_11 = substr(inspect_raw_edu$GEO.id,10,21))
head(edu11_digital_code)  # using head so PDF does not print excess pages of dataset
```

```
##                 GEO.id    GEO.id2                          GEO.display.label
## 1 1400000US53033000100 5.3033e+10    Census Tract 1, King County, Washington
## 2 1400000US53033000200 5.3033e+10    Census Tract 2, King County, Washington
## 3 1400000US53033000300 5.3033e+10    Census Tract 3, King County, Washington
## 4 1400000US53033000401 5.3033e+10 Census Tract 4.01, King County, Washington
## 5 1400000US53033000402 5.3033e+10 Census Tract 4.02, King County, Washington
## 6 1400000US53033000500 5.3033e+10    Census Tract 5, King County, Washington
##   total no_schooling nursery_school kindergarten X1st_grade X2nd_grade
## 1  5708           82              0            0          0          0
## 2  6079          115              0            0          0          0
## 3  2152           49              0            0          0          0
## 4  5084           60              0            0          0          0
## 5  4498           60              0            0          0          0
## 6  2333            6              9            0          0          0
##   X3rd_grade X4th_grade X5th_grade X6th_grade X7th_grade X8th_grade X9th_grade
## 1         59         59          0         44          0        110          0
## 2          0          0          0         66          3          0         41
## 3          1          0          0          0          0          0          7
## 4          0          0         30         43          0         28         20
## 5          0          0          0          0          0         32          0
## 6          9          2          0          0          0          0          0
##   X10th_grade X11th_grade X12th_grade_no_diploma high_school_diploma
## 1          28          27                   112                833
## 2          17          42                   125                614
## 3          59          62                    35                346
## 4          60          54                    18                769
## 5          19          41                    76                730
## 6           0           0                    18                154
##   ged_or_alternative_credential some_college_less_than_1_year
## 1                           239                          259
## 2                           169                          229
## 3                            61                          198
## 4                           248                          472
## 5                            40                          307
```

```
## 6                            26                           79
##   some_college_1_or_more_years_no_degree associates_degree bachelors_degree
## 1                                   669               470             1600
## 2                                   739               458             2105
## 3                                   172               357              571
## 4                                   864               432             1315
## 5                                   628               323             1345
## 6                                   246               125              880
##   masters_degree professional_school_degree doctorate_degree digital_code_11
## 1            584                        319              214     53033000100
## 2           1045                         77              234     53033000200
## 3            170                         31               33     53033000300
## 4            526                         80               65     53033000401
## 5            602                        185              110     53033000402
## 6            480                        187              112     53033000500
```

(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?

```
beat_census <- left_join(beats_ds, edu11_digital_code, by = "digital_code_11")
beat_census %>%
 filter(total > 0) %>%
    summarise(count = n())
```

```
##   count
## 1    31
```

```
head(beat_census) # using head so PDF does not print excess pages of dataset
```

```
##   Name                              Location Latitude Longitude    census_tract
## 1   B1 (47.7097756394592, -122.370990523069) 47.70978 -122.3710 530330014004000
## 2   B2 (47.6790521901374, -122.391748391741) 47.67905 -122.3918 530330032021003
## 3   B3 (47.6812920482227, -122.364236159741) 47.68129 -122.3642 530330029003016
## 4   C1 (47.6342500180223, -122.315684762418) 47.63425 -122.3157 530330065001015
## 5   C2 (47.6192385752996, -122.313557430551) 47.61924 -122.3136 530330075022001
## 6   C3 (47.6300792887474, -122.292087128251) 47.63008 -122.2921 530330063002008
##   state_code county_code digital_code_11               GEO.id     GEO.id2
## 1         53         033     53033001400 1400000US53033001400 53033001400
## 2         53         033     53033003202                 <NA>          NA
## 3         53         033     53033002900 1400000US53033002900 53033002900
## 4         53         033     53033006500 1400000US53033006500 53033006500
## 5         53         033     53033007502                 <NA>          NA
## 6         53         033     53033006300 1400000US53033006300 53033006300
##                          GEO.display.label total no_schooling nursery_school
## 1 Census Tract 14, King County, Washington  4155            0              0
## 2                                    <NA>    NA           NA             NA
## 3 Census Tract 29, King County, Washington  3524            0              0
## 4 Census Tract 65, King County, Washington  3842            1              0
## 5                                    <NA>    NA           NA             NA
## 6 Census Tract 63, King County, Washington  4266            5              0
##   kindergarten X1st_grade X2nd_grade X3rd_grade X4th_grade X5th_grade
## 1            0          0          0         15          0          0
```

```
## 2           NA         NA         NA         NA         NA         NA
## 3            0          0          0          0          0          0
## 4            0          0          0          0          0          0
## 5           NA         NA         NA         NA         NA         NA
## 6            0          0          0          0          0          0
##   X6th_grade X7th_grade X8th_grade X9th_grade X10th_grade X11th_grade
## 1          0          0         33         18         110          20
## 2         NA         NA         NA         NA          NA          NA
## 3          0          0          0         17           0          32
## 4          0          1          0          2           0          20
## 5         NA         NA         NA         NA          NA          NA
## 6          0          0          5         20          18           0
##   X12th_grade_no_diploma high_school_diploma ged_or_alternative_credential
## 1                     34                472                           100
## 2                     NA                 NA                            NA
## 3                     16                196                            88
## 4                      0                136                            15
## 5                     NA                 NA                            NA
## 6                      0                 54                            15
##   some_college_less_than_1_year some_college_1_or_more_years_no_degree
## 1                           245                                   536
## 2                            NA                                    NA
## 3                           134                                   447
## 4                           145                                   416
## 5                            NA                                    NA
## 6                           161                                   401
##   associates_degree bachelors_degree masters_degree professional_school_degree
## 1               310             1301            760                        64
## 2                NA               NA             NA                        NA
## 3               111             1310            864                       212
## 4               204             1620            692                       365
## 5                NA               NA             NA                        NA
## 6               212             1822            915                       432
##   doctorate_degree
## 1              137
## 2               NA
## 3               97
## 4              225
## 5               NA
## 6              206
```

There are 25 police beats that do not have an associated census data in the data frame.

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?

```
crime_beat_cen <- mutate(beat_census, Beat = Name)
crime_beat_cen <- left_join(ds_cleaned, crime_beat_cen, by = "Beat")
crime_beat_cen <- select(crime_beat_cen, -Name)

head(crime_beat_cen) # using head so PDF does not print excess pages of dataset
```

```
## # A tibble: 6 x 47
## # Groups:   year [1]
```

```
##   Report.Number Occurred.Date Occurred.Time Reported.Date Reported.Time
##           <dbl> <chr>                <int> <chr>                 <int>
## 1       2.01e13 03/17/2008            1000 03/17/2008             2245
## 2       2.01e12 01/08/2008             800 01/08/2008             1925
## 3       2.01e13 03/17/2008            2322 03/17/2008             2327
## 4       2.01e13 03/17/2008            2030 03/17/2008             2338
## 5       2.01e13 03/17/2008            2339 03/17/2008             2339
## 6       2.01e13 03/18/2008              41 03/18/2008               41
## # ... with 42 more variables: Crime.Subcategory <chr>,
## #   Primary.Offense.Description <chr>, Precinct <chr>, Sector <chr>,
## #   Beat <chr>, Neighborhood <chr>, year <chr>, Location <chr>, Latitude <dbl>,
## #   Longitude <dbl>, census_tract <chr>, state_code <chr>, county_code <chr>,
## #   digital_code_11 <chr>, GEO.id <chr>, GEO.id2 <dbl>,
## #   GEO.display.label <chr>, total <int>, no_schooling <int>,
## #   nursery_school <int>, kindergarten <int>, X1st_grade <int>, ...
```

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

The final dimensions of the joined data set is 519,305 $\times$ 47.