IMT 573: Problem Set 4

Working with Data: Part II

Alex Johnson

Due: October 30, 2022

Collaborators:

username: biggestkoalas at

‘$/urlhttps://stackoverflow.com/questions/65795510/r-call-geolocator-latlon-function-returns-na$‘

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

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

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

(b) Looking at Years That Crimes Were Committed

1. Let’s start by looking at the years in which crimes were committed.
2. What is the earliest year in the dataset?
3. Are there any distinct trends with the annual number of crimes committed in the dataset?

The earliest year in the data set is 1908.

From 1908 to 1999 annual crime totals were low or increased by a small amount, then in 2000s the amount significantly increases seeing its most significant increase in 2008 where it increases by over 40 thousand.

(c) Looking at Frequency of Beats

1. What is a Police Beat? How frequently are the beats in the Crime Dataset listed?
2. Are there any anomolies with how frequently some of the beats are listed?
3. Are there missing beats?

(1c) Solutions

1. According to data.seattle.gov, '*Beat*' is defined as a designated police sector boundary where offense(s) occurred. Beat is a further breakdown of '*Sector*' [23 total] and '*Precinct*' [6 total]
2. beat districts CS, CTY, DET, H1, INV, K, LAPT, N, S, SS, W, WS, and X9 report an abnormally small count relative to the rest of the Beats in set
3. Comparing the 64 unique Beats identified on 'ds_beat' with the full list of Beat's by geolocation as available at data.seattle.gov, for which there are 55 unique identified Beat geolocations, there are 10 unlisted Beats in 'ds_beat' that are not official Beats–examples are S and SS (city lists S1, S2, and S3 as Beats), W and WS (city lists W1:W3 only), CS, CTY, DET, INV, LAPT, and X9 (which is '99' on the official data set)

(d) Importing Police Beat Data and Filtering on Frequency *Initial task*: 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."

(1d) Questions

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

(1d) Solutions

1. The '$inspect_raw_beat_ds$' lists 57 unique Beats whereas the original 'ds_beat' lists 64.
2. The 13 Beats listed in the original that are not in the new set do not affect numeric dispersions across categories as the unique Beats from old set had counts of less than 10 crimes.
3. As stated above, these unique observations are rather infrequent, but the most impact came from one unnammed Beat category in 'ds_beat', which had 3213 crimes, but this concern should be overlooked considering the unnamed Beat values relatively small share of observations compared to the set.
4. The frequency of each beat is so low that I don't believe removal would change the scope of the crime dataset.

(1d) Tasks

1. Remove all instances in the Crime Dataset that have beats which occur fewer than 10 times across the Crime Dataset: See '$beat_ten$'
2. Also remove any observations with missing beats.
3. After only keeping years of interest and filtering based on frequency of beat, how many observations do we have in the Crime Dataset?

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

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_2 <- filter(ds_beat, Name != "CTY", Name != "DET", Name != "INV",
Name != "K", Name != "N", Name != "S", Name != "SS", Name != "W", Name != "WS", Name != "")

ds_cleaned
```

```
## # A tibble: 519,305 x 12
## # Groups:   year [12]
##    Report~1 Occur~2 Occur~3 Repor~4 Repor~5 Crime~6 Prima~7 Preci~8 Sector Beat
##       <dbl> <chr>     <int> <chr>     <int> <chr>   <chr>   <chr>   <chr>  <chr>
##  1  2.01e13 03/17/~    1000 03/17/~    2245 MOTOR ~ VEH-TH~ SOUTHW~ W      W1
##  2  2.01e12 01/08/~     800 01/08/~    1925 BURGLA~ BURGLA~ EAST    C      C1
##  3  2.01e13 03/17/~    2322 03/17/~    2327 CAR PR~ THEFT-~ EAST    C      C1
##  4  2.01e13 03/17/~    2030 03/17/~    2338 ROBBER~ ROBBER~ EAST    G      G3
##  5  2.01e13 03/17/~    2339 03/17/~    2339 AGGRAV~ ASSLT-~ SOUTH   S      S2
##  6  2.01e13 03/18/~      41 03/18/~      41 TRESPA~ TRESPA~ WEST    M      M1
##  7  2.01e12 01/08/~    1915 01/08/~    1930 THEFT-~ THEFT-~ NORTH   N      N2
##  8  2.01e12 01/08/~    1800 01/08/~    1925 THEFT-~ THEFT-~ EAST    G      G3
##  9  2.01e13 03/18/~     200 03/18/~     204 BURGLA~ BURGLA~ EAST    G      G3
## 10  2.01e13 03/18/~     205 03/18/~     205 DUI     DUI-LI~ WEST    Q      Q1
## # ... with 519,295 more rows, 2 more variables: Neighborhood <chr>, year <chr>,
## #   and abbreviated variable names 1: Report.Number, 2: Occurred.Date,
## #   3: Occurred.Time, 4: Reported.Date, 5: Reported.Time, 6: Crime.Subcategory,
## #   7: Primary.Offense.Description, 8: Precinct
```

```
ds_cleaned_2
```

```
## # A tibble: 55 x 2
##    Name  count
##    <chr> <int>
##  1 B1    11131
##  2 B2    13759
##  3 B3    13034
##  4 C1     8271
##  5 C2     6866
##  6 C3     7424
##  7 CS        1
```

```
##  8 D1    13202
##  9 D2    12046
## 10 D3    10131
## # ... with 45 more rows
```

After 2008 there were 519,305 observations.

(e) Importing and Inspecting Police Beat Data (1e) Instructions

1. To join the Beat Dataset to census data, use census tract information.
2. Use the '*censusr*' package to extract the 15-digit census tract for each police beat using the corresponding latitude and longitude.
3. Do this using each of the police beats listed in the Beats Dataset.
4. Do not use a for-loop for this but rely on R functions (e.g. 'apply' functions).
5. Add a column to the Beat Dataset that contains the 15-digit census tract for each beat. (HINT: '*censusr*''s '$call_geolocator_latlon$' function useful)

```
# 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, benchmark, vintage)
{
  if (missing(benchmark)) {
    benchmark <- "Public_AR_Census2020"
  }
  else {
    benchmark <- benchmark
  }
  if (missing(vintage)) {
    vintage <- "Census2020_Census2020"
  }
  else {
    vintage <- vintage
  }
  call_start <- "https://geocoding.geo.census.gov/geocoder/geographies/coordinates?"
  url <- paste0("x=", lon, "&y=", lat)
  benchmark0 <- paste0("&benchmark=", benchmark)
  vintage0 <- paste0("&vintage=", vintage, "&format=json")
  url_full <- paste0(call_start, url, benchmark0, vintage0)
  r <- httr::GET(url_full)
  httr::stop_for_status(r)
  response <- httr::content(r)
  return(response$result$geographies$`Census Blocks`[[1]]$GEOID)
  if (length(response$result$geographies$`2020 Census Blocks`[[1]]$GEOID) ==
      0) {
```

```
    message(paste0("Lat/lon (", lat, ", ", lon, ") returned no geocodes. An NA was returned."))
    return(NA_character_)
  }
  else {
    if (length(response$result$geographies$`2020 Census Blocks`[[1]]$GEOID) >
        1) {
      message(paste0("Lat/lon (", lat, ", ", lon, ") returned more than geocode. The first match was re
    }
    return(response$result$geographies$`2020 Census Blocks`[[1]]$GEOID)
  }
}
#this function was borrowed from url:
#https://stackoverflow.com/questions/65795510/r-call-geolocator-latlon-function-returns-na

myfun <- func

beat_censustract <- mutate(rbeat, census_tract = mapply(myfun,
Latitude,Longitude))

beat_censustract
```

```
##    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    D1 (47.6274421308028, -122.345705781837) 47.62744 -122.3457
## 8    D2 (47.6256548876049, -122.331370005506) 47.62565 -122.3314
## 9    D3 (47.6103493249325, -122.328653706199) 47.61035 -122.3286
## 10    E (47.6201542748144, -122.304782602556) 47.62015 -122.3048
## 11   E1 (47.6203486882073, -122.324419823241) 47.62035 -122.3244
## 12   E2  (47.6118432671102, -122.32016086571) 47.61184 -122.3202
## 13   E3  (47.603162336406, -122.319319689671) 47.60316 -122.3193
## 14   F1 (47.5484146593035, -122.354809670155) 47.54841 -122.3548
## 15   F2 (47.5254502461741, -122.365817548329) 47.52545 -122.3658
## 16   F3 (47.5261052985115, -122.336388313318) 47.52611 -122.3364
## 17   G1 (47.6091373306494, -122.307899616793) 47.60914 -122.3079
## 18   G2 (47.5958952989518, -122.306633195511) 47.59590 -122.3066
## 19   G3 (47.6031821881675, -122.292398835358) 47.60318 -122.2924
## 20   J1  (47.676809900774, -122.337899655521) 47.67681 -122.3379
## 21   J2 (47.6613374516723, -122.363818988307) 47.66134 -122.3638
## 22   J3 (47.6563781774877, -122.336468775341) 47.65638 -122.3365
## 23   K1 (47.6077552981764, -122.334107460638) 47.60776 -122.3341
## 24   K2 (47.5998930290529, -122.326813620856) 47.59989 -122.3268
## 25   K3 (47.5903972078525, -122.333545010682) 47.59040 -122.3336
## 26   L1 (47.7265488817709, -122.302631931191) 47.72655 -122.3026
## 27   L2 (47.7095588837442, -122.303661007867) 47.70956 -122.3037
## 28   L3 (47.6808531540255, -122.277032733938) 47.68085 -122.2770
## 29   M1 (47.6157584422587, -122.350867935301) 47.61576 -122.3509
## 30   M2 (47.6146150193586, -122.340275405136) 47.61462 -122.3403
## 31   M3 (47.6077571617787, -122.340896390036) 47.60776 -122.3409
```

```
## 32   N1 (47.7226875390406, -122.340459039106) 47.72269 -122.3405
## 33   N2  (47.698470493249, -122.351867710243) 47.69847 -122.3519
## 34   N3 (47.7045005246442, -122.329961214037) 47.70450 -122.3300
## 35   O1 (47.5822859359213, -122.311799603309) 47.58229 -122.3118
## 36   O2 (47.5656855826482, -122.330941962362) 47.56569 -122.3309
## 37   O3 (47.5345836385751, -122.303020266287) 47.53458 -122.3030
## 38   Q1  (47.650261230265, -122.400003042555) 47.65026 -122.4000
## 39   Q2 (47.6428529450151, -122.362673076853) 47.64285 -122.3627
## 40   Q3 (47.6269804063179, -122.362807276708) 47.62698 -122.3628
## 41   R1 (47.5758114569194, -122.288707022144) 47.57581 -122.2887
## 42   R2  (47.562285343514, -122.304240734006) 47.56229 -122.3042
## 43   R3 (47.5527951110333, -122.268210782218) 47.55280 -122.2682
## 44   S1 (47.5439339496481, -122.286476209963) 47.54393 -122.2865
## 45   S2 (47.5263519484816, -122.274095175041) 47.52635 -122.2741
## 46   S3 (47.5093533353672, -122.259542630385) 47.50935 -122.2595
## 47   SE (47.5476766838051, -122.284789228904) 47.54768 -122.2848
## 48   SW (47.5478566154038, -122.361787408364) 47.54786 -122.3618
## 49   U1 (47.6848677676269, -122.309913082907) 47.68487 -122.3099
## 50   U2  (47.6585545300635, -122.30659481859) 47.65855 -122.3066
## 51   U3 (47.6660083487855, -122.312204733721) 47.66601 -122.3122
## 52   W1 (47.5788164080083, -122.378814011668) 47.57882 -122.3788
## 53   W2 (47.5607068301888, -122.386946475037) 47.56071 -122.3869
## 54   W3 (47.5255479889804, -122.384581696918) 47.52555 -122.3846
##       census_tract
## 1  530330014004000
## 2  530330032021003
## 3  530330029003016
## 4  530330065001015
## 5  530330075022001
## 6  530330063002008
## 7  530330067023005
## 8  530330066001024
## 9  530330083001003
## 10 530330076002008
## 11 530330074061003
## 12 530330075031010
## 13 530330086002008
## 14 530330108001006
## 15 530330114012005
## 16 530330113001013
## 17 530330087001011
## 18 530330090002011
## 19 530330078001032
## 20 530330046001004
## 21 530330048004017
## 22 530330054021000
## 23 530330081021013
## 24 530330092001007
## 25 530330093002014
## 26 530330002022000
## 27 530330011001013
## 28 530330039002001
## 29 530330080041001
## 30 530330072023012
```

```
## 31 530330081011008
## 32 530330006021015
## 33 530330017012001
## 34 530330012013006
## 35 530330094003018
## 36 530330093003097
## 37 530330109001016
## 38 530330057002005
## 39 530330059023009
## 40 530330070011013
## 41 530330095003028
## 42 530330100011021
## 43 530330102004012
## 44 530330110012003
## 45 530330118013007
## 46 530330119011009
## 47 530330103013013
## 48 530330108002003
## 49 530330026001015
## 50 530330053032015
## 51 530330044021006
## 52 530330098012011
## 53 530330105021014
## 54 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 could have been much 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

(1f) Instructions and Question

1. First, create a column that contains the state code for each beat in the Beats Dataset.
2. Next, 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.
3. Are the extracted state and county codes what you would expect them to be? Why or why not?

```
beats <- mutate(beat_censustract, state_code =
substr(beat_censustract$census_tract,1,2),county_code =
substr(beat_censustract$census_tract,3,5))
```

(1f) Solutions

```
beats
```

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

```
## 52   W1 (47.5788164080083, -122.378814011668) 47.57882 -122.3788
## 53   W2 (47.5607068301888, -122.386946475037) 47.56071 -122.3869
## 54   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  530330067023005         53         033
## 8  530330066001024         53         033
## 9  530330083001003         53         033
## 10 530330076002008         53         033
## 11 530330074061003         53         033
## 12 530330075031010         53         033
## 13 530330086002008         53         033
## 14 530330108001006         53         033
## 15 530330114012005         53         033
## 16 530330113001013         53         033
## 17 530330087001011         53         033
## 18 530330090002011         53         033
## 19 530330078001032         53         033
## 20 530330046001004         53         033
## 21 530330048004017         53         033
## 22 530330054021000         53         033
## 23 530330081021013         53         033
## 24 530330092001007         53         033
## 25 530330093002014         53         033
## 26 530330002022000         53         033
## 27 530330011001013         53         033
## 28 530330039002001         53         033
## 29 530330080041001         53         033
## 30 530330072023012         53         033
## 31 530330081011008         53         033
## 32 530330006021015         53         033
## 33 530330017012001         53         033
## 34 530330012013006         53         033
## 35 530330094003018         53         033
## 36 530330093003097         53         033
## 37 530330109001016         53         033
## 38 530330057002005         53         033
## 39 530330059023009         53         033
## 40 530330070011013         53         033
## 41 530330095003028         53         033
## 42 530330100011021         53         033
## 43 530330102004012         53         033
## 44 530330110012003         53         033
## 45 530330118013007         53         033
## 46 530330119011009         53         033
## 47 530330103013013         53         033
## 48 530330108002003         53         033
## 49 530330026001015         53         033
## 50 530330053032015         53         033
```

```
## 51 530330044021006          53          033
## 52 530330098012011          53          033
## 53 530330105021014          53          033
## 54 530330116011009          53          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.

(1g) Instructions

1. To join the census data to the Beats Dataset, we must have this code for each of the beats.
2. Extract the 11-digit code for each of the beats in the Beats Dataset.
3. The 11 digits consist of the 2 state digits, 3 county digits, and 6 tract digits.
4. Add a column with the 11-digit code for each beat.

(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.

(i) Join Datasets (1i) Instructions and Question

1. Join the census data with the Beat Dataset using the 11-digit codes as keys.
2. 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).
3. Are there any police beats that do not have any associated census data?

```
beat_edu <- left_join(beats, edu_11_digital_code, by = "digital_code_11" )
beat_edu_2 <- right_join(beats, edu_11_digital_code, by = "digital_code_11" )
```

(1i) Tasks

1. Then, join the Crime Dataset to our joined beat/census data.
2. Again, be sure you do not lose any observations from the Crime Dataset.
3. What are the final dimensions of the joined dataset?

(1i) Solutions

There are 519,305 observations across 47 variables–these are the dimensions of the saved data set

```
write.csv(crime_beat_census, "crime_beat_census.csv", row.names = FALSE)

crime_tibble <- as_tibble(read.csv("crime_beat_census.csv", header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = ""))

crime_tibble
```

```
## # A tibble: 519,305 x 47
##    Report~1 Occur~2 Occur~3 Repor~4 Repor~5 Crime~6 Prima~7 Preci~8 Sector Beat
##       <dbl> <chr>     <int> <chr>     <int> <chr>   <chr>   <chr>   <chr>  <chr>
##  1  2.01e13 03/17/~    1000 03/17/~    2245 MOTOR ~ VEH-TH~ SOUTHW~ W      W1
##  2  2.01e12 01/08/~     800 01/08/~    1925 BURGLA~ BURGLA~ EAST    C      C1
##  3  2.01e13 03/17/~    2322 03/17/~    2327 CAR PR~ THEFT-~ EAST    C      C1
##  4  2.01e13 03/17/~    2030 03/17/~    2338 ROBBER~ ROBBER~ EAST    G      G3
##  5  2.01e13 03/17/~    2339 03/17/~    2339 AGGRAV~ ASSLT-~ SOUTH   S      S2
##  6  2.01e13 03/18/~      41 03/18/~      41 TRESPA~ TRESPA~ WEST    M      M1
##  7  2.01e12 01/08/~    1915 01/08/~    1930 THEFT-~ THEFT-~ NORTH   N      N2
##  8  2.01e12 01/08/~    1800 01/08/~    1925 THEFT-~ THEFT-~ EAST    G      G3
##  9  2.01e13 03/18/~     200 03/18/~     204 BURGLA~ BURGLA~ EAST    G      G3
## 10  2.01e13 03/18/~     205 03/18/~     205 DUI     DUI-LI~ WEST    Q      Q1
## # ... with 519,295 more rows, 37 more variables: Neighborhood <chr>,
## #   year <int>, Location <chr>, Latitude <dbl>, Longitude <dbl>,
## #   census_tract <dbl>, state_code <int>, county_code <int>,
## #   digital_code_11 <dbl>, GEO.id <chr>, GEO.id2 <dbl>,
## #   GEO.display.label <chr>, 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>, ...
```