Chicago_Theft_Data

** Data Reading

```
library(tidyverse)
library(jsonlite)
library(lubridate)
data<- read_json("https://data.cityofchicago.org/resource/ijzp-q8t2.json?primary_type=THEFT&$limit=3000
```

- A for loop that iterates through years, starting in 2016 and ending in 2019. Using the loop to alter the parameters of the API URL for each year, then running read_json() such that each iteration of the loop loads a different year's worth of theft data. Using bind_rows() to append each year of data to a preallocated dataframe called thefts.

```
thefts<- tibble()
for(i in c(2016:2019)){
  url<- paste0("https://data.cityofchicago.org/resource/ijzp-q8t2.json?primary_type=THEFT&$limit=40000&
  theft_data<-read_json(url,simplifyVector = TRUE)
  theft_data[22]<- NULL
  thefts <- bind_rows(thefts,theft_data)
}
```

- Using the thefts dataframe, using the lubridate functions we extract the year, month, day, week, and hour columns. Additionally,we drop any rows that have an NA value for the latitude or longitude columns.

```
thefts1<- thefts%>%
  mutate(dates=ymd_hms(date))%>%
  mutate(year=year(dates))%>%
  mutate(month=month(dates))%>%
  mutate(day=day(dates))%>%
  mutate(hour=hour(dates))
thefts1 <- thefts1 %>%
  drop_na(longitude,latitude)
```

- Creating a new column called category in the thefts dataset. Using ifelse() or case_when(), we setcategory equal to "petty" when the description column equals less than $500, pocket-picking, or pursesnatching. Setting category equal to "grand" for all other values of description.

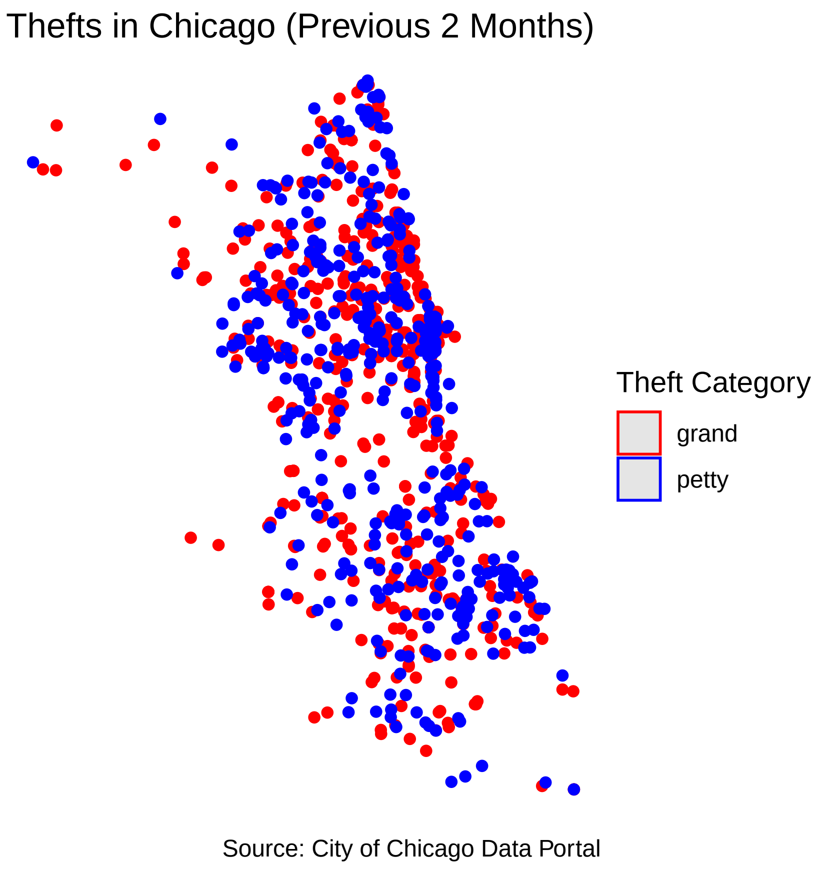
```
thefts1 <- thefts1%>%
  mutate(category=case_when(
    description == "$500 AND UNDER" ~ "petty",
    description == "POCKET-PICKING" ~ "petty",
    description == "PURSE-SNATCHING" ~ "petty",
    TRUE ~ "grand"
    ))
```

- Now that we've loaded and cleaned our dataset, let's take a look at the geographic distribution of thefts. We need to convert the latitude and longitude columns in the data into spatial geometries (points) before plotting. Using the st_as_sf() function to convert the respective columns into an sf geometry column. We specify the CRS as 4326 when converting, and use the remove = FALSE argument to keep the original latitude and longitude columns in the data. We will have a new column in our dataset named geometry if successful.

```r
library(sf)
thefts1 <- st_as_sf(thefts1,
                   coords = c("longitude", "latitude"),
                   crs = 4326,
                   remove = FALSE)
```

- Next, we create a filtered version of the thefts dataset that contains only the past two months of data. Calling this dataset thefts_fil. To perform this filtering, we can use the now() function from lubridate to get the current month. Don't just type in the month manually, as then your report won't be self-updating. Using thefts_fil, we replicate the plot below using ggplot() and geom_sf().

```r
thefts_fil <- thefts1%>%
  filter(month == month(now())-2| month == month(now())-1)
thefts_fil_sample <- thefts_fil%>%
  sample_frac(0.1)
library(ggplot2)
ggplot() +
  geom_sf(
    data = thefts_fil_sample ,
    aes(color= category)) +
  theme_void() +
  labs(
    title = "Thefts in Chicago (Previous 2 Months)",
    caption = "Source: City of Chicago Data Portal"
  ) +
  scale_colour_manual (
    values = c("petty" = "blue", "grand" = "red"),
    name = "Theft Category"
  )
```


Thefts in Chicago (Previous 2 Months)
Theft Category
grand
petty
Source: City of Chicago Data Portal

- This plot is still pretty hard to read, and it's difficult to discern what conclusions we should draw from it. To make the map clearer, we can aggregate the individual-level data to the Census tract level and examine data from a longer time period.

#——————————————————————————————

- We start by downloading Census tracts for Cook County, IL using tidycensus. We want to get the geometries for each tract, so be sure to set geometry = TRUE when using get_acs(). We also want to retrieve the total population for each tract. NOTE: You should use the 2016 5-year ACS for all your boundaries and variables. Save the Census tract data to a dataframe named cook.

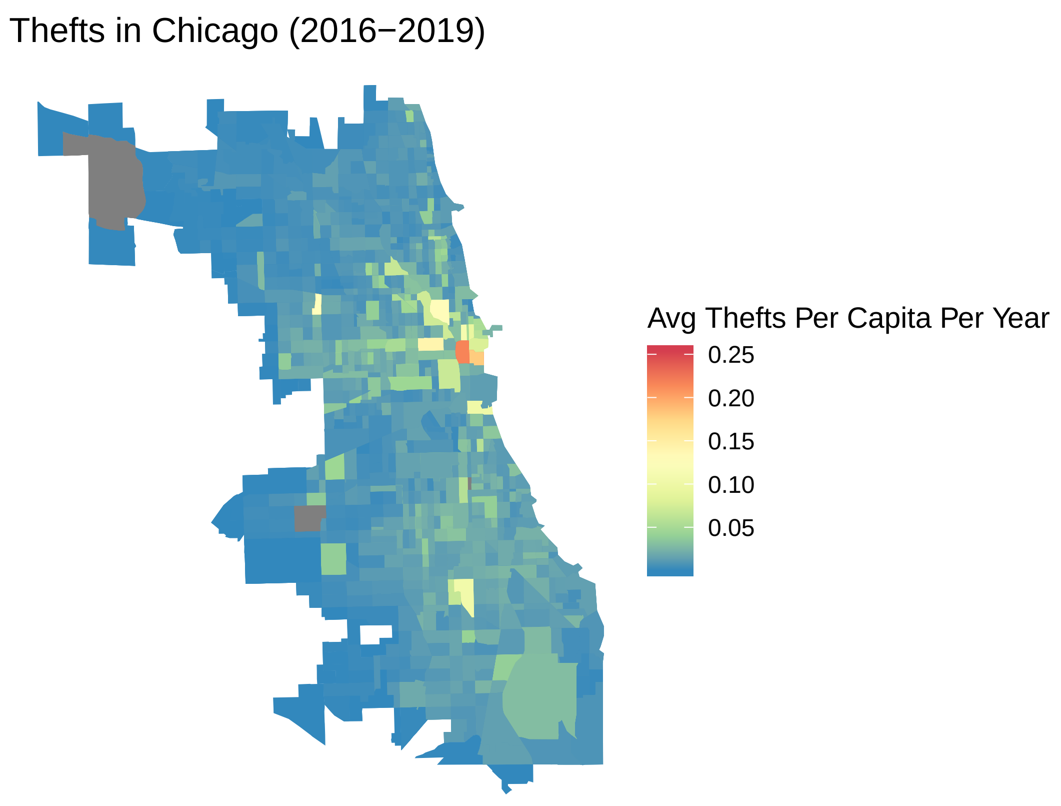
- Our goal is to determine which Census tract each theft occured in. To do so, we need to perform a point-inpolygon merge. Using st_join() to perform a point-in-polygon merge of the full thefts dataset and cook. We have to change the CRS of cook to 4326 before performing the join. We can do this with the st_transform() function. The lecture script contains a relevant example of the format for a point-in-polygon merge. We save the result of our point-in-polygon merge to a dataframe named thefts_merged.

```r
library(tidycensus)
census_data<-load_variables(2016,"acs5")
cook_county<- get_acs(
  geography = "tract",
  variables = "B01003_001",
  state = "IL",
  county= "Cook",
  geometry = TRUE
)
cook_county<-st_transform(cook_county,4326)
thefts_merged<-st_join(
  thefts1,
  cook_county,
  join=st_within
)
```

*The thefts_merged dataframe should now be a combination of the original thefts data and data from each theft's respective Census tract, including the tract's GEOID. We can aggregate by this GEOID column to get various summary statistics for each tract. Before aggregating however, get rid of the geometry column. The geometry column contains point geometries which are no longer needed after merging, and getting rid of it will speed up future operations. Setting the geometry column to NULL using the standard assignment operator (<-) or st_set_geometry(). Next we use, group_by() and summarize() to get the average number of thefts per year for each Census tract. Assigning the result to a new dataframe called thefts_agg.

*Finally, we join thefts_agg back to the cook dataframe using a simple left_join(). Drop any rows with NA values from our resulting dataset. In the joined data, create a new variable called thefts_pc equal to the number of thefts per capita for each tract. Finally, we replicate the map to the best of our ability.

```r
thefts_merged$geometry <- NULL
thefts_agg <- thefts_merged %>%
  group_by(NAME, year,GEOID)%>%
  summarise(avg_thefts = n())
theft_joined <- cook_county%>%
  left_join(thefts_agg, by = "GEOID")%>%
  drop_na()%>%
  mutate(thefts_pc = avg_thefts / estimate)
library(ggplot2)
library(sf)
ggplot()+
  geom_sf(
    data = theft_joined,
    aes(fill = thefts_pc), color =NA
  ) +
  theme_void()+
  labs(
    title = "Thefts in Chicago (2016-2019)",
    caption = "Source: City of Chicago Data Portal"
  )+
  scale_fill_distiller (
    name = "Avg Thefts Per Capita Per Year",
    palette = "Spectral"
  )
```



Source: City of Chicago Data Portal

**Why do you think thefts per capita is higher in the Loop and northwest side?

Answer: Since there is more population in the loop & Northwest side and a lot of tourists visit the area, a lot of petty crimes like pick-pocketing tend to occur there. A lot of crimes could also be reported in that areas when compared to others.

*What changes could we make to the map to further clarify the spatial distribution of thefts?

Answer: We can add petty vs grand data to know further information about the thefts.

Running a regression.

We use the tidycensus package to retrieve median household income, percent white, percent below the poverty line, and percent with a bachelor's degree for each Census tract in Cook County. We need to calculate the percentage values by dividing the number of people for each value by the total tract population (for example, 50 white people / 200 total people = 25% white). Additionally, we are choosing one additional variable from the Census to include in your regression.

Finally, we design and run a regression using lm() with thefts as our dependent variable and the Census variables as our independent variables. Printing the results of our regression model using summary() or the stargazer library.

```
cook_data<-get_acs(
  geography = "tract",
  state="IL",
  county = "Cook",
  variables = c(medium_income="B06011_001", white="B02001_002", total_pop="B01003_001", bachelors="B060
  year = 2016,
  geometry = TRUE
)
```

```
cook_data <-cook_data%>%
  select(-moe)%>%
  spread(variable,estimate)
cook_data<- cook_data %>%
  mutate(percent_white= (white/total_pop)*100) %>%
  mutate(percent_bachelors= (bachelors/total_pop)*100) %>%
  mutate(percent_below_poverty= (below_poverty/total_pop)*100)

cook_data <- st_transform(cook_data, 4326)
cook_data <- st_join(
  cook_data,
  theft_joined,
  join = st_within)

library(stargazer)
fit<-lm(thefts_pc~ medium_income +percent_white + percent_bachelors+ percent_below_poverty, data= cook_
stargazer(fit,type= 'text')
```

```
##
## ===================================================
##                          Dependent variable:
##                      ---------------------------
##                               thefts_pc
## ---------------------------------------------------
## medium_income                 0.00000***
##                               (0.00000)
##
## percent_white                 -0.0002***
##                               (0.00001)
##
## percent_bachelors              0.0004***
##                               (0.0001)
##
## percent_below_poverty          0.0002***
##                               (0.00003)
##
## Constant                       0.006***
##                                (0.001)
##
## ---------------------------------------------------
## Observations                     3,162
## R2                               0.119
## Adjusted R2                      0.118
## Residual Std. Error        0.016 (df = 3157)
## F Statistic            106.912*** (df = 4; 3157)
## ===================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

Questions that arise from our Regression Model:

1.Do you need to include weights in your regression? Fixed effects? Interaction terms? What should we do with missing values? Does our regression specification make sense?

Answer: Yes we need to include weights in our regressions because different variables effect the dependent variable differently or unevenly.We also need to include fixed effect and interaction terms as certain variables are dependent on each other. Take a larger smaple size to solve the problem of missing values.

2. Do you think the coefficients here are reasonable? Which one most influences the number of thefts?

Answer: The coefficients are not exactly reasonable, according to the table percentage of bachelors is influencing the number of thefts.

3.Are there variables that you think should be included? Can this regression be interpreted as causal?

Answer: We can include more variables like theft reports, criminal records,etc. It is not exactly causal.