HW 4

Due: Tuesday February 20, 2pm

Instructions

- Produce your assignment as a RMarkdown document rendered to pdf (knit to pdf).
- Also submit your Rmd file (it will not be graded but we want it for reference purposes).
- Show all the code (use `echo=TRUE, eval=TRUE` as option in R chunks) as well as the results.
- 100 total points (Points distribution is shown in [])
- See Syllabus for HW policies.
- For this HW, you will be using the "Chicago Crimes From 2001-Present" dataset, which is extracted from *Chicago Data Portal*. This dataset chronicles all crimes reported to the Chicago Police Department from 2001 to present.
- Due to the size of this dataset, **we will only analyze data recorded in the last 5 years**. For your convenience, we have extracted the relevant variables and saved the preprocessed dataset into Canvas on Files/data/crime_data.rds.
- Because TA created this assignment, please direct any questions regarding interpretation of the problems to the TA.
- Prior to beginning these exercises, make sure you have installed the required packages and loading their corresponding libraries. Run the following code to do this.

```
library(plyr)
library(dplyr)
library(tidyr)
library(lubridate)
library(ggmap)
library(ggplot2)
library(grid)
library(reshape2)
```

- For faster data import/export, we saved the dataset (crime_data.rds) in RDS format. Once you have downloaded the data and set your working directory, save it to `violence` using the following line of code:

```
violence <- readRDS('./crime_data.rds')
```

Exercise 1: [50]

Use `ggplot2` to create the following density plot to visualize the spatial distribution of `Assault` and `Homicide` over the city map of Chicago. The following subproblems will walk you through the process.

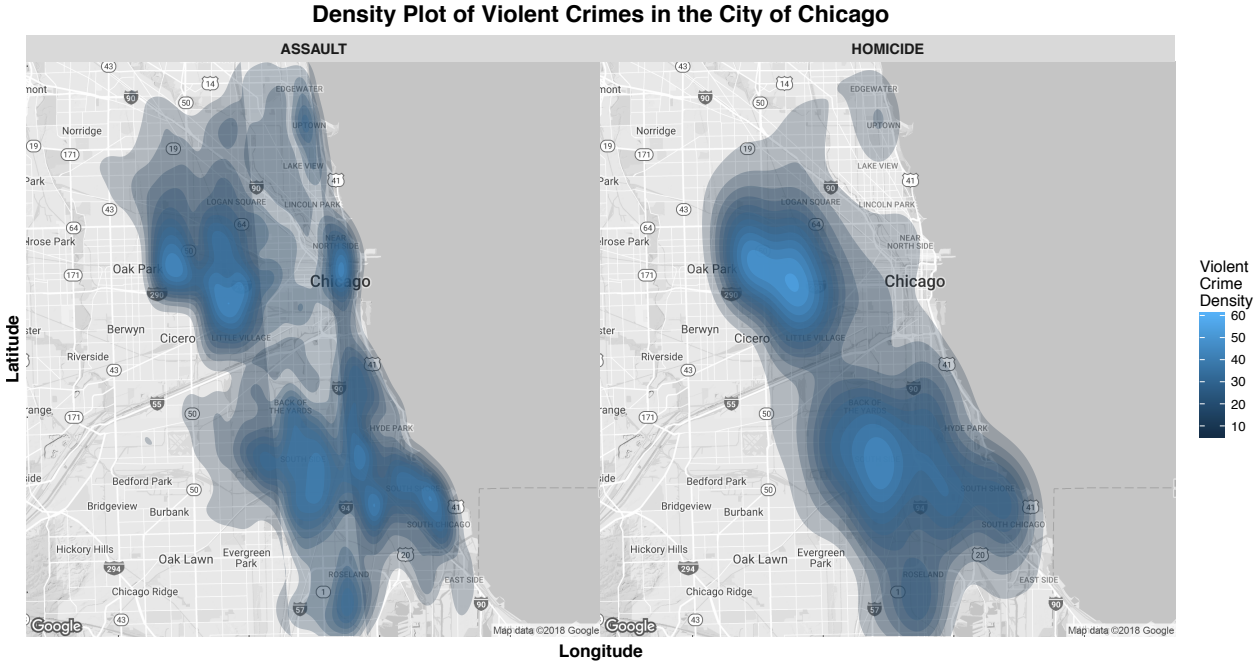
- Use `str()` to learn about data types of all variables in `violence`.
- Only keep observations corresponding to `Assault` and `Homicide` for this problem. [2]
- Use the `get_map()` function from `ggmap` package to load the city map of Chicago. Use the following line of code for the `location` argument in `get_map()`. Please go through the documentation of `get_map()` and set other arguments accordingly to **match** the sample plot. The sample plot uses map zoom of 11, map type of "terrain", black-and-white background and the source is Google Maps. Set the output of `get_map()` to variable `map`. [3]

```
location <- unlist(geocode('4135 S Morgan St, Chicago, IL 60609'))+c(0,.02)
```

- Perform 2D kernel density estimation by binning crime occurrence by `Longitude` and `Latitude`. This can be done using `stat_density_2d()` from `ggplot2`. Please go through the documentations of

`stat_density_2d()` and set the function arguments accordingly. Make sure you use the aesthetics argument `fill`, `alpha`, `geom`, and `size` to **match** the aesthetics of the sample plot. **Please set `n=500` to set number of grid points in each direction**. Set the output of `stat_density_2d()` to variable `contours`. [15]

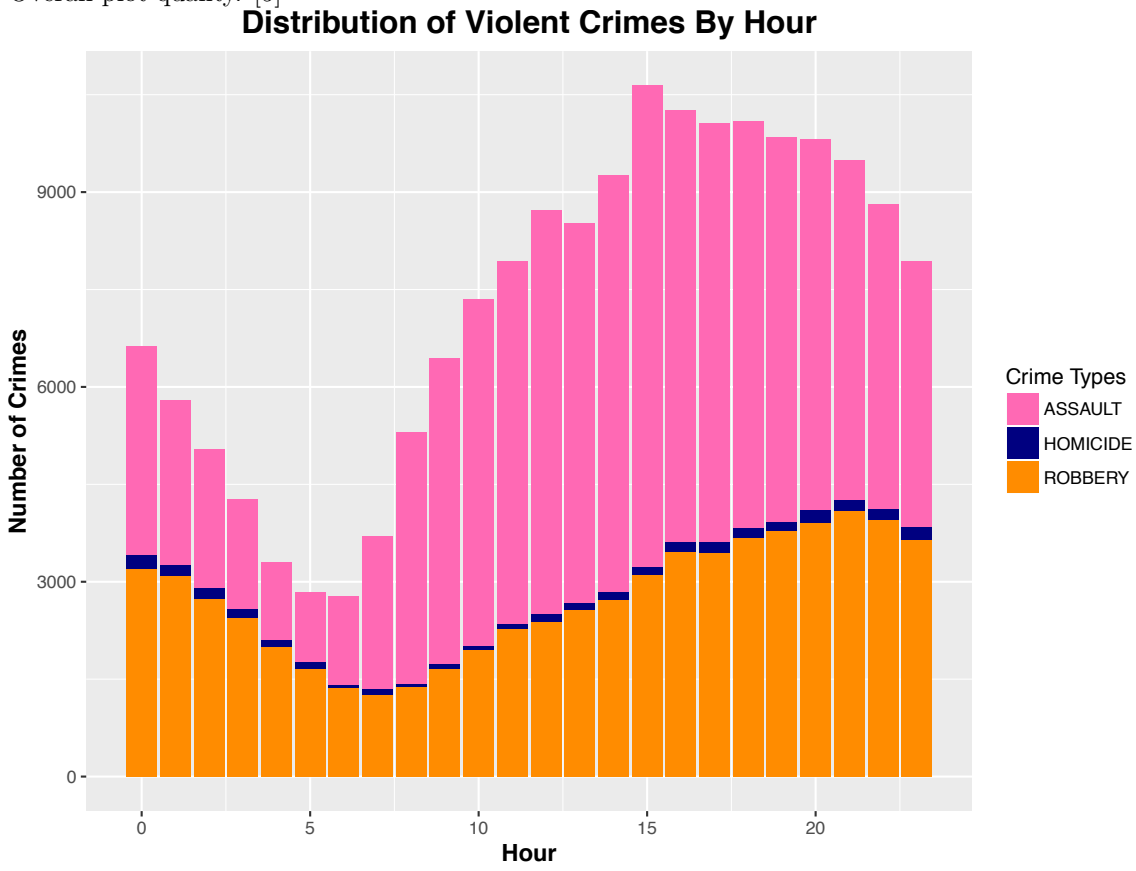
- Use `ggmap()` and results from previous parts (`map`, `contours`) to replicate the sample plot.
  - Make sure the spatial density of `Assault` and `Homicide` are displayed in two side-by-side panels with clearly labeled panel titles. [5]
  - Clearly label your x-axis and y-axis. Make sure the font sizes are big enough for readability. (Hint: use boldface and font size of at least 12) [3]
  - Make a plot title and make sure the title appears in the center of the plot (as in the sample plot). Use boldface and a font size larger than axis labels. [3]
  - Remove plot margins. [2]
  - Remove panel margins. [2]
  - Make sure your legend shows the color scale as well as legend label. [5]
  - Overall plot quality. [10]
  - Note 1: It is fine if `ggmap` gives a warning message about "Removed # rows containing non-finite values (stat_density2d)" because some of the coordinates in the dataset fall outside of the city map.
  - Note 2: You might have noticed that there is a vertical band in the density plot of `Assault`. Because rows corresponding to `Assault` in the original dataset has lots of missing values in some variables, we removed those rows during preprocessing and those rows have coordinates that fall in the vertical band. Due to missing data, the kernel density estimation method can't get a consistent density estimation in those regions and thus giving rise to the discontinuity in density estimation.



Exercise 2: [20]

Now that we have looked at the spatial distribution of violent crimes in Chicago, we want to explore if there is difference among different violent crimes in terms of time of occurrence. More specifically, we want to visualize the number of occurrence of `Assault`, `Homicide`, and `Robbery` during each hour of the day from 2012 to 2017.

- Since the dataset doesn't have variables denoting hour of the day, we need to extract the hour during which each crime occurred from the column `Date`. For example, if a crime was committed on `01/01/2012 01:40:00 AM`, we say that this crime was committed during the hour of 1 (i.e. 1 AM in the morning). Hint: both `as.POSIXct()` and the package `lubridate` are useful. [5]
- Use `ggplot2` to replicate the sample plot.
  - Clearly label your x-axis and y-axis. Make sure the font sizes are big enough for readability. (Hint: use boldface and font size of at least 12) [2]
  - Make a plot title and make sure the title appears in the center of the plot (as in the sample plot). Use boldface and a font size larger than axis labels. [2]
  - Remove plot margins. [2]
  - Make sure your legend shows the color scale as well as legend label. [2]
  - Manually set color to **match** the sample plot. The hex code for colors are `c('#FF69B4', '#000080','#FF8C00')`. [2]
  - Overall plot quality. [5]



Exercise 3: [30]

Now we want to explore how the number of occurrence of `Assault` and `Homicide` vary over the years. Specifically, we want to use `ggplot2` to plot the time series denoting number of occurrence of `Assault` and `Homicide` on each day from 2012 to 2017.

- Use both `date()` function from package `lubridate` and `as.POSIXct()` to extract date from the column `Date`. [3]
- Create a data frame with 3 columns - `Date`, `Number.of.Assault`, and `Number.of.Homicide`. More specifically, you need to store the dates on which there are at least 1 occurrence of `Assault` **AND** `Homicide` in the `Date` column in this new data frame. Then store the number of `Assault` and `Homicide` on those dates in the columns `Number.of.Assault` and `Number.of.Homicide`, respectively. Your resulting data frame should have 1539 rows. [12]
- Use the `melt()` function from package `reshape2` to modify the new data frame which you created in the previous part. Because you want to plot both time series on the same plot but color each time series according to the crime type (i.e. `Assault` or `Homicide`), this step prepares the data frame which `ggplot` needs to create the time series plot. Specifically, you need to have a `Date` column, a column which labels each row as either `Assault` or `Homicide`, and a column denoting the number of crimes (either `Assault` or `Homicide`) on that date. [5]
- Make the time series plot. For this plot, you can choose any two colors that you like.
  - Clearly label your x-axis and y-axis. Make sure the font sizes are big enough for readability. (Hint: use boldface and font size of at least 12) [2]
  - Make a plot title and make sure the title appears in the center of the plot (as in the sample plot). Use boldface and a font size larger than axis labels. [1]
  - Make sure your legend shows the color scale as well as legend label. [2]
  - Overall plot quality. [5]