Stat 385 Final Project Proposal

Team Member:

- Jiayi Chen (jchen246)
- Vetrie Senthilkumar (vetries2)
- Tae Kyung Lee (tkl2)

Part 1: Introduction

For our project, we chose to analyze crime in Chicago to identify patterns and trends that could potentially be used to reduce crime in one of America's most troubled cities. We obtained our dataset directly from the City of Chicago's publicly available data. We chose this topic because we considered it to be a very relevant issue that needs to be addressed. Chicago's reputation as a violent city has continued growing in recent years, to the point where many of us have been desensitized to acts of brutality such as assault, robbery, and homicide since we see it every day on the news. In fact, Chicago had 762 homicides in 2016, its highest homicide rate in 19 years, according to CNN. During this time span, it seems that the city has been unable to find any feasible, permanent solutions to its crime problems. Thus, due to the severity of the issue and the fact that many of our fellow students come from Chicago or neighborhoods near it, we felt invested in this topic and thought that it could be a rewarding challenge to take on.

We plan to approach this problem by dividing the city into locations based on the severity and prevalence of crime in that specific region. We will first divide the locations by district, then by street address, and finally by latitude and longitude. This will allow us to use Google Maps to isolate specific areas with high rates of crime. This information would be useful in directing the focus of the Chicago Police Department to the most problematic, crime-riddled areas, helping prevent crime before it occurs. Furthermore, we would like to analyze which types of crime (narcotics, assault, robbery, etc.) occur most frequently in each location since different types of crime could require different strategies by the police department. The dataset also provides us with the setting (residence, street, hotel, etc.) in which each crime occurred. We plan on scrutinizing this information to see if any correlations emerge between certain types of crimes and certain settings. This information might not be immediately useful to us but could prove valuable to forensics teams when they examine the setting of the crime. Finally, we also plan on tracking crime throughout the past couple years to gauge whether the crime situation is improving or worsening in certain locations and in the city of Chicago overall. To accomplish our goals for this project, we must use statistical programming methods to analyze, extract, and visualize data. Since the main purpose of our project is pinpointing locations within Chicago that have the highest crime rates, we need to be able to subset out data based on variables like district, street address, and latitude and longitude. Furthermore, our data has a tabular structure so knowledge about data frames and vectorized operations is essential to analyze and extract data by rows and columns within our data frame. In addition, our dataset contains a lot of text-based information which requires proficiency in regex to process and analyze. Also, we plan on creating graphs using ggplot to show how the crime rates for different types of crime and crime in general have changed over the years. These are just a few examples of how our project will utilize the statistical programming methods taught in this class.

Part2: Related Work

Link to project: related_project_page

Link to dataset: related_dataset_page

I've included a link to a project we used as inspiration above. This project is a good example of the types of analysis and types of visualizations we would like to include in our project. One aspect of this project we found impressive was the time series analysis and the user interface that allowed users to interact with the time series graphs. These graphs made it easy to observe how crime rates varied over time. We also noticed how this project separated the different types of crimes into categories and provided visualizations on their frequencies. Thus, it was simple to understand which types of crime were more prevalent and problematic and which types were less of an issue. Another feature that caught our attention was the map of the Chicago region that showed what type of crime occurred where and the heatmap that accompanied it. We feel that including a similar geographical overview makes the project feel more applicable and realistic and gives a different perspective on crime's impact on chicago.

Although we plan to use some ideas from existing projects, we also want our project to be original with its own unique components. We wanted to use the Google Maps API to create a more detailed map of Chicago that would allow users to mouse over regions to obtain the district the region belongs too and the crime info of that region. Furthermore, when doing our analysis, we want to narrow down problematic locations even further. The project above shows which districts have the highest crime rates but we want to be able identify locations on a smaller scope, such as certain streets or maybe even a range of longitude and latitude coordinates. In addition, we also wanted to do further analysis on whether certain types of crime are associated with a particular setting and how the crime rate varies based on the season.

Part3: Method

3.1 General Overview of Code Structure

1. We will use read.csv to load our Chicago crime .csv file into R and store it into a data frame.
2. We will initially use methods like head(), tail(), summary(), etc. to gain some insight about our data set before we proceed with our analysis.
3. Next, we will utilize different subset methods to extract the information of certain columns within the data frame. For categorical variables that contain textual info, we will need to use regex and stringr methods like str_detect() and str_extract()
4. Then, we must pipe in the subsetted data to ggplot2 functions like geom_hist() to graph the distribution of numeric variables and geom_bar() to graph the distribution of categorical variables
5. Next, we will plot different time series graphs that shows fluctuations in Chicago's crime rate using ggplot's geom_line(). To do this, we need to extract dates and different types of crimes from our data set using dplyr methods such as filter() and select(). We can also use facet_wrap to plot similar time series graphs for individual districts.
6. Afterward, we will identify the locations with the highest crime rates by subsetting our data using dpylr based on variables such as street name and latitude and longitude. We can then use ggplot, particularly geom_bar(), to display the crime composition for these locations.
7. Now that we've identified problematic areas within Chicago, we will write code that accesses the Google Maps API and pinpoints these locations, creating a heatmap of crime across Chicago. We also want to create another map where individual acts of crime are plotted in different colors, with each color representing a different type of crime.
8. Finally, we will use Shiny to transform our project into a web app with a user interface. We would like to allow users to interact with the maps created in the step above. For example, we plan on using methods such as selectInput() and numericInput() to allow users to choose the type of crime and year they are interested in analyzing. The maps would then change according to their input.
9. We would also like our web app to include some of the times series graphs we created in step 3. We expect to use the renderPlot() function for this part. A feature we are interested in adding to our time series graphs is a scroll bar so users can observe in detail how crime patterns have varied over time by scrolling through our plots. We also want to add buttons to our plots to allow users to select

different periods of time (month, year, all time). To achieve this, we plan on using the sliderInput() and actionButton() functions.

3.2 Packages used in implementation of the project.

3.2.1 To load data

RMySQL - used to read in data from a database

XLConnect, xlsx - used to read and write Micrososft Excel files from R. You can also just export your spreadsheets from Excel as .csv's.

3.2.2 To manipulate/translate data

dplyr - Essential shortcuts for subsetting, summarizing, rearranging, and joining together data sets. dplyr is our go to package for fast data manipulation.

tidyr - Tools for changing the layout of data sets.

stringr - Tools for regular expressions and character strings.

lubridate - Tools that make working with dates and times easier.

dplyr - Summarizing data

3.2.3 To visualize data

ggplot2 - R's package for making graphics

ggvis - Interactive, web based graphics built with the grammar of graphics. maps - Easy to use map polygons for plots.

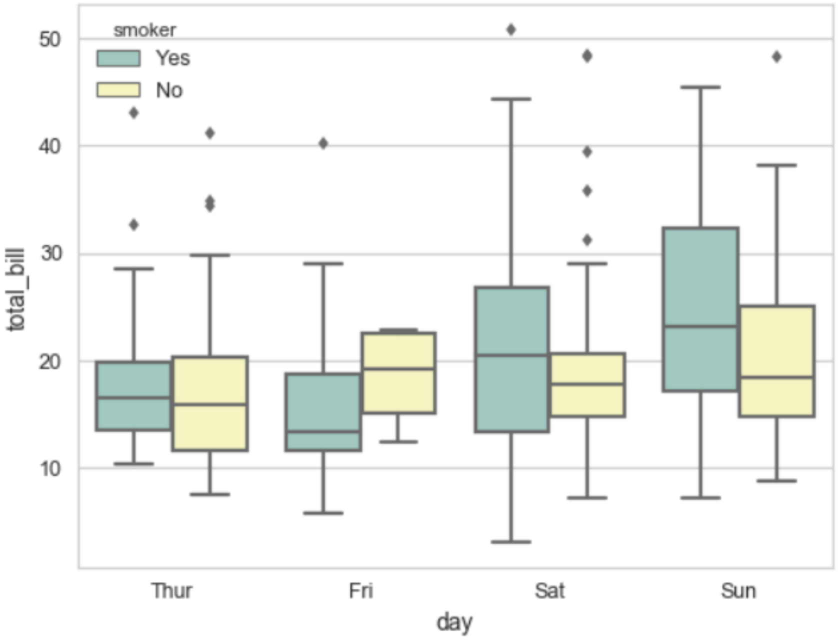
ggmap - Download street maps straight from Google maps and use them as a background in ggplots. gganimate - Show An Animation Of A Ggplot2 Object

3.2.4 To report results

shiny - A perfect way to explore data and share findings with non-programmers.

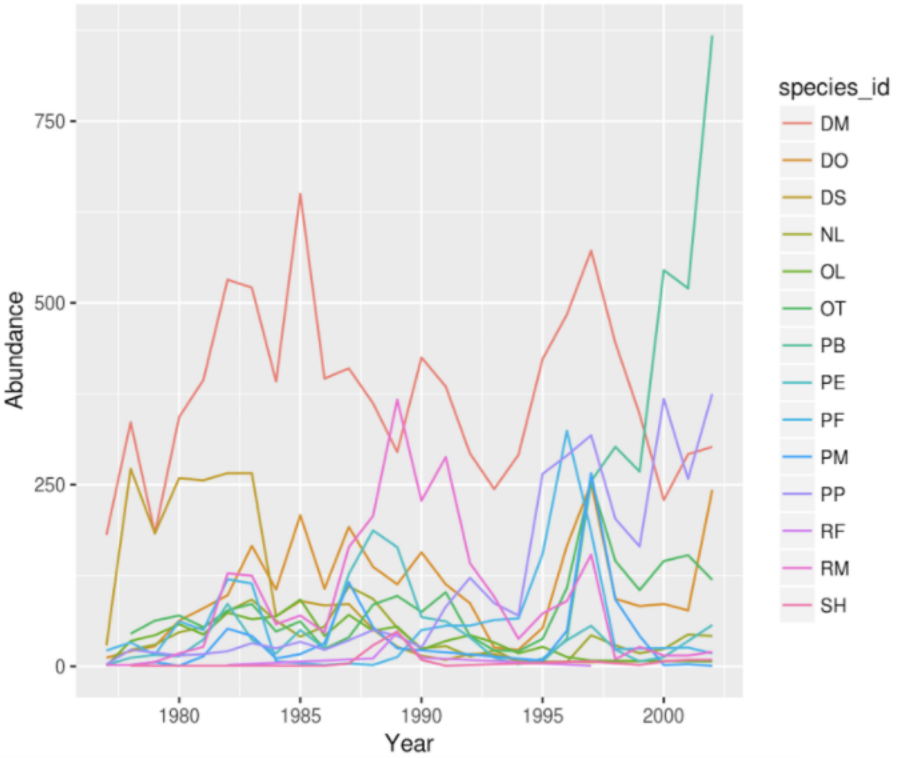
3.3 Visualisations or sample graphs:

3.3.1 Box-and-whisker Plot:



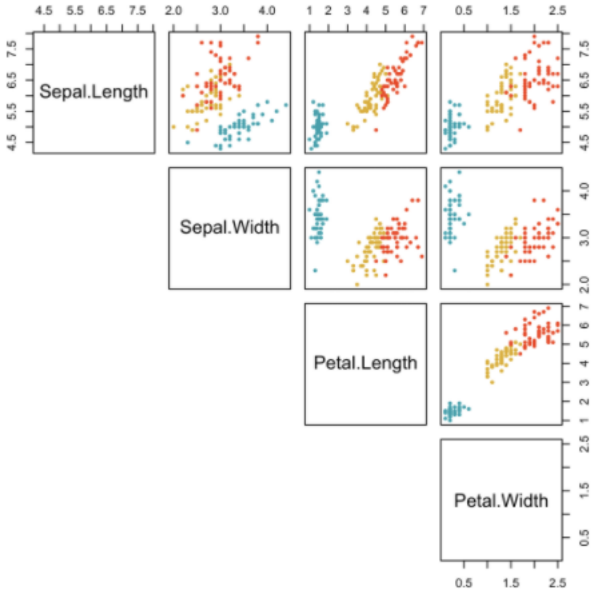
Display the distribution of ($date, *primarytype*,location description, $year)

3.3.2 Lineplot:



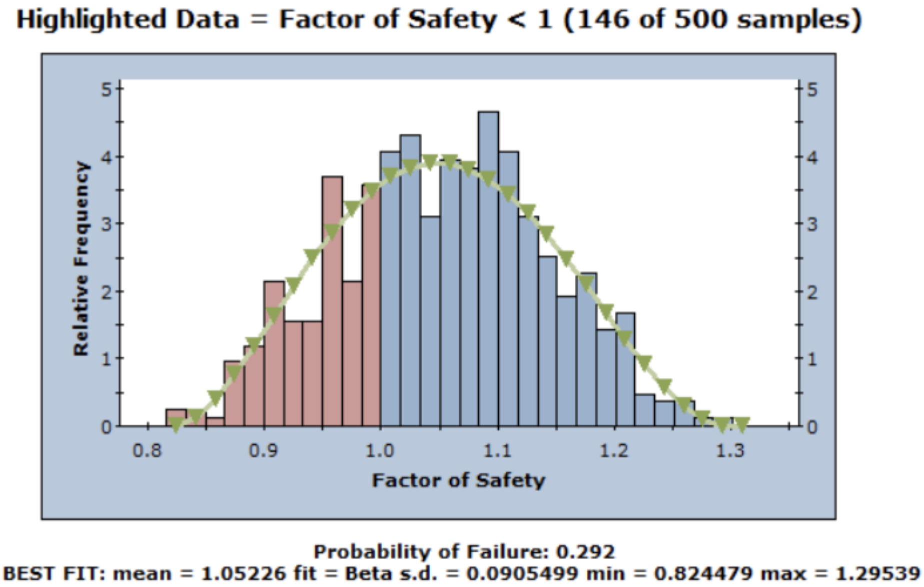
Display the distribution of ($date of the crime vs $different types of crime)

3.3.3 Matrix of scatterplots:



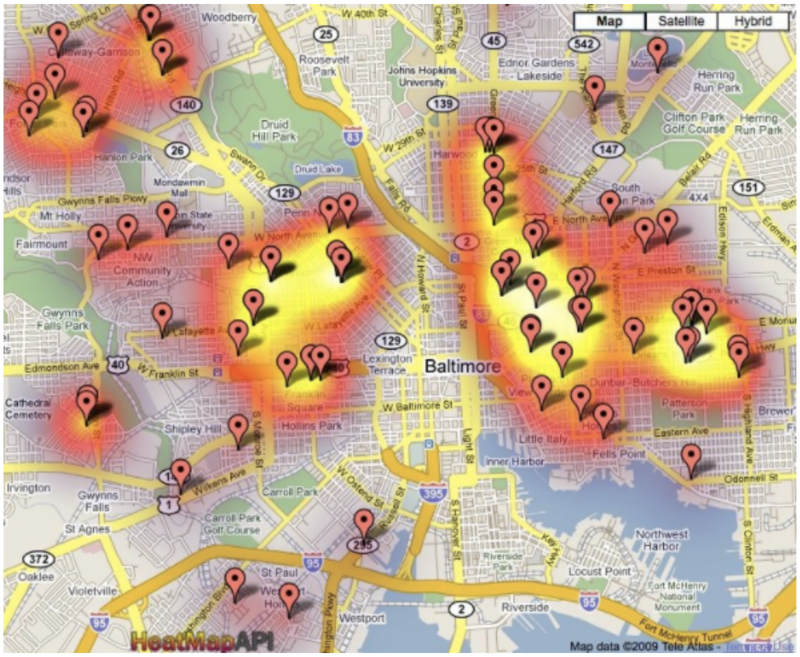
Plots a matrix of scatterplots. Each variable gets plotted against another. Such as (date, primary type, location description, year)

3.2.4 Histogram:

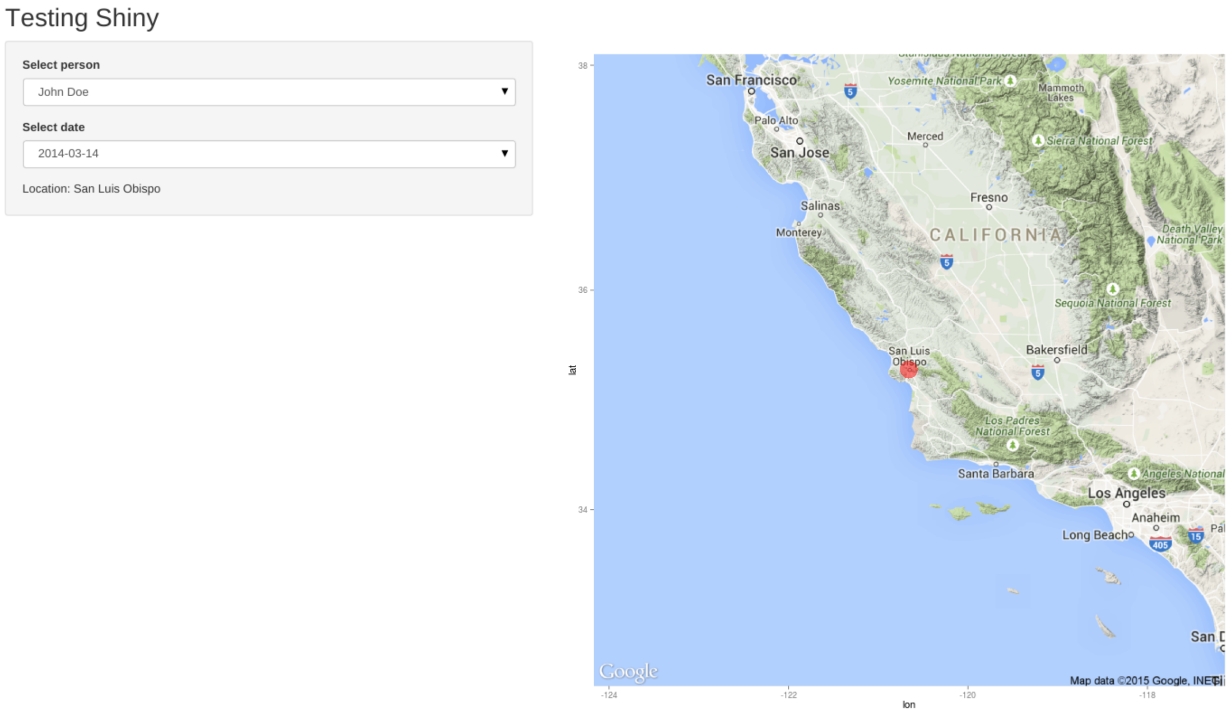


A histogram plots the frequency of observations. (Such as date, primary type, location description, year.)

3.3.5 Interactive Interface: Heatmap:



Heatmap is a graphical map representation of where different kinds of crime happened based on areas of Chicago. We plan to get access to location via Google map API. It gives a visualization where marks on a chart are represented as colors.



Testing Shiny

Select person

John Doe

Select date

2014-03-14

Location: San Luis Obispo

Figure 1:

3.4 Project Interface

Sample interface using shiny and google map api

Work cited: https://stackoverflow.com/questions/31836417/interacting-with-ggmap-in-shiny-can-the-user-zoom-scroll-etc

We plan to use Google map as the main interface of the project. We plan to project latitude and longitude into the Api and then Google map should generate a marker for each of the coordinates. And markers can be set in different color to distinguish different primary types of crimes on the map. In this way when users click on the link of a certain marker, it will show more details about a particular crime incident.

Also, we plan to implement some features using shiny that allow users to select from a drop-down list, such as $Date, $Primary type, $Community Area. And the system will filter the info that users entered to highlight certain markers. We think with the use of Google Maps API and Shiny, the interface is a great visual representation of the crime incidents in chicago. Furthermore, we plan on including some our time series graphs on our web app generated through Shiny. Users will be able to scroll through the graphs and click on time periods to see how crime rates have fluctuated over time.

3.5 Future Use of Project

The information obtained through this project could prove valuable in identifying patterns in criminal activities across Chicago. By analyzing Chicago in a smaller scope using latitude, longitude coordinates as well as specific streets, we could provide guidance to law enforcement about which parts of Chicago require the most attention. Furthermore, it would bring more attention to these troubled regions and this attention would hopefully manifest into new initiatives and programs to help those communities in need. Also, our heatmap could be used by both Chicago residents and visitors to avoid dangerous locations and use safer routes to navigate the Chicago region. Furthermore, our time series analysis graphs track how different types of crime have varied over the years. Law enforcement could use this information to adjust their approach to battling crime since different types of crime require different departments and different strategies to combat effectively. Lastly, we hope that our project will inspire others to search for answers to stopping Chicago's crime issues. Perhaps our project could even be used as a model for others who wish to conduct research and analysis on this issue. We feel that this is an important problem that deserves a solution and hope others feel the same after seeing our project.

Part 4: Feasibility

One of the challenges of the project was gathering a dataset that satisfied all of the requirements and captured our interest. Using the time series analysis and other types of visualizations, we are planning to create a project that illustrates how Chicago has been impacted by crime. Our project will use aspects from a previous project, but will implement our own unique ideas and vision. Specifically, our user interface will allow the user to interact with the details and info of criminal activity in Chicago. We feel the most difficult part of this project will be creating the user interface and using new concepts like shiny. We anticipate that this will take the most amount of time, but we are confident that the core idea behind our project can be completed by the end of the semester. After completing the project proposal, the group will spend the next two weeks working on the project demo video. We are planning to meet at least once or twice a week to ensure fluent communication across all members and combining our individual work to form a cohesive project.Thus, we have decided to meet after each class. Each member will have specific roles to complete. One member, Vetrie, will focus on data visualizations and initial data analysis. Vetrie will create graphs such as the box-and-whisker plot, line plot, and matrix plot. The other two members, Tae and Jiayi, will primarily devote their time to developing the user interface using Shiny and the Google Maps API.They will be responsible for implementing features like the heatmap and for adding tools like scroll bars, sliders, and buttons to allow users to interactively explore our data. Since Shiny is a new concept and the Google Maps API is something that hasn't been covered in class, we anticipate this part of the project to be time consuming and potentially frustrating. Therefore, we believe that this is a fair distribution of work across group members that will ensure that our project gets completed by the deadline.

Part 5: References

5.1 list (5+) of papers or items read to write this proposal:

- Azadeh Ansari and Rosa Flores, "Chicago's 762 homicides in 2016 is highest in 19 years", 2017, https://www.cnn.com/2017/01/01/us/chicago-murders-2016/index.html
- Garrett Grolemund, "Quick list of useful R packages", 2018, https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages
- Vivek Mangipudi, "ANALYSIS OF CRIMES IN CHICAGO 2001 - 2017", 2017, https://rstudio-pubs-static.s3.amazonaws.com/294927_b602318d06b74e4cb2e6be336522e94e.html
- Andrew V. Papachristos, "48 YEARS OF CRIME IN CHICAGO: A Descriptive Analysis of Serious Crime Trends from 1965 to 2013", 2013
- Sai Krishna Vithal Lolla, "Crime Occurrence Analysis in Chicago City" , 2013, https://www.jmp.com/about/events/summit2013/resources/Poster25_Lolla_Liu.pdf Udeh Tochukwu, "CRIME MINING AND INVESTIGATION USING R" , 2015, https://www.researchgate.net/publication/277020625_CRIME_MINING_AND_INVESTIGATION_USING_R

5.2 list all R packages or software referenced:

```r
citation(package="XLConnect")
```

```
##
## To cite package 'XLConnect' in publications use:
```

```
##
##   Mirai Solutions GmbH (2018). XLConnect: Excel Connector for R. R
##   package version 0.2-15. http://www.mirai-solutions.com
##   https://github.com/miraisolutions/xlconnect
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {XLConnect: Excel Connector for R},
##     author = {{Mirai Solutions GmbH}},
##     year = {2018},
##     note = {R package version 0.2-15},
##     url = {http://www.mirai-solutions.com
## https://github.com/miraisolutions/xlconnect},
##   }
```

```
citation(package="dplyr")
```

```
##
## To cite package 'dplyr' in publications use:
##
##   Hadley Wickham, Romain Francois, Lionel Henry and Kirill Müller
##   (2017). dplyr: A Grammar of Data Manipulation. R package version
##   0.7.4. https://CRAN.R-project.org/package=dplyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {dplyr: A Grammar of Data Manipulation},
##     author = {Hadley Wickham and Romain Francois and Lionel Henry and Kirill Müller},
##     year = {2017},
##     note = {R package version 0.7.4},
##     url = {https://CRAN.R-project.org/package=dplyr},
##   }
```

```
citation(package="tidyr")
```

```
##
## To cite package 'tidyr' in publications use:
##
##   Hadley Wickham and Lionel Henry (2018). tidyr: Easily Tidy Data
##   with 'spread()' and 'gather()' Functions. R package version
##   0.8.0. https://CRAN.R-project.org/package=tidyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions},
##     author = {Hadley Wickham and Lionel Henry},
##     year = {2018},
##     note = {R package version 0.8.0},
##     url = {https://CRAN.R-project.org/package=tidyr},
##   }
```

```
citation(package="lubridate")
```

```
##
## To cite lubridate in publications use:
##
##   Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made
##   Easy with lubridate. Journal of Statistical Software, 40(3),
##   1-25. URL http://www.jstatsoft.org/v40/i03/.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Dates and Times Made Easy with {lubridate}},
##     author = {Garrett Grolemund and Hadley Wickham},
##     journal = {Journal of Statistical Software},
##     year = {2011},
##     volume = {40},
##     number = {3},
##     pages = {1--25},
##     url = {http://www.jstatsoft.org/v40/i03/},
##   }
```

```
citation(package="dplyr")
```

```
##
## To cite package 'dplyr' in publications use:
##
##   Hadley Wickham, Romain Francois, Lionel Henry and Kirill Müller
##   (2017). dplyr: A Grammar of Data Manipulation. R package version
##   0.7.4. https://CRAN.R-project.org/package=dplyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {dplyr: A Grammar of Data Manipulation},
##     author = {Hadley Wickham and Romain Francois and Lionel Henry and Kirill Müller},
##     year = {2017},
##     note = {R package version 0.7.4},
##     url = {https://CRAN.R-project.org/package=dplyr},
##   }
```

```
citation(package="ggplot2")
```

```
##
## To cite ggplot2 in publications, please use:
##
##   H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
##   Springer-Verlag New York, 2009.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     author = {Hadley Wickham},
##     title = {ggplot2: Elegant Graphics for Data Analysis},
##     publisher = {Springer-Verlag New York},
##     year = {2009},
##     isbn = {978-0-387-98140-6},
##     url = {http://ggplot2.org},
##   }
```

```r
citation(package="maps")
```

```
##
## To cite package 'maps' in publications use:
##
##   Original S code by Richard A. Becker, Allan R. Wilks. R version
##   by Ray Brownrigg. Enhancements by Thomas P Minka and Alex
##   Deckmyn. (2018). maps: Draw Geographical Maps. R package version
##   3.3.0.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {maps: Draw Geographical Maps},
##     author = {Original S code by Richard A. Becker and Allan R. Wilks. R version by Ray Brownrigg. Er
##     year = {2018},
##     note = {R package version 3.3.0},
##   }
##
## ATTENTION: This citation information has been auto-generated from
## the package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

```r
citation(package="ggmap")
```

```
##
## To cite ggmap in publications, please use:
##
##   D. Kahle and H. Wickham. ggmap: Spatial Visualization with
##   ggplot2. The R Journal, 5(1), 144-161. URL
##   http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     author = {David Kahle and Hadley Wickham},
##     title = {ggmap: Spatial Visualization with ggplot2},
##     journal = {The R Journal},
##     year = {2013},
##     volume = {5},
##     number = {1},
##     pages = {144--161},
##     url = {http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf},
##   }
```

```r
citation(package="shiny")
```

```
##
## To cite package 'shiny' in publications use:
##
##   Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan
##   McPherson (2017). shiny: Web Application Framework for R. R
##   package version 1.0.5. https://CRAN.R-project.org/package=shiny
```

```
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {shiny: Web Application Framework for R},
##     author = {Winston Chang and Joe Cheng and JJ Allaire and Yihui Xie and Jonathan McPherson},
##     year = {2017},
##     note = {R package version 1.0.5},
##     url = {https://CRAN.R-project.org/package=shiny},
##   }
```

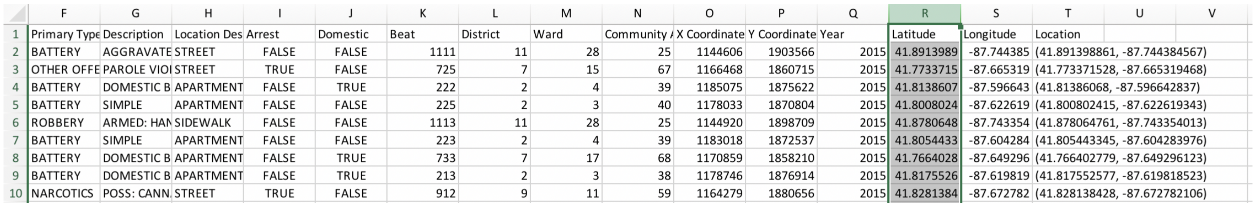
Part 6: Appendix

6.1 sketches of visualisations or the shiny application:

Already reported in 3.4 Project Interface

6.2 Functions Overview:

- read.csv() takes in a .csv file as input and returns a data frame that contains the data of the .csv file. Since our dataset is contained in a .csv file, this method is essential to our project and will be the first method called
- head() accepts a data structure as input and returns the first six elements. We will use the head() function on our data frame created above to gain some more insight into the structure of our data, especially the first couple observations, before we begin a more in-depth analysis
- tail() accepts a data structure as input and returns the last six elements. Similar to head(), we will use tail() to gain more insight about our data before we begin analyzing it
- summary() is a generic function that accepts an object as input and returns a statistical summary of the data the object contains. We will use summary() on our data frame to get a better sense of the distributions of different variables the data contains. Like head() and tail(), summary() will help us get a better understanding of our data before the analysis phase.
- str_detect() accepts two parameters in the following order, a vector of strings or string object and a regex pattern to search for within each string input. Based on whether or not the pattern can be found in each string, - str_detect() will return a sequence of booleans. We plan on using str_detect() to find the number of occurrences a certain type of crime occurs. Since the different types of crime are represented as text, str_detect() will be a useful method in isolating specific types of crimes from our data set. We can then produce visualizations for these variables using ggplot's functions.
- str_extract() accepts the same two parameters that str_detect() does. If the regex pattern is found in a string, it will return the portion of the string that matches the regex pattern. We plan on using str_extract() to extract street names from our data set so we can identify which streets are the most dangerous. It can also be used to extract the year from the date variable.
- geom_bar() when used with ggplot() takes in a column of a data frame as input and produces a bar graph based on the categorical variable. We plan on producing graphs for the text-based variables that were derived using str_detect and str_extract above using geom_bar()
- geom_hist() is similar to geom_bar() but is used to graph the distributions of quantitative variables instead. We will use geom_hist() to produce visualizations for numeric variables in our data
- geom_line() is another important ggplot function. It accepts an x variable and a y variable and produces a line graph based on these inputs. We plan on using geom_line() to produce our time series analysis graphs, using the date as our x variable and number of crime occurrences as our y variable. These



	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Primary Type	Description	Location Des	Arrest	Domestic	Beat	District	Ward	Community A	X Coordinate	Y Coordinate	Year	Latitude	Longitude	Location		
2	BATTERY	AGGRAVATE	STREET	FALSE	FALSE	1111	11	28	25	1144606	1903566	2015	41.8913989	-87.744385	(41.891398861, -87.744384567)		
3	OTHER OFFE	PAROLE VIOI	STREET	TRUE	FALSE	725	7	15	67	1166468	1860715	2015	41.7733715	-87.665319	(41.773371528, -87.665319468)		
4	BATTERY	DOMESTIC B	APARTMENT	FALSE	TRUE	222	2	4	39	1185075	1875622	2015	41.8138607	-87.596643	(41.81386068, -87.596642837)		
5	BATTERY	SIMPLE	APARTMENT	FALSE	FALSE	225	2	3	40	1178033	1870804	2015	41.8008024	-87.622619	(41.800802415, -87.622619343)		
6	ROBBERY	ARMED: HAN	SIDEWALK	FALSE	FALSE	1113	11	28	25	1144920	1898709	2015	41.8780648	-87.743354	(41.878064761, -87.743354013)		
7	BATTERY	SIMPLE	APARTMENT	FALSE	FALSE	223	2	4	39	1183018	1872537	2015	41.8054433	-87.604284	(41.805443345, -87.604283976)		
8	BATTERY	DOMESTIC B	APARTMENT	FALSE	TRUE	733	7	17	68	1170859	1858210	2015	41.7664028	-87.649296	(41.766402779, -87.649296123)		
9	BATTERY	DOMESTIC B	APARTMENT	FALSE	TRUE	213	2	3	38	1178746	1876914	2015	41.8175526	-87.619819	(41.817552577, -87.619818523)		
10	NARCOTICS	POSS: CANN.	STREET	TRUE	FALSE	912	9	11	59	1164279	1880656	2015	41.8281384	-87.672782	(41.828138428, -87.672782106)		

Figure 2: dataset

variables will be obtained using str_extract() and dplyr, which is discussed below. Furthermore, we can use the facet wrap feature to produce similar line graphs for each individual district

- filter() is a dplyr function that takes in a tabular data structure and a logical condition and returns the rows of the tabular data structure for which the logical condition evaluates to true. We plan on using filter() to subset our data so we can obtain the frequencies of each type of crime. This will then be used in our time analysis graphs

- select() is another useful dplyr function that accepts a tabular data structure and a sequence of variable names. It will then return a similar data structure that only contains the variables specified. We felt that using select() is an easy way to access specific sets of variables within our data. For example, the variables latitude and longitude are highly related and should be grouped together when they are extracted to pinpoint precise locations for the heatmap.

- sliderInput() is a Shiny UI input function that creates a slider for our user interface. It accepts parameters that allow us to set a label, set a default value the slider will be positioned at, and max and min values the slider cannot go beyond, in addition to numerous other optional parameters. We hope to incorporate this feature in our time series analysis graphs to allow users to scroll through different time periods.

- actionButton() is a Shiny UI function that creates buttons users can click on to change the state of the user interface. It accepts parameters for the button's label and the dimensions of the button in addition to more optional parameters. We plan on using this function to create buttons for the time series analysis graphs that allow users to switch between different time periods.

- numericInput() is a Shiny UI function that allows users to enter in numeric data through the user interface. It accepts parameters for its label, default value, and max and min values the user can enter in, plus other optional parameters. We will implement this feature such that users can enter in a year within a given range and the user interface will display the crime statistics for that particular year.

- selectInput() is a Shiny UI function that allows users to select an option from a drop down menu that contains a list of possible choices. It accepts parameters for its label, list of choices, and default choice, in additional to other optional parameters. We will add this feature to our user interface to allow users to choose a specific type of crime they wish to analyze. The user interface would then display statistics and graphs for that particular type of crime.

- renderPlot() is a Shiny rendering function that allows us to add graphs and plots to our Shiny app. This function takes in an expression parameter that is used to generate the graph as well as height and width parameters that are used to set the dimensions of the plot. This function is essential since it will be used to add our data visualizations, especially our time series analysis plots, to our Shiny app.

6.3 Sample Data of 10 observations: