Using Kriging to Understand Property Crime Patterns in San Francisco

---

A Comprehensive Evaluation Report
Presented to
The Statistics Faculty
Amherst College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts
in
Statistics

Timothy W. Lee

February 2018

Acknowledgements

I'd like to thank many people for their support with this project. I'd like to thank my statistics adviser, Dr. Nicholas Horton for his support despite his incredibly busy schedule. I'd like to thank Dr. Xiaofei Susan Wang, former Amherst College faculty member now at Yale, for teaching Intermediate Statistics. It was the course that motivated me to become a statistics major. I would not have gotten to learn so much and experience so many wonderful opportunities without her help. I'd like to thank Hampshire College's Dr. Ethan Meyers for designing a course that let me explore the intersection of psychology and statistics. I'd like to thank Dr. Albert Kim for his fantastic lectures in the capstone course. I'd like to thank Dr. Amy Wagaman and Dr. Pam Matheson for their support in the comprehensive project. I'd like to thank the entire statistics department for their support in ensuring that each and every one of us is successful in learning statistics. I'd also like to thank Jonathan, Sarah, Pei, Vickie, and Leonard for the enjoyable times we've had with working on projects and other assignments together this past year in STAT-495. I'd like to also thank all my friends who have brought joy to my life. Finally, I'd like to thank my family for their unconditional love and support.

Table of Contents

Abstract

This study intends to continue the work that was completed in the STAT-495 final project. I continue my examination of the Kaggle San Francisco Crime dataset. Throughout the report, I examine the possiblity of ordinary kriging to better understand crime categorization. Instead of looking predicting 39 different possible categories, I predict property crime as defined by the FBI. Using this framework, I report my findings and conclude with future directions for understanding crime data.

Introduction

This report is an extension of my STAT-495 final project, Kaggle: San Francisco Crime Classification, that Sarah Teichman, Jonathan Che, and I completed in the fall of 2016. I am continuing the use of this dataset. For the STAT-495 final project, I examined crime data in San Francisco using a dataset from [Kaggle] (`https://www.kaggle.com/c/sf-crime`). My team analyzed the Kaggle San Francisco dataset that included approximately 12 years of crime reports in the city of San Francisco. We submitted our predictions to Kaggle in order to see how well we could predict crime. This time, however, I am also combining the dataset with spatial statistics. I have not been able to take a course in this topic, though I have always been interested in the influence of space and time on behavior.

Thus, the focus of my final project is understand the influence of location on crime through kriging. My group often saw trends in the data, but did not have statistical tools to prove that one area had more crime than another area. Thus, I extend on previous general trends that we saw with spatial statistical methods to predict the occurrence of crime across the entire San Francisco grid.

The report is organized as follows:

- In Chapter 1, I provide an exposition on kriging as applied to the `meuse` dataset that is preloaded into different R spatial packages. I provide a clear, classical use of kriging to demonstrate its capabilities with more "perfect" data.
- In Chapter 2, I provide additional background about the Kaggle San Francisco Crime dataset and additional exploratory analysis. There is also a Shiny application included in my data exploration.
- In Chapter 3, I apply the technique of kriging to the San Francisco crime dataset and discuss some conclusions.
- In Chapter 4, I summarize my conclusions and future directions.

Chapter 1

An Exposition on Kriging

1.1 Introduction

In this section, I provide an exposition on the technique of kriging. It's intuitive that spatial data nearly always has spatial correlation, as seen by common phrases like "location, location, location" in describing its significance in people's lives. A location measurement is generally related to the area around it. For many spatial measurements, it is generally safe to assume that nearby locations will be similar instead of having random, unrelated characteristics. This intuition, then, allows for interpolation. It is possible to predict values at a locations with no observed measurements given that we have nearby location data. One common interpolation technique on which I will focus is kriging. Additional techniques that are popular include "nearest neighbor" interpolation and "inverse distance weighted" interpolation, though I focus primarily on kriging in this report (Bivand, Pebesma, & Gomez-Rubio, 2008).

Kriging is a linear interpolation technique, which furthermore has types of kriging variations. The kriging techniquea that are most commonly used are ordinary kriging and universal kriging. Ordinary kriging involves a calculation of weighted linear combinations from the sample in order to predict values at unknown locations. Some variations of ordinary kriging would be to use universal kriging or regression kriging, which further incorporates spatial trends and additional non-coordinate variables (Heuvelink, 2015).

At a high-level, kriging consists of exploring the data, creating a variogram, fitting the variogram, and creating predictions. In the below sections, I explain ordinary kriging using a dataset that makes understanding the analyses more straightforward. After providing an exposition, I apply kriging to the Kaggle San Francisco Crime

Table 1.1: Meuse Dataset

x	y	cadmium	copper	lead	zinc	elev	dist
181072	333611	11.7	85	299	1022	7.909	0.0013580
181025	333558	8.6	81	277	1141	6.983	0.0122243
181165	333537	6.5	68	199	640	7.800	0.1030290
181298	333484	2.6	81	116	257	7.655	0.1900940
181307	333330	2.8	48	117	269	7.480	0.2770900
181390	333260	3.0	61	137	281	7.791	0.3640670

dataset.

1.2 Data Manipulation before Kriging with the `meuse` dataset

For an exposition on kriging, I will demonstrate the usefulness of kriging with a common dataset that is packaged within the `gstat` and `sp` packages. The package `gstat` has functions for geostatistical analysis (Pebesma & Graeler, 2017). The `sp` package includes functions and data structures that allow for data manipulation of spatial data (Pebesma et al., 2018).

The `meuse` dataset has measurements on soil pollution along the Meuse River. This dataset comes directed included in the `gstat` package and is included in the `gstat` package for demonstration. Although the Meuse River runs by multiple countries, the data was collected near the city of Maastricht in the Netherlands.

1.2.1 Data Import

After loading the packages, I examine the dataset. The `meuse` dataset is a `data.frame` type. The dataset includes measurements for the location, using the Universal Transverse Mercator (UTM) coordinate system with EPSG projection of 28992 (more projection code information can be found on this link (`http://spatialreference.org/`)). There are also measurements for various concentrations of cadmium, copper, lead, and zinc in the topsoil of the Meuse river flood plain. The units for the concentrations are in parts per million (ppm). Afterwards, there are also measurements of elevation and distance from the river in meters. Please refer to `Table 1.1` below.

1.2.2 Data Manipulation

The meuse dataset is imported as a normal data.frame structure that is usable in ggplot2 and other tidyverse packages. However, in order to use gstat for spatial analysis, it is important to change the data structure such that it is compatible. The observations are given in locations in UTM, which means that it is important to set the projection of the meuse dataset as much. This will change the data.frame object into a SpatialPointsDataFrame object. In this format, it is possible to do data analysis in R.

1.2.3 Data Exploration

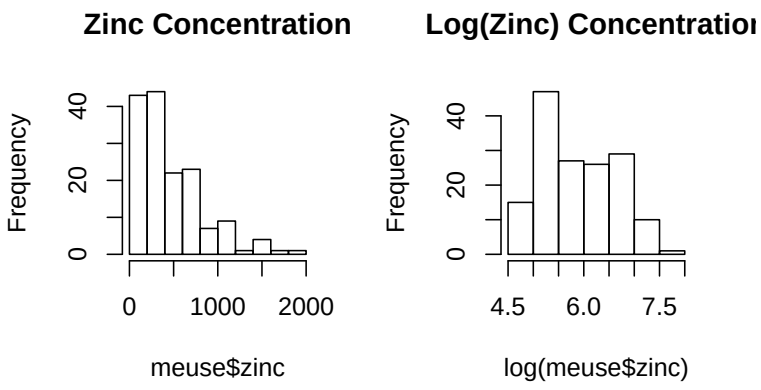
Similiar to standard exploratory analysis, it is also possible to do exploratory analysis with a SpatialPointsDataFrame. There seems to be a spatial trend in that zinc values tend to be higher closer to the river bank. Please note that standard kable() functions do not work for a SpatialPointsDataFrame.

```
#Summary of the zinc values
summary(meuse$zinc)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  113.0   198.0   326.0   469.7   674.5  1839.0
```

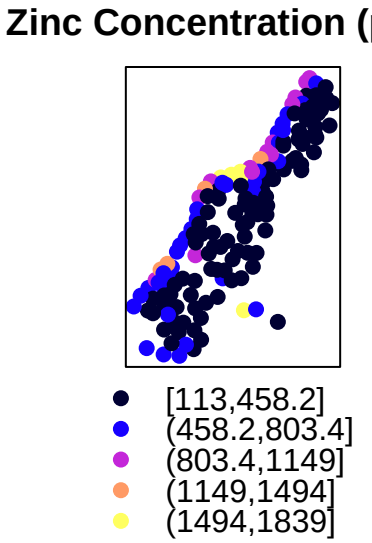
Data Transformation

Because the data are skewed, it may also be better to transform the zinc values, as seen by the histograms.



Using a `sp` Plot

The `sp` package also has the ability to show concentrations spatially. While it is also possible to produce the below plot in `ggplot2`, the `sp` package has capabilities to interact with a spatial dataframe, so there is no need to convert the dataset. It appears that there is a difference in concentration along the edges of the data showing that there may be a spatial trend along the river.



1.2.4 Fitting a linear trend surface

Since our exploration seems to show a relationship, we can further examine the dataset by simply fitting a linear regression model, as recommended by Bivand and colleagues (2008). There seems to be a trend in that the X and Y coordinates are significant at an alpha-level of 0.05. A linear regression does not take into account spatial correlation, but it can be useful to use as a quick analysis to see if there are any potential trends in the data. Furthermore, this may implicate that there is an orientation in the data. This may influence the type of kriging that we can use, as discussed later in this chapter. Nevertheless, before diving into more computationally-heavy functions with a spatial component, examining a linear regression can be helpful.

Table 1.2: Basic Linear Regression

	coefficient	Std. Error	t-value	p-value
(Intercept)	-42.870	16.132	-2.657	0.009
x	-0.001	0.000	-7.241	0.000
y	0.001	0.000	7.102	0.000

1.3 Variogram

In geostatistical analysis, a variogram shows that semivariance is a function of distance. In other words, there may be an association between the distance between points depending on how far the points are from each other. Near observations are more related than far observations. For spatial data, it is important to determine if this is true. One can visualize this with variograms. This will occur in multiple steps:

- First, I plot a variogram cloud.
- Second, I plot a variogram that uses the variogram cloud to produce a more interpretable plot.
- Third, I plot a variogram that uses different orientations to determine if there is isotropy.

The variogram can help us determine if the data satisfy the conditions necessary for kriging (Bohling, 2005; Bivand et al., 2008; Heuvelink, 2015):

1) stationarity assumption - the mean and variance become constant throughout the field
2) isotropy assumption - uniformity in the orientations of the semivariance

The semivariance calculation can be defined as below, with h being the distance. This is done for every possible pairing with n observations for a total of $\frac{1}{2} * n * (n-1)$ calculations.

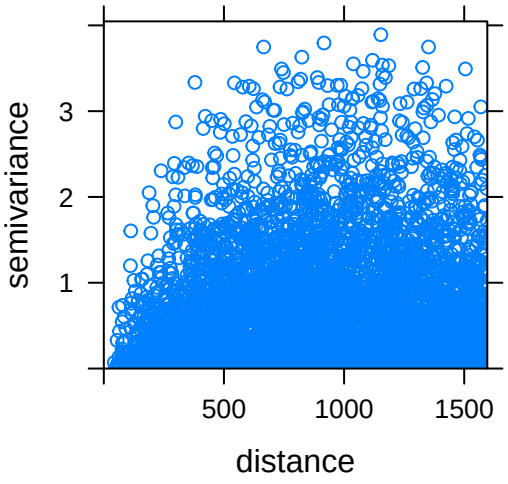
The semivariance is calculated as follows with h being a distance and x being an observation.

$$\gamma(h) = \frac{1}{2} E[(Z(x) - Z(x+h))^2]$$

1.3.1 Variogram Cloud Plot

The variogram cloud looks at how similar the data are and their relative distance. Separation is on the x-axis, and the semi variance is on the y-axis. The points are calculated for every combination of distance pairs between the points. A total of $\frac{1}{2} * n * (n-1)$ points are plotted. While this is useful for seeing the underlying calculations, it is difficult to interpret given the high number of points. Though we see a vague trend of semivariance increasing as distance increases.

```
vcloud <- variogram(log(zinc) ~ 1, data = meuse, cloud = TRUE)
plot(vcloud)
```

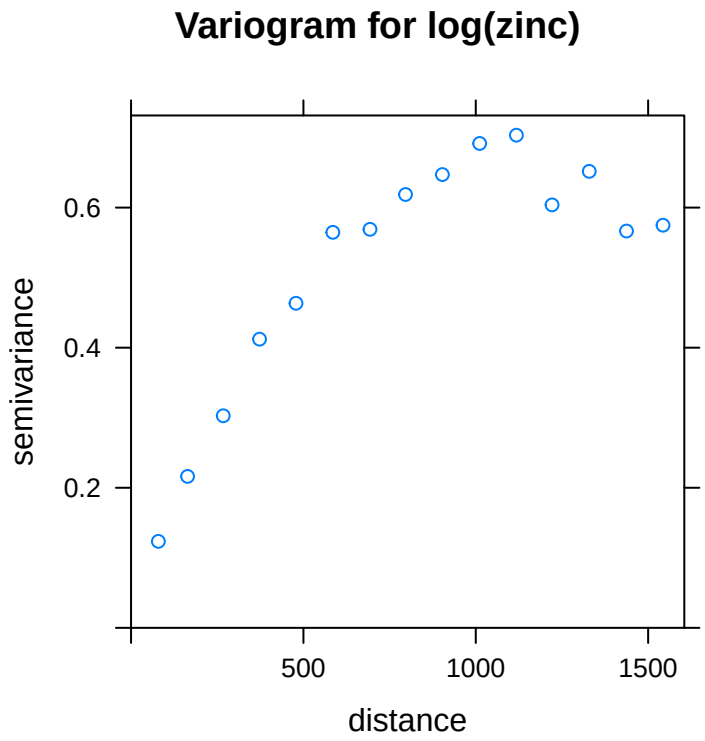


1.3.2 Variogram Plot

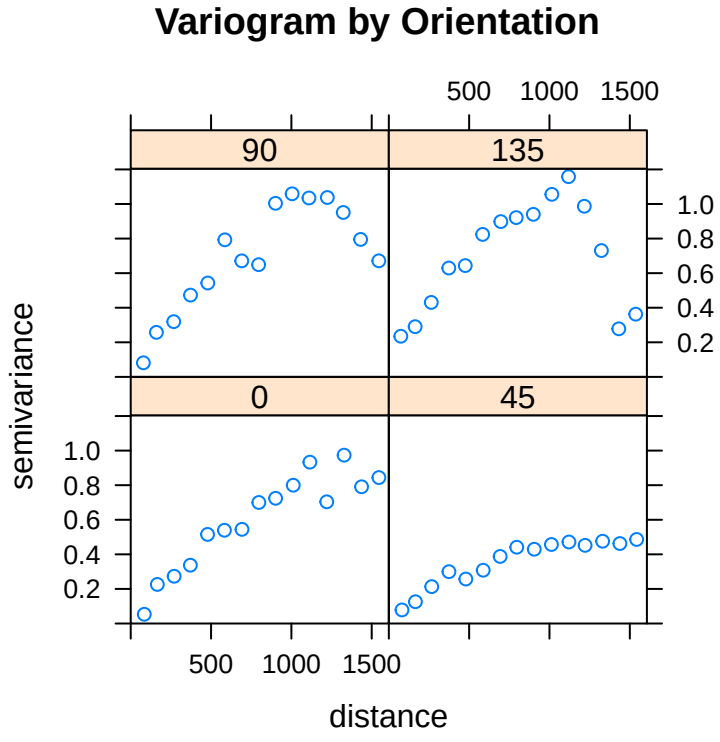
The variogram cloud plot can be difficult to interpret when every combination of two points is included. Instead, it can still be useful by averaging the data points within each bin. When dividing the bin into approximately sized bins, it can be easier to interpret. As distance increases, the semivariance increases. After a distance of about 800m, the plot levels off. This indicates that there seems to be little spatial correlation after that distance. This also indicates that it satisfies the stationarity condition, since the binned variogram does not diverge to infinity.

However, the data do not satisfy the isotropy condition, because the binned variograms are not consistent throughout different orientations. Nevertheless, for example purposes of demonstrating ordinary kriging, I will continue with the ordinary kriging example while also touching using universal kriging instead.

```
vgm <- variogram(log(zinc) ~ 1, data = meuse, cloud = FALSE)
plot(vgm, main = "Variogram for log(zinc)")
```




```
vgm.aniso <- variogram(log(zinc) ~ 1, data = meuse,
                       alpha = c(0, 45, 90, 135))
plot(vgm.aniso, main = "Variogram by Orientation")
```

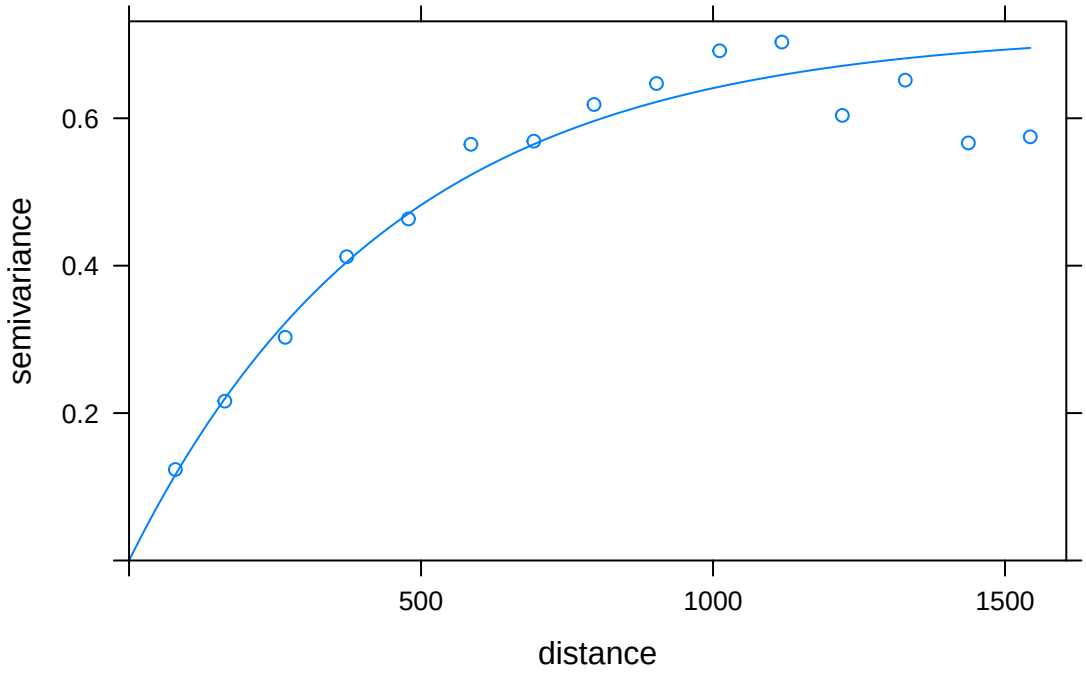


1.3.3 Fitted Variogram

After plotting the variogram, it is possible fit a variogram model. There are three variable inputs that need to be specified (Data Camp, 2018). Three of those features that are important include the (1) nugget, (2) the sill, and (3) the range. The nugget is the semivariance value at the y-intercept, which is the zero distance. The sill refers to the location where the variogram levels off. This is usually reported as the difference between the sill and the nugget. The range is the point at which the sill is reached.

Five models that are common include the nugget, spherical, exponential, Gaussian, and power models. More information can be found in another resource (Bohling, 2005; Bivand et al., 2008). These models and their parameters (nugget, sill, and range) can be used to fit an optimal model.

```
nugget <- 0.1
psill <- 0.6
range <- 400

vmodel <- fit.variogram(vgm, model = vgm(
  model = "Ste",
  nugget = nugget,
  psill = psill,
  range = range
))

plot(vgm, model = vmodel)
```



1.4 Kriging

With a fitted variogram, it is possible to krige the data. Since kriging allows for prediction, I predict values over a grid of points. At its core, kriging is assigning weights to the mean of the residuals from the mean function. It is a non-linear regression that minimizes the variance. Below are the equations that explain the process of ordinary kriging.

Kriging Equation:

$$\hat{Z}(x_0) = \sum_{i=1}^{n} \lambda_i * Z(x_i)$$

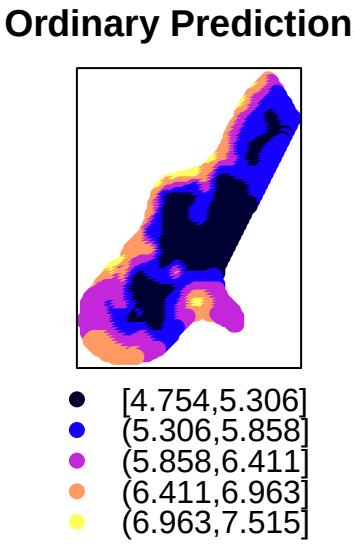
Minimizing Variance:

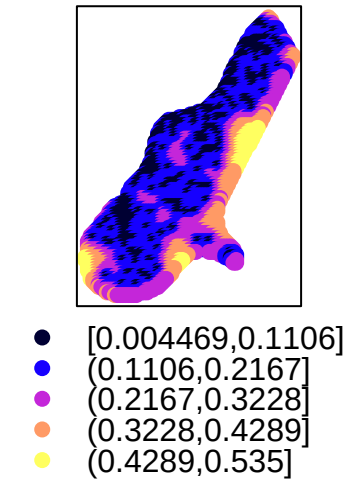
$\sigma^2(x_0) = E[(\hat{Z}(x_0) - Z(x_0))^2]$ under the unbiased condition of $\sum_{i=1}^{n} \lambda_i = 1$

The plots for the prediction and variance for the ordinary kriging estimates are below. Areas along the left edge, which corresponds to the river, seem to have higher concentrations of zinc. Areas that are landlocked, however, do not seem to have higher concentrations of zinc. The variance is lower for the left edge, because there are probably more data points being collected After all, there would likely be more measurements along the river because a landlocked plot of land may not provide much useful information.

```
data(meuse.grid)
coordinates(meuse.grid) = ~x+y
proj4string(meuse.grid) <- CRS("+init=epsg:28992")
```

```
[using ordinary kriging]
```



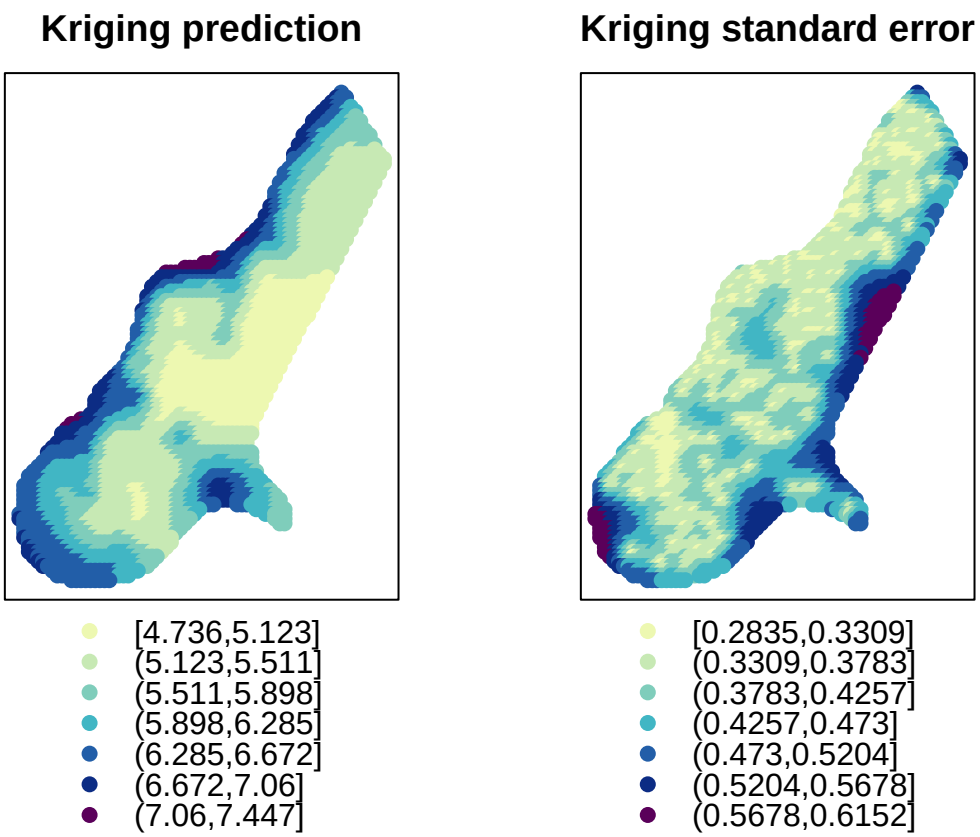


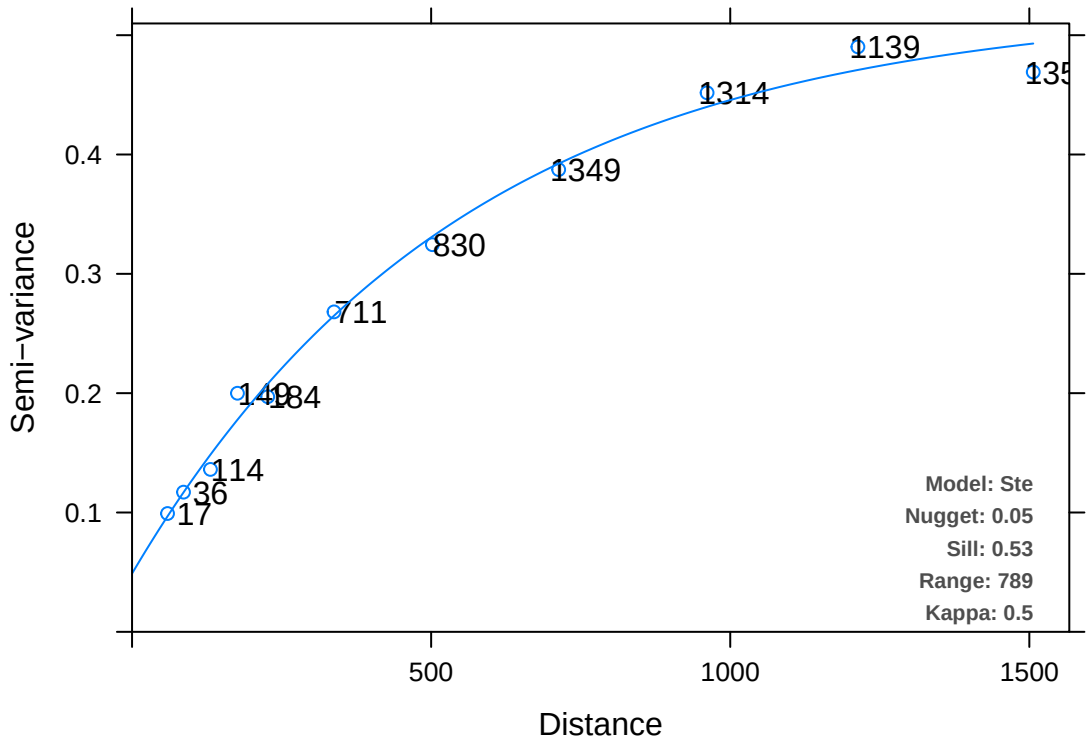
1.4.1 Automatic Universal kriging

Since there is a relationship between zinc and distance to a specific orientation (i.e. the river), then it would be better to use universal kriging. This appears to be the case for the dataset, so universal kriging may be a better option due to its more lenient anisotropy condition. Another difference between the ordinary kriging example

above the universal kriging is that universal kriging takes into account non-stationary conditions, which uses polynomials instead of a linear combination to determine weighting system.

The kriging process can also be automated using the `automap` package. As seen below, the output above can be simplified into a few lines of code. As seen in the output, it summarizes the distance pairs, determines the size the bins, autofits the variogram, and generates plots for the kriging prediction and standard errors.


Kriging prediction
Kriging standard error
[4.736,5.123]
(5.123,5.511]
(5.511,5.898]
(5.898,6.285]
(6.285,6.672]
(6.672,7.06]
(7.06,7.447]
[0.2835,0.3309]
(0.3309,0.3783]
(0.3783,0.4257]
(0.4257,0.473]
(0.473,0.5204]
(0.5204,0.5678]
(0.5678,0.6152]


Experimental variogram and fitted variogram model
Semi−variance
0.1
0.2
0.3
0.4
Distance
500
1000
1500
17
36
114
149
184
711
830
1349
1314
1139
Model: Ste
Nugget: 0.05
Sill: 0.53
Range: 789
Kappa: 0.5

Chapter 2

Exploring the Kaggle San Francisco Crime Dataset

In the first chapter, I explored kriging as a concept for my dataset using the `meuse` dataset. In the following two chapters, I conduct exploratory data analysis and apply the kriging technique to my dataset. First, I use standard packages for analyzing the dataset. Then, I change the dataset so that it is a data structure that uses `gstat` and `sp`, two common packages for spatial data analysis. I describe the process that I used to complete exploratory data analysis and have also included a Shiny application that aided me in presentation and further exploratory analyses. I have integrated explanations of the interactive features in the Shiny application into explanations below. A link to the Shiny application is available by clicking here.

2.1 Introduction

As mentioned earlier, I am using the dataset that I used in the final group project in STAT-495. It is a dataset from Kaggle that examines crime in San Francisco. The data has information about the crime timestamp, category of the crime incident, description of the crime, day of the week, name of the Police Department District, resolution, address, longitude, and latitude.

In addition, the Kaggle competition has test, training, submission datasets. The goal of the project was to predict the outcome of the crime category on the even-numbered weeks when given a training set of the odd-numbered weeks. There are 878,049 observations in the training set and 884,262 observations in the test and submission set. My group compared and contrasted different machine learning methods and transformed existing variables for our models. At the end of the course, we gave a

20-minute presentation to demonstrate our findings along with an executive summary that can be found by clicking on the link here. Please note that you must be signed into an Amherst College account to view the link to the summary.

My comprehensive project, as an extension, will be examining the spatial component. I continue the data exploration that was done earlier and look for a way to apply ordinary kriging with an indicator variable. I am also only using the training set, since it is the only complete dataset. This means that my data only includes the odd-numbered weeks.

2.1.1 Shiny App

As incorporated into the explanations of the various stages of exploratory data analysis, the Shiny application can be useful for comparing different groupings and visualizations of the data. It can be directly accessed by following this link.

To summarize the Shiny application features that were discussed through the section, it contains the following interactive features:

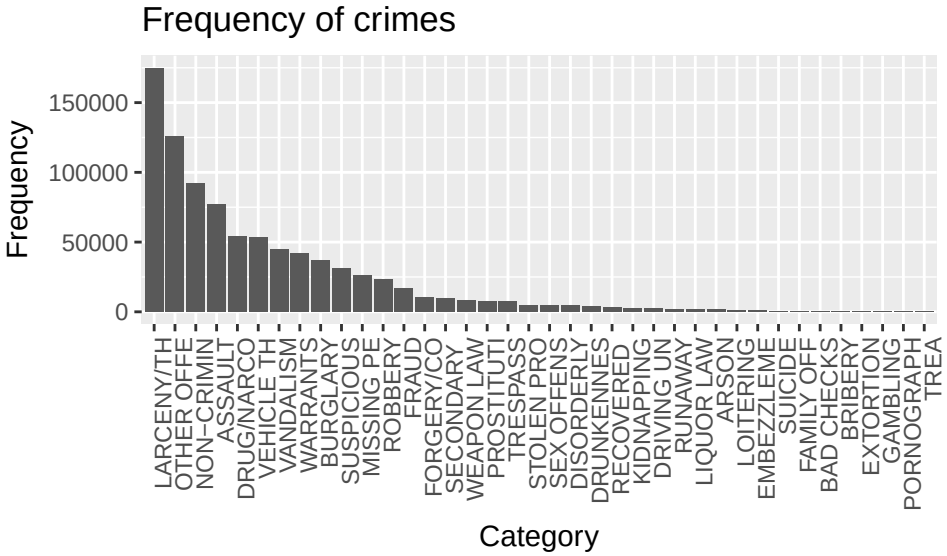
- `leaflet` plot of points
- Density plot
- Histograms of crime count by police district
- Data viewer
- Variogram and variogram cloud
- Summary of spatial dataset

2.2 Crime Exploratory Analysis

Kriging is typically used on a quantitative outcome, but the Kaggle San Francisco crime dataset is based on categorical outcomes. To apply kriging to my dataset, I needed to fit my dataset to the constraints of kriging. Because of the scope of the project, I was interested in applying indicator kriging to predict two categories rather than the 39 categories listed in the dataset.

In order to determine on which variable to krige as an indicator outcome, I explore potential categories: Larceny/theft, other offenses, non-criminal, violent crime, and

property crime. As seen in the plot below, larceny/theft, other offenses, and non-criminal offenses are the most common crimes that occur in San Francisco. The categories in the violent crime category uses the Federal Bureau of Investigation's (FBI) definition of violent crime: It includes assault and robbery. The other two categories in the FBI's definition, murder and rape, were not labeled as categories in the Kaggle San Francisco Crime dataset. Similarly, the FBI's definition of property crime was used to group larceny/theft, vehicle theft, burglary, and arson together.



In the Shiny application, I consider these five potential indicators through an interactive application. In the sidebar, each of those five categories are included. Within the spatial plots navigation panel section, there are four tabs: an interactive map, a density plot, a histogram of crime by police district, and an interactive dataset table explorer.

2.3 Logistic Regression

Unlike kriging, a logistic regression does not take into the spatial association that observations may share with each other in a manner that is as rigorous as kriging. Nevertheless, conducting a logistic regression analysis can provide a quick analysis before considering the additional steps required with kriging (i.e. fitting a variogram, creating a grid, interpolating onto a grid).

As seen with the table below, location seems to play a signficant role in predicting crime. At an alpha-level of 0.05, there does not appear to be any significant variables for predicting property crime in a specific orientation. This is a good indicator that the isotropy condition will be satisfied when we examine the variogram by orientation.

	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1	(Intercept)	-916.27	532.48	-1.72	0.0853	-1965.47	127.60
2	X	-5.64	3.94	-1.43	0.1528	-13.35	2.14
3	Y	5.97	4.07	1.47	0.1427	-1.91	14.09

2.4 Exploratory Analysis using Spatial Packages

In my exploratory analysis of determining the indicator variable on which to krige, I examined variograms of the data to determine if there was an association of location and crime. In order to do this, I used the `sp` and `gstat` packages to convert and plot the spatial dataset. There is a sidebar that allows the use to select each of the five categories to compare the datasets. Within the variogram plots navigation panel section, there are three tabs: a variogram, a variogram cloud, and a summary of the dataset. Please note that this is a smaller subset of the data due to the lengthy amount of time for R to plot a variogram cloud. When there are 400 observations, a variogram cloud must plot the distance between every point combination.

Please refer to the Appendix for the full code for the conversion from a `data.frame` to `SpatialPointsDataFrame` for the Kaggle San Francisco Crime dataset. I have included the converted dataset in that code. As seen in these coordinates and visualized in the above plots, there seems to be crime occurring throughout San Francisco, and more of it is concentrated in the northeast corner of the city. There is crime occuring else in the city as well, but it does not seem to be as concentrated as the other area.

2.4.1 Distribution of Property Crime

This can also generally be seen in the Shiny application. However, this plot is included to show where property crime occurs in the sample used for kriging. Please refer to Figure 2.1.

2.4.2 The Summary Table

The summary table provides information on our projected dataset. The coordinates have been converted from latitude-longitude coordinates to UTC (northings-eastings) coordinates. This was done so that the Kaggle San Francisco Crime dataset would be consistent with the prediction grid that had its coordinates retrieved from an online source that used UTC coordinates. More information is provided in the next chapter. Also, please note that there are no packages for converting the output from an object

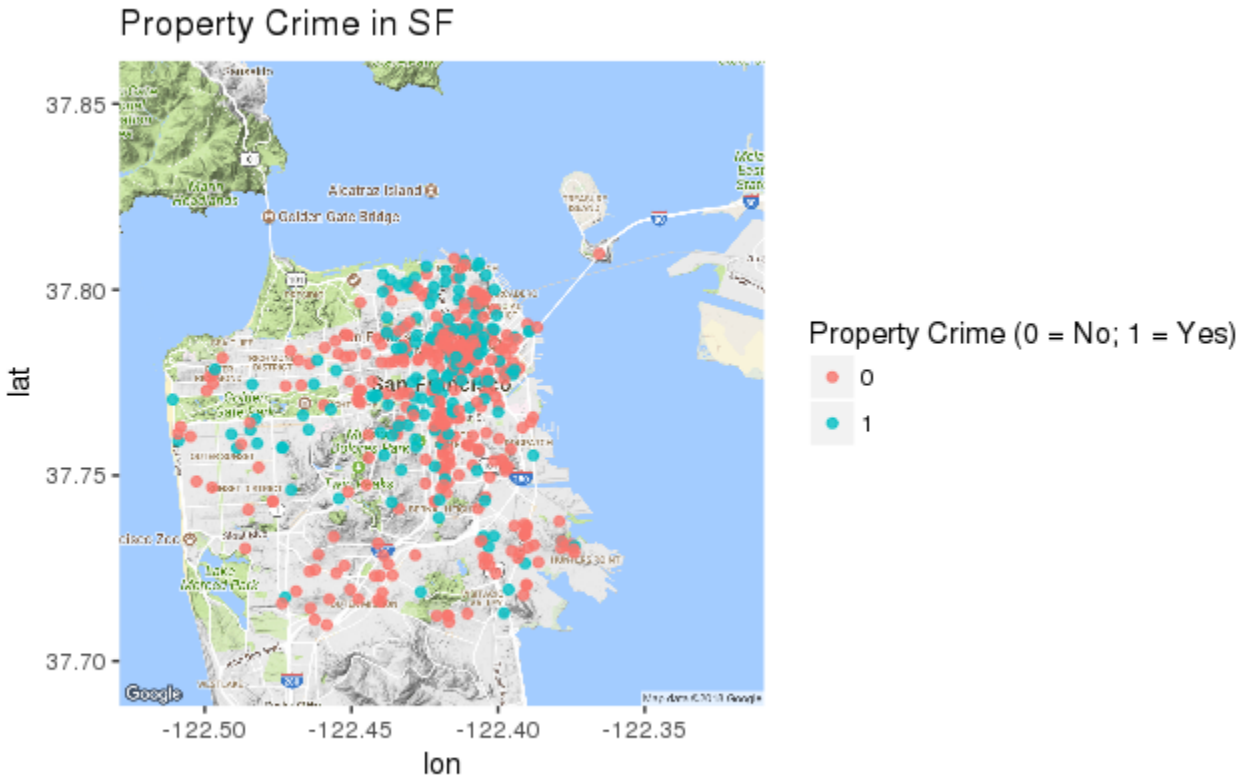


Figure 2.1: Property Crime in San Francisco.

of class `SpatialPointsDataFrame` to a LaTeX table. Packages, such as `xtable` and `kable` do not process the data in the same as a standard table.

Property crimes occur throughout San Francisco, as seen in the summary table. When I multiply the mean by the sample size, there seems to about 150 instances of property crime in San Francisco out of the 500 instances of crime.

```
# Summary statistics
summary(crime_spatial_df_new)
```

```
Object of class SpatialPointsDataFrame
Coordinates:
      min     max
X  543680  555243
Y 4173681 4184724
Is projected: TRUE
proj4string :
[+init=epsg:32610 +proj=utm +zone=10 +datum=WGS84 +units=m +no_defs
+ellps=WGS84 +towgs84=0,0,0]
Number of points: 500
Data attributes:
 PropertyCrime         X                Y
 Min.   :0.000   Min.   :543680   Min.   :4173681
 1st Qu.:0.000   1st Qu.:549933   1st Qu.:4178077
 Median :0.000   Median :551467   Median :4181066
 Mean   :0.314   Mean   :550878   Mean   :4180040
 3rd Qu.:1.000   3rd Qu.:552189   3rd Qu.:4182036
 Max.   :1.000   Max.   :555243   Max.   :4184724
```

Chapter 3

Applying Kriging to San Francisco Crime Data

3.1 Introduction

In this section, I cover the application of kriging to the Kaggle San Francisco Crime dataset. Similar to the `meuse` dataset, I will follow the general outline of exploring the data, creating a variogram, fitting the variogram, and creating predictions. However, there are some key differences in the application of the dataset.

The San Francisco crime dataset from Kaggle was structured to predict crime category, which was divided into 39 potential crime categories. For consistency in the classification of property crime, I follow the Federal Bureau of Investigation's (FBI) classification of property crime: larceny/theft, burglary, arson, and vehicle theft. For kriging, it is also necessary to create a prediction grid for the data. It is also possible to create prediction points for the missing data. I decided to split the data on property and non-property crime.

Furthermore, while the `meuse` dataset had a grid for prediction loaded into the `gstat` package, I needed to create a grid for the prediction area using the San Francisco Open Data Initiative website's available data.

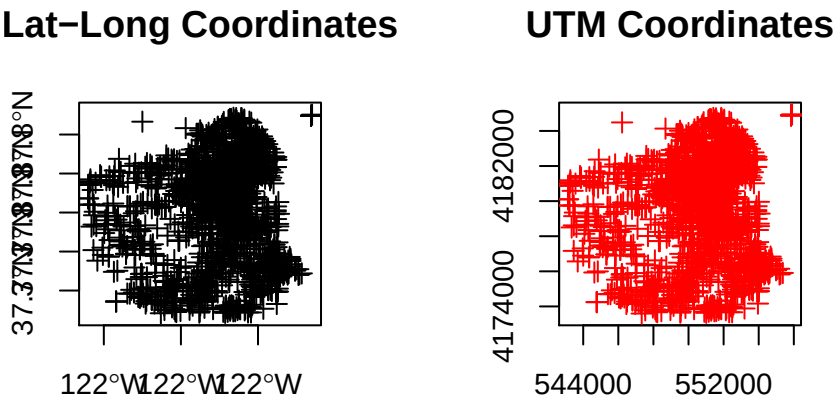
For kriging, I do the following steps as seen in the structure of chapters 2 and 3:

1. I create a variogram and determine an appropriate fit.
2. I make a prediction grid based on the organization of San Francisco.
3. I make predictions using kriging for the grid of San Francisco.
4. I synthesize the results and draw conclusions about the data.

3.2 Data Input

After loading in the Kaggle San Francisco Crime dataset, I sample 2000 observations from the dataset. Kriging can be computationally intensive, so limiting the size of the dataset can be useful for ensuring that the analyses occur in a reasonable amount of time. For instance, calculating the variogram cloud takes 0.5 * 1000 * 999 = 499,500 distance calculations for the variogram cloud.

After importing the dataset, it was necessary to convert the dataset into a `SpatialPointsDataFrame` in order for the use of spatial packages, such as `gstat`. I converted the coodinates from longitude and latitude to UTM coordinates. In the plot below, I have illustrated the conversion process: Although the units along the x and y axes have changed, the locations of the crime observations have not changed.

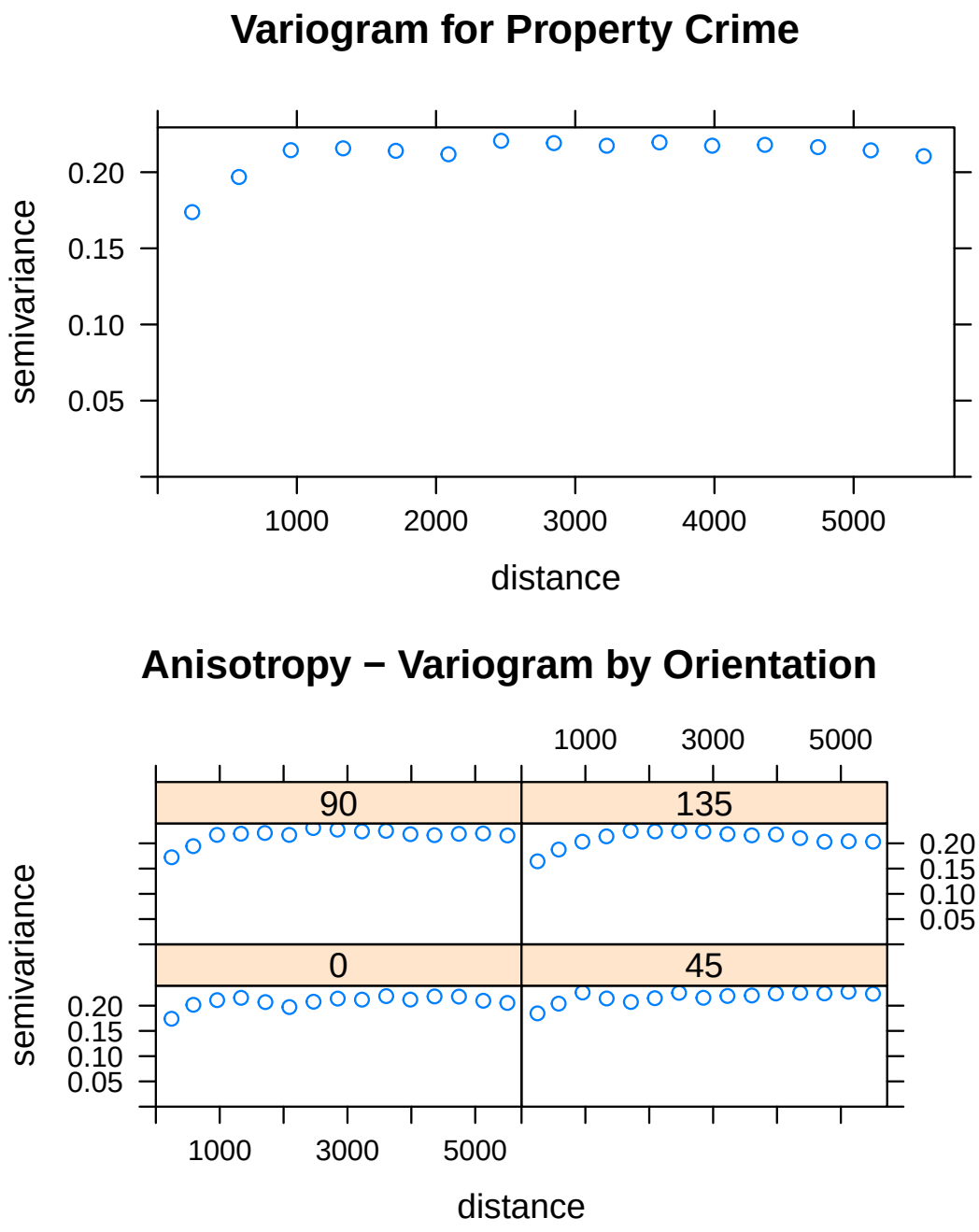


3.3 Variogram

Similar to the `meuse` dataset, I created a variogram cloud, variogram, and variograms by various orientations. The variogram cloud, as described in the first chapter, shows all the possible pairings' semivariance as a function of the distance. The variogram bins the data together. Finally, the variograms for the different orientations shows if property crime is dependent on just orientation or distance.

3.3.1 Examining the variogram

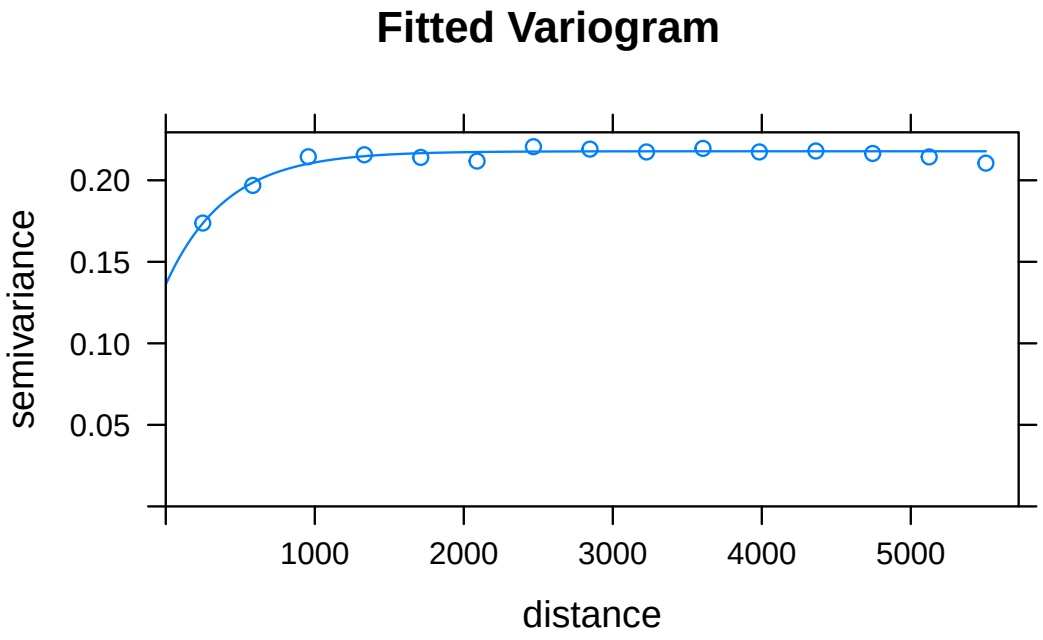
With the variogram, it shows that the stationarity condition is met, because the variogram levels off. As distance increases, the spatial correlation does not increase to infinity. Furthermore, the variograms for the different orientations show that orientation of the pairings is not a significant factor, thus satisfying the isotropy condition.



3.3.2 Fitting a variogram

Next, because I select the nuggest, sill, and range for the fitted variogram. I also specify the model as being an "Ste" model (Matern model), though it is possible change the model to spherical, Gaussian, exponential, etc. There are other options that may work better, but for my analyses I stay with the "Ste" model since it fits well with my data. Further work could compare various models more formally in addition to quick judgments of fit from changing model shapes.

```
nugget <- 0.16
psill <- 0.07
range <- 1000

vmodel <- fit.variogram(vgm, model = vgm(
  model = "Ste",
  nugget = nugget,
  psill = psill,
  range = range
))

plot(vgm, model = vmodel, main="Fitted Variogram")
```



3.3.3 Creating the Grid

The `meuse` dataset had a preloaded grid. However, for the San Francisco crime dataset, I needed to create a grid from externa data (DataSF, 2018). In the San Francisco Open Data Initiative archives, I found a relevant file of a map of San Francisco in a ".csv" format and used online software to convert to a shapefile (".shp"). I uploaded the map into R from the San Francisco Open Data, which involved importing a ".shp" file along with its associated files that are related to it. Afterwards, I specified the dimensions necessary in order a grid that could be used in as a spatial data type. Below is a plot of the prediction grid made from the dataset.

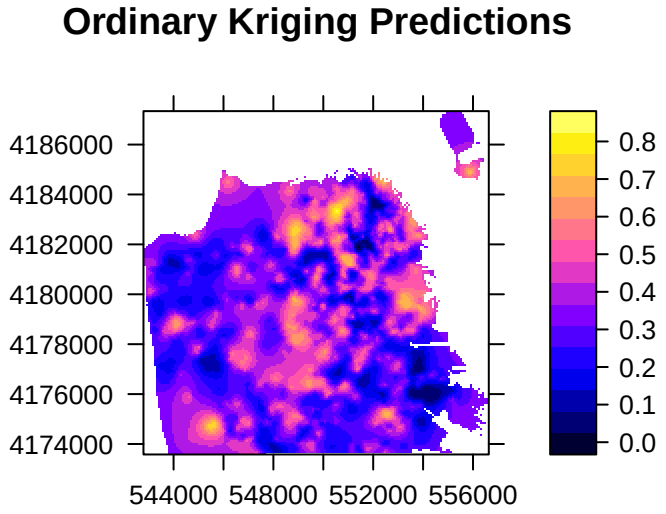
3.4 Kriging

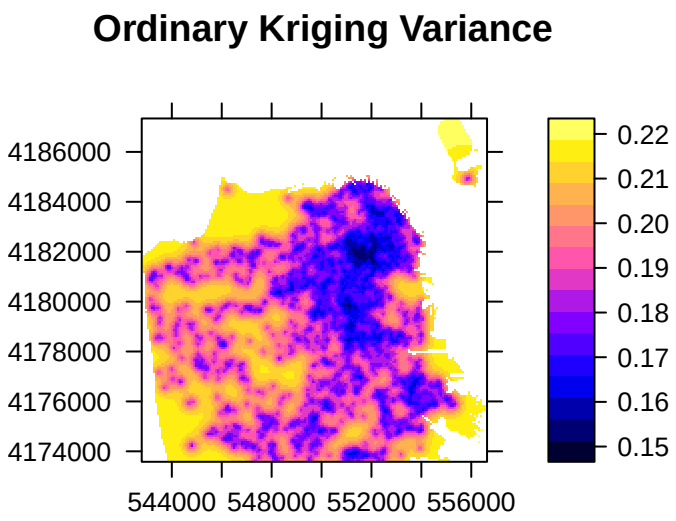
3.4.1 Ordinary Kriging

In the following section, we see that ordinary kriging can be applied to the dataset. As mentioned above, the conditions of stationarity and isotropy are satisfied, which allows us to use ordinary kriging.

As seen on the grid predictions, the northeast corner of San Francisco, wihch is also the downtown area, has some areas that are predicted to be high in property crime. Other areas are lower in crime, but there are some smaller high likelihood areas in the northwest and southwest. While the outer areas are predicted to be a smaller probability of property crime, the variance is also higher for the outer areas.

```
km <- krige(PropertyCrime ~ 1, remove.duplicates(crime_spatial_df_new),
            newdata = spgrid, model = vmodel)
```

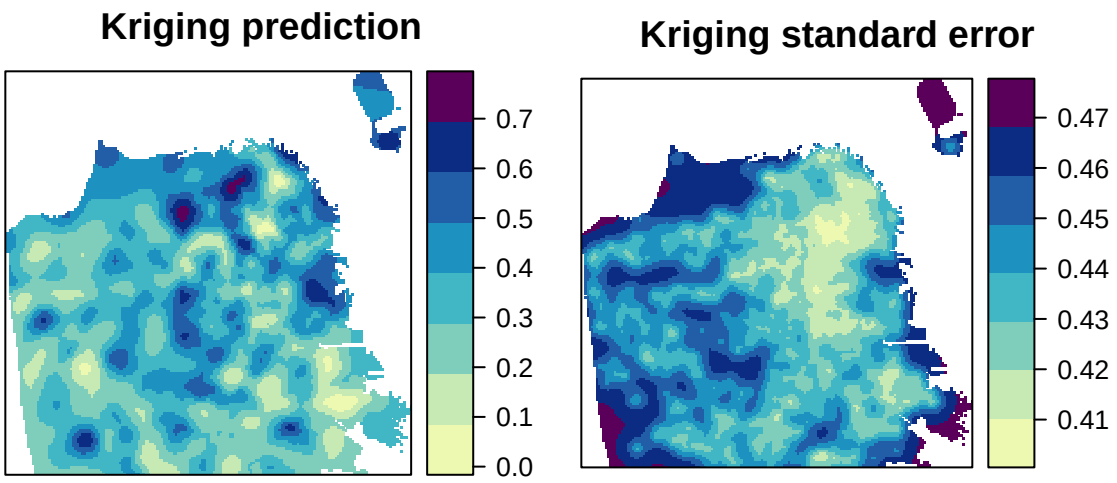


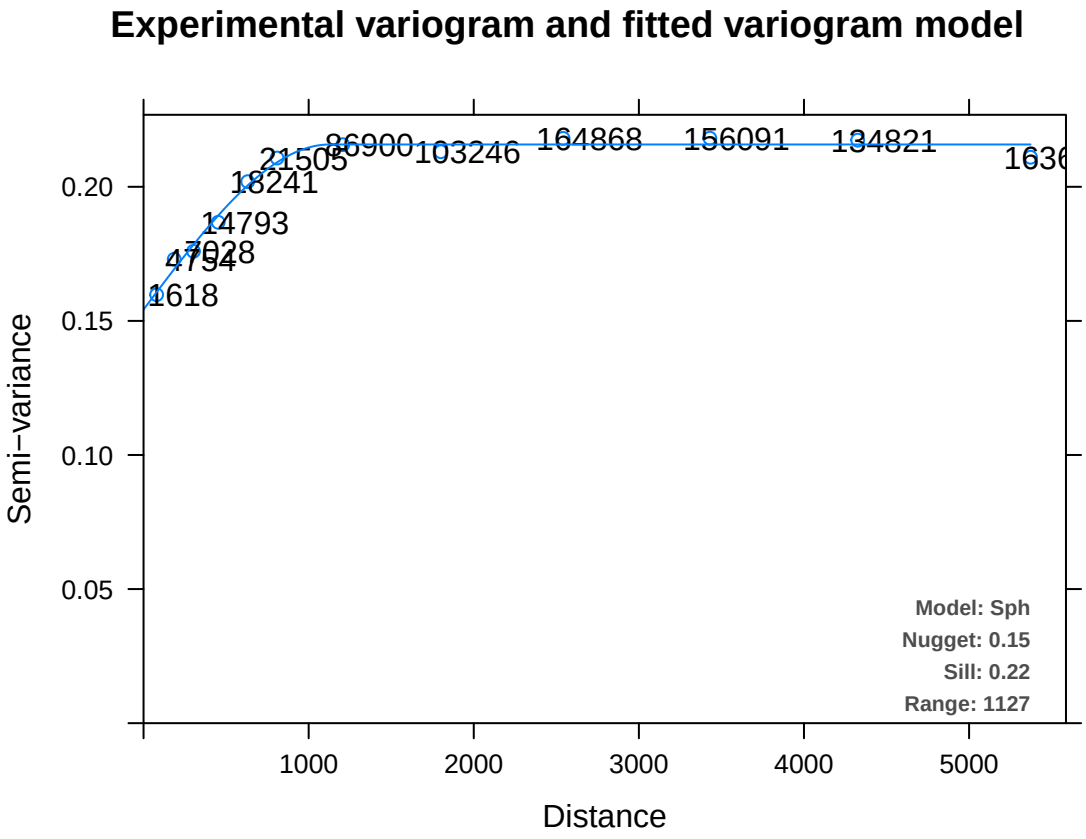


3.4.2 Automatic Universal Kriging

As seen earlier, the kriging process can also be automated (i.e. automatically fit a variogram and create the plots) with the `automap` package. The package automatically uses universal kriging, because the isotropy condition is lessened with universal kriging (and is a more common but more a complicated theoretically kriging technique).

When examining the indicator outcome of property crime, I will use ordinary kriging to examine my results. Due to coding convenience, I used the `autoKrige()` function for a universal kriging application. While I have less of a mathematical understanding of universal kriging because it's more complicated than ordinary kriging, the conclusions are the same given that my data already satisfied the assumptions for ordinary kriging about stationarity and isotropy.


Kriging prediction
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Kriging standard error
0.47
0.46
0.45
0.44
0.43
0.42
0.41


Experimental variogram and fitted variogram model
Semi−variance
0.20
0.15
0.10
0.05
1618
4754
7028
14793
18241
21505
86900
103246
164868
156091
134821
1000
2000
3000
4000
5000
Distance
Model: Sph
Nugget: 0.15
Sill: 0.22
Range: 1127

3.5 Drawing Conclusions

According to my ordinary kriging grid predictions, there are some distinct patterns that appear. Property crime tends to be more likely in the northeast and central regions of San Francisco. There are more pockets of high property crime probability than in other areas, thought the northeast region seems to have more hubs. The northeast region looks particularly unlikely to have property crime. In the southwest and southeast regions, there is usually one or two "hubs" of crime in one area. There seems to be some in the central area of San Francisco as well. It is also important to note that variance is lower in the northeast region versus the other areas of San Francisco.

The kriging grids, unfortunately, may be difficult to read due to the lack of labeling of neighborhoods and landmarks. As a a result, I decided to plot some predictions onto a Google Maps version of San Francisco. I took the 300 most probable locations and plotted them on a `ggmap` that gives better detail into the specific areas in which crime occurs.

This plot further confirms the study findings that property crime has the highest probability of occuring in northeast San Francisco with many crime clusters in that area. Furthermore, there seems to be property crime in many areas close to parks more generally in San Francisco: Lake Merced Park, Twins Peaks, Mission Dolores, North Beach, and Presidio/UCSF. I wouldn't be surprised if a lot of vehicle theft occurs in those areas due to the high number of cars that may be in the area that could be potential victims of car theft.

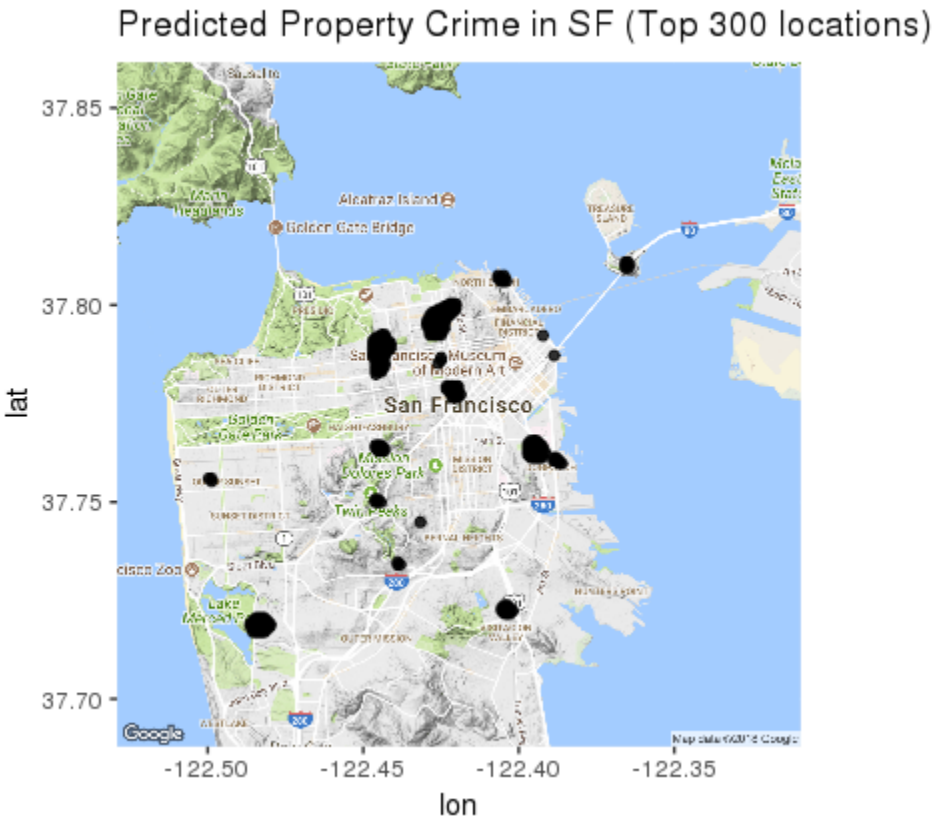


Figure 3.1: Predicted Property Crime in San Francisco.

Chapter 4

Conclusion

4.1 Project Conclusion

The intention of the comprehensive project was to provide an exposition on kriging, conduct exploratory data analysis on the Kaggle San Francisco Crime dataset, and apply the kriging technique to the crime dataset. As seen in the exploratory analysis and the kriging application to the Kaggle San Francisco Crime dataset, property crime tends to occur in the northeastern part of San Francisco. This corresponds with the specific clustered areas in San Francisco. Furthermore, we see that more crime generally occurs in the northeast area and around parks more generally in San Francisco. A potential suggestion would be to increase policing of park areas, such as Lake Merced Park, Presidio, and the Twin Peaks, in order to keep property crimes lower in those areas. In these non-northeast San Francisco area, there is typically a lower probablility of property crime, which would be areas that one should choose to live if they want to reduce the likelihood of being a victim of property crime.

4.2 Future Directions

If this project were continued, there would still be plenty of research that would be possible. While I used the Kaggle dataset that only had the odd weeks, it would be potentially more precise to use the dataset that is on the San Francisco Open Data Initiative's website (DataSF, 2018). In addition, though computationally intensive to do given the time constraints of one kriging method, it would be interesting to have an interactive kriging Shiny application to examine multiple trends and categories. Additional next steps might include exploring Area-to-Area Kriging or Area-to-Point

Kriging, which was done in another paper that examined car theft in the Baltic States (Kerry, Goovaerts, Haining, & Ceccato, 2010). By continuing the use of spatial statistics for crime data, it may help us understand crime patterns better and make the world a safer place.

Appendix A

The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readibility and/or setup.

In the main Rmd file:

```
# This chunk ensures that the acstats package is
# installed and loaded. This acstats package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(acstats)){
  library(devtools)
  devtools::install_github("Amherst-Statistics/acstats")
}
library(acstats)
```

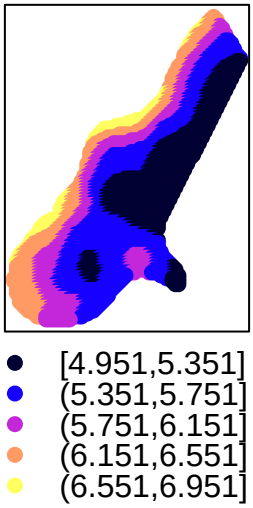
In Chapter 1:

```
# Geostatistical packages
library(sp)
library(gstat)
library(texreg)
library(knitr)
library(xtable)

# Loading meuse dataset
data(meuse)
#names(meuse)
```

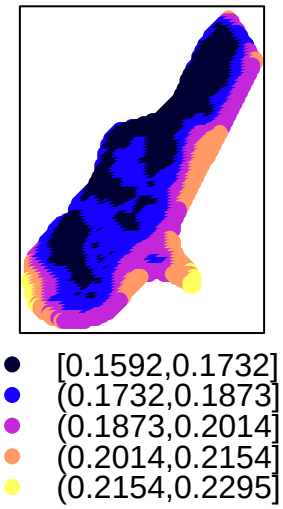
```
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS("+init=epsg:28992")
class(meuse)
```

```
# Linear Regression
model_trend <- lm(log(zinc) ~ x + y, data = as.data.frame(meuse))

# Summary
smod <- data.frame(xtable(summary(model_trend)))
```

```
# Do kriging predictions over the grid
grid_km <- krige(log(zinc) ~ x + y, meuse, newdata = meuse.grid, model = vmodel)
names(grid_km)

spplot(grid_km, "var1.pred")
```




```
spplot(grid_km, "var1.var")
```

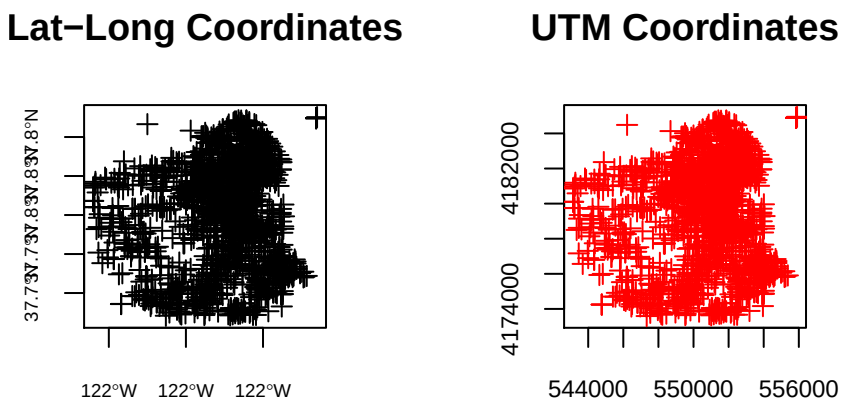


```
library(automap)
zinc_auto <- autoKrige(log(zinc) ~ x + y,
                       input_data = meuse,
                       new_data = meuse.grid)
```

In Chapter 2:

```
load(file = "data.rda")
library(tidyverse)
library(stringr)
library(dplyr)
library(ggplot2)
library(knitr)
library(broom)
library(xtable)

options(xtable.floating = FALSE)
options(xtable.timestamp = "")
set.seed(79)
options(digits = 3)

knitr::opts_chunk$set(
  echo = TRUE, fig.width=5, fig.height=3, message=FALSE, warning = FALSE
  )
```

```
library(sp); library(gstat); library(raster)
# Tell R these are lat/longitude points
coords <- SpatialPoints(crime_df[,c("X", "Y")], proj4string = CRS("+proj=longlat +
crime_spatial_df <- SpatialPointsDataFrame(coords, crime_df)
proj4string(crime_spatial_df) <- CRS("+proj=longlat +datum=WGS84")

# Transform into UTM - Verification
cord.dec <- SpatialPoints(crime_df[,c("X", "Y")], proj4string =CRS("+proj=longlat'
cord.UTM <- spTransform(cord.dec, CRS("+init=epsg:32610"))
par(mfrow = c(1, 2))
plot(cord.dec, axes = TRUE, main = "Lat-Long Coordinates", cex.axis = 0.60)
plot(cord.UTM, axes = TRUE, main = "UTM Coordinates", col = "red", cex.axis = 0.75
```




```
par(mfrow = c(1,1))


# Transform into UTM
crime_spatial_df <- SpatialPointsDataFrame(cord.dec, crime_df)
crime_df_new <- cbind(crime_df[c("PropertyCrime")], cord.UTM@coords)
crime_spatial_df <- SpatialPointsDataFrame(cord.dec, crime_df_new)
crime_spatial_df_new <- spTransform(crime_spatial_df,
  CRS("+init=epsg:32610 +proj=utm +zone=10 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +
```

```
library(ggmap)
map <- get_map(location="sanfrancisco",zoom=12,source="google")
#load("map.Rda")

ggmap(map) +
  ggtitle("Property Crime in SF") +
  geom_point(
    data=as.data.frame(crime_df),
    aes(x=X, y=Y, color=as.factor(PropertyCrime)),
    show.legend=TRUE,
    alpha=0.8
    ) + labs(color = "Property Crime (0 = No; 1 = Yes)")
```

In Chapter 3:

```
# This chunk ensures that the acstats package is
# installed and loaded. This acstats package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(dplyr))
```

```
    install.packages("dplyr", repos = "http://cran.rstudio.com")
if(!require(ggplot2))
    install.packages("ggplot2", repos = "http://cran.rstudio.com")
if(!require(acstats)){
  library(devtools)
  devtools::install_github("Amherst-Statistics/acstats")
  }
library(acstats)

knitr::opts_chunk$set(
  echo = TRUE, fig.width=5, fig.height=3, message=FALSE, warning = FALSE
  )
```

```
# Transform into UTM
crime_spatial_df <- SpatialPointsDataFrame(cord.dec, crime_df)
crime_df_new <- cbind(crime_df[c("PropertyCrime")], cord.UTM@coords)
crime_spatial_df <- SpatialPointsDataFrame(cord.dec, crime_df_new)
crime_spatial_df_new <- spTransform(crime_spatial_df, CRS("+init=epsg:32610 +proj=

# Summary statistics
plot(crime_spatial_df_new)
```

```
summary(crime_spatial_df_new)
print(crime_spatial_df_new)
```

```
vcloud <- variogram(PropertyCrime ~ 1, data = remove.duplicates(crime_spatial_df_n
#plot(vcloud, main = "Variogram Cloud for Property Crime")
```

```
# Data Import
data <- rgdal::readOGR("planning_neighborhoods.shp")

# Data Manipulation
proj4string(data) <- "+init=epsg:32610 +proj=utm +zone=10 +datum=WGS84 +units=m +n

# Creating the Grid
geo_bounds <- data
```

```
#plot(geo_bounds); points(crime_spatial_df_new)
bbox(geo_bounds)

grid <- GridTopology(c(542713, 4173500), c(80, 80), c(250, 250))
```

```
# Create points with the same coordinate system as the boundary
gridpoints <- SpatialPoints(grid, proj4string = CRS(projection(geo_bounds)))
plot(gridpoints)
```

```
# Crop out the points outside the boundary
cropped_gridpoints <- crop(gridpoints, geo_bounds, gridpoints)
plot(cropped_gridpoints)
```

```
ph_auto_grid <- autoKrige(PropertyCrime ~ X + Y, input_data = remove.duplicates(crime_s
#save(ph_auto_grid, file = "ph_auto_grid.Rda")
```

```
km_df <- data.frame(km)
km_df_top <- km_df %>%
  dplyr::arrange(desc(var1.pred))
km_df_top <- km_df_top[1:300, ]

km.UTM <- SpatialPoints(km_df_top[,c("X", "Y")], proj4string =CRS("+init=epsg:32610 +pr
km.dec <- spTransform(km.UTM, CRS("+proj=longlat +datum=WGS84"))
km_df_top <- cbind(km.dec@coords, km_df_top[3:4])

km_top_map <- ggmap(map) +
  ggtitle("Predicted Property Crime in SF (Top 300 locations)") +
  geom_point(
    data=as.data.frame(km_df_top),
```

```
    aes(x=X, y=Y),
    show.legend=TRUE,
    alpha=0.8
    ) + labs(color = "Property Crime (0 = No; 1 = Yes)")

#save(km_top_map, file="km_top_map.Rda")
load("km_top_map.Rda")
km_top_map
```

References

Bivand, R. S., Pebesma, E. J., & Gomez-Rubio, V. (2008). *Applied spatial data analysis with r*. New York, NY: Springer.

Bohling, G. (2005, october). C&PE 940. Retrieved from `http://people.ku.edu/~gbohling/cpe940/`

Data Camp. (2018, february). Spatial statistics in r. Retrieved from `https://www.datacamp.com/courses/spatial-statistics-in-r`

DataSF. (2018). Data sf. Retrieved from `https://datasf.org/opendata/`

Heuvelink, G. (2015). Geostatistics for soil mapping. Retrieved from `https://www.youtube.com/watch?v=FWmADoAbXNg&list=PLAh8kwUozOdceDXrMIjpASva2QNWjX7o3`

Kerry, R., Goovaerts, P., Haining, R. P., & Ceccato, V. (2010). Applying geostatistical analysis to crime data: Car-related thefts in the baltic states. *Geographical Analysis*, 53–77.

Pebesma, E., & Graeler, B. (2017). Package 'gstat'. Retrieved from `https://cran.r-project.org/web/packages/gstat/gstat.pdf`

Pebesma, E., Bivand, R., Rowlingson, B., Gomez-Rubio, V., Hijmans, R., Sumner, M., ... O'Rourke, J. (2017). Package 'sp'. Retrieved from `https://www.youtube.com/watch?v=FWmADoAbXNg&list=PLAh8kwUozOdceDXrMIjpASva2QNWjX7o3`