Cpln HW 4: Point Pattern Analysis

Ajjit Narayanan

Bill Cohen

## Introduction

Ensuring access to healthy, locally grown food has been a challenge for many American cities. As one of the ways to get fresh and healthy food to residents of the different Philadelphia neighborhoods, the Philadelphia Food Trust has set up numerous farmers markets throughout the city. These markets have been shown to offer numerous benefits to consumers who shop there and to their communities. That said, not all neighborhoods in Philadelphia have access to farmers markets; in fact, parts of South Philadelphia and North Philadelphia, and essentially all of Northeast Philadelphia have no farmers markets at all, depriving people who live there and their communities of the numerous benefits listed above.

In this study, we look at the distribution of farmers markets in Philadelphia in 2013 and 2016, using data provided by the City of Philadelphia and accessed through the PA Spatial Data Access website (Pennsylvania Spatial Data Access). We present several point pattern analyses to assess the degree to which the markets are randomly distributed, clustered, or dispersed. A Nearest Neighbor Analysis and K-Function Analysis are conducted using both ArcMap and R, and the result are compared.

## Methods

A point process is a series of events, incidents, or locations of interest with a spatial, and often temporal, distribution across a study area. In this paper, the point process is the location of farmers markets in Philadelphia in 2013 and 2016. Through point pattern analyses, we seek to characterize this spatial distribution to identify potential geographic disparities in access to local, healthy foods. The spatial distribution can be characterized as random, as defined below, clustered, with groups of points located near one another, or dispersed, with points uniformly located away from one another.

In our point pattern analyses we will test the hypothesis that the spatial distribution of points has complete spatial randomness (CSR). For a study region that is divided into sub regions, a point process is said to be completely spatially random if it meets two conditions: 1) The probability of a point occurring in any sub region is directly proportional to the area of the sub region. 2) The location of a point is completely independent from the location of any other point.

If is divided into square cells of equal size, the sub regions are called quadrats. Assessing the point distribution using this type of subdivision is called the Quadrat Method, and offers a simple way of analyzing point density. However, the results of this method are severely limited because they depend entirely on the size of the quadrat area chosen, a type of modifiable areal unit problem (MAUP). Patterns may exist in region at scales smaller or larger than can be detected using a given cell size. Additionally, the Quadrat Method does not take into account distance between points or the type of clustering that may exist. Because of these limitations, we do not use the Quadrat Method is this study. We present two newer methods of point pattern analysis that can overcome some of these limitations.

### NNI

The Nearest Neighbor Index (NNI) seeks to characterize spatial randomness by comparing the observed average distance, , between each point and its nearest neighbor with the expected average distance, , to each nearest neighbor under conditions of complete spatial randomness.

or

The observed average distance is determined by summing the distance between each point and its nearest neighbor and dividing by the total number of points : . The expected average distance under CSR is given by: where is the area of the minimum enclosing rectangle.

The point pattern is said to be random when NNI is close to 1, meaning the observed average distance approximates the expected CSR average distance. An NNI value close to 0 indicates clustering, and a value close to 2 indicates dispersion, with an upper limit of 2.149.

A significance test is used to determine whether or not the NNI indicates randomness. The null hypothesis, , is that the observed point pattern is random, and therefore the NNI is significantly close to 1. The alternative hypothesis, , is that the observed point pattern is not random, and therefore the NNI is significantly close to 0 (indicating clustering) or 2 (indicating dispersion). The test statistic used has a standard normal z-distribution, , where SE is an estimate of the standard error of observed average distances for points within a minimum enclosing rectangle of area given by .

If z < |1.96|, we fail to reject , indicating a spatially random pattern. If z > 1.96, we reject in favor of , indicating significant dispersion. If z < -1.96, we reject in favor of , indicating significant clustering.

The Nearest Neighbor Index offers a global measure of point distribution across a study region that does not suffer from the same resolution and MAUP issues as the Quadrat Method. However there are still limitations with NNI analysis. As seen in the equations above, the area of the minimum enclosing rectangle factors heavily in the results. Just a single outlier located far from otherwise clustered points can greatly affect the analysis as the minimum enclosing rectangle expands greatly. The shape and size of the actual study region (in our case Philadelphia) is also omitted from the analysis. In this study, much of the irregularly shaped Philadelphia region is not included. We adjust the area in ArcMap to reflect the area of the Philadelphia region and improve the analysis, but the irregular shape is not considered. Additionally, as a global measure, NNI does not account for patterns that vary with scale.

### K-function Analysis

In order to account for the variation of spatial patterns at different scales, K-function analysis uses local neighborhoods ranging in size. Here, a local neighborhood is defined for each point using a circle with radius . The average number of other points, or neighbors, in each neighborhood, standardized by the point density of the study region, provides a measure of the type of spatial pattern at distance . By evaluating the K-function using many neighborhood sizes, we can see how point patterns vary at different scales.

where is the number of points in the study region with area , and is the number of neighbors for each point in a neighborhood defined by a circle with radius .

At neighborhood of radius , if , the points exhibit complete spatial randomness. A value of indicates clustering while a value of indicates dispersion. However these thresholds can be more easily interpreted by transforming using the function:

where is the non-negative radius of the neighborhood.

In this case, CSR occurs when , clustering when , and dispersion when at scale .

However the function used by ArcGIS does not deduct the value of from the first term, so and the threshold value for comparison is rather than 0.

Hypothesis testing for K-function analysis uses a permutation based test to generate confidence intervals. The bounds of the confidence interval are defined by the lowest and highest values of L(d) calculated from a user-defined number of randomly generated point patterns for each distance . In this analysis we use 99 random permutations to generate the confidence intervals, or confidence envelopes as we sometimes call them.

As with the Quadrat Method and Nearest Neighbor Index, we test for the null hypothesis that the point pattern exhibits complete spatial randomness. We identify CSR for neighborhood with radius , and fail to reject at distance , when the observed value of L(d), denoted falls within the confidence interval, with the lower envelope (bound) donated by and upper enveloped (bound) denoted by , i.e. < < . Unlike the previous methods however, two alternative hypotheses are tested, one for clustering and one for dispersion. If is greater than the upper envelope , then is rejected in favor of , suggesting significant dispersion. Conversely, if falls below the lower envelope , is rejected in favor of , suggesting significant clustering.

K-function analysis generates neighborhoods that can extend beyond the boundaries of the study region. For points located at distance or closer to a boundary, the circular neighborhood used will necessarily include areas without any neighbors, i.e. outside of the study region. Without correction, this empty area will be incorrectly considered as “no neighbors” rather than “no data” in the analysis.

Two edge corrections are used to compensate for these boundary issues. Ripley’s Edge Correction works with rectangular study areas by weighting points less than distance from a boundary more heavily. For irregular boundaries, as is the case with our study region, the Simulate Outer Boundary Values Edge Correction takes those points within maximum distance from a boundary and artificially mirrors them across the boundary so they are included in the neighborhoods analyzed for real points close to the boundary. We use this second method of edge correction in our analysis because of Philadelphia’s irregular border.

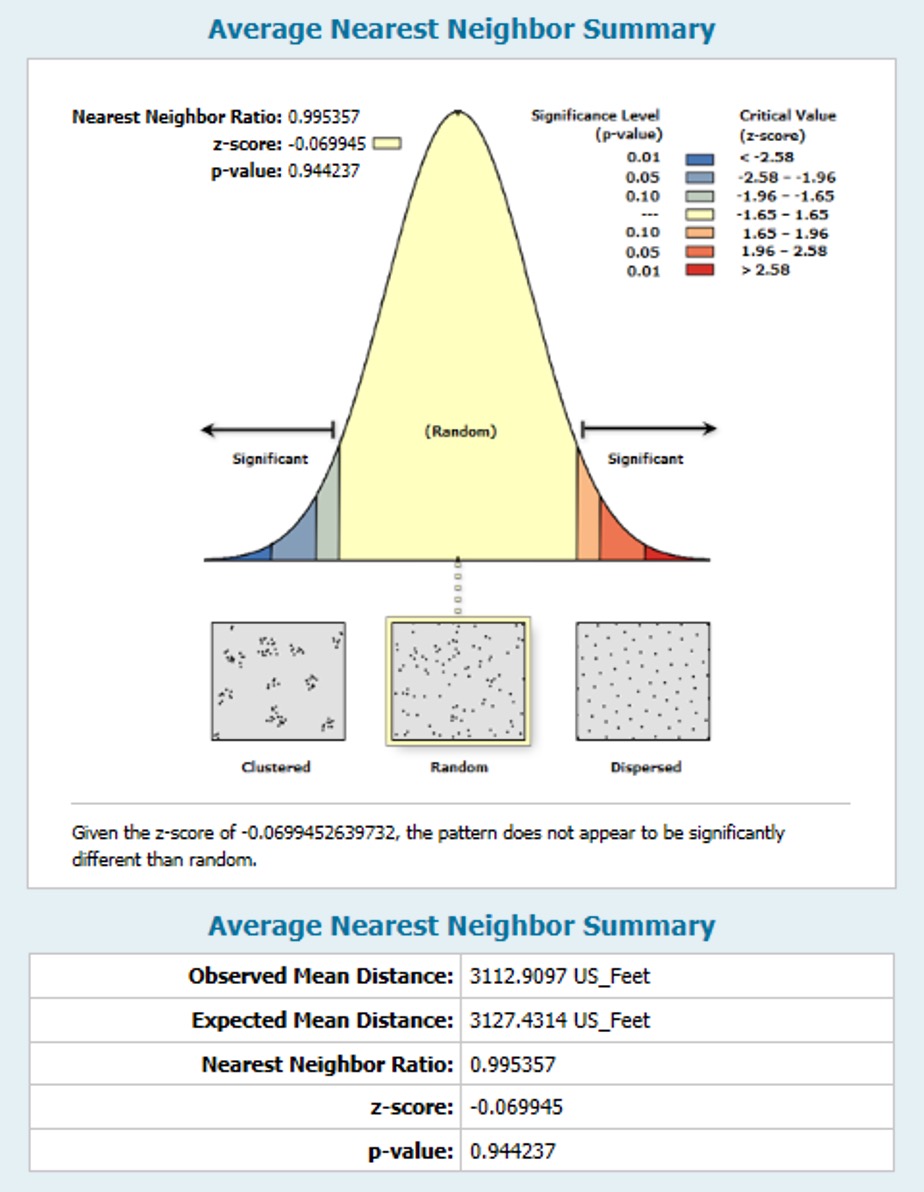
In some (many) cases, it is necessary to consider a reference measure of density, such as population, when assessing point patterns. Most service or retail locations can best be considered in terms of the population density of the area they serve. So an actual random point process may not look exactly random on a map because it has to be weighted by the number of people within each part of the city

To incorporate density of a variable into K-function analysis in ArcMap, a probability raster must be generated, proportional to the variable density to be considered. The Create Spatially Balanced Points tool generates a non homogeneous point pattern according to the probability raster. After repeating this process many times and calculating values of L(d), the highest and lowest values of L(d) are used as the confidence envelope, to be compared with the sample point pattern at each distance band .

## Results

## Nearest Neighbor Index Results

The results from the 2013 NNI summary using the minimum enclosing rectangle in ArcMap and R are below:



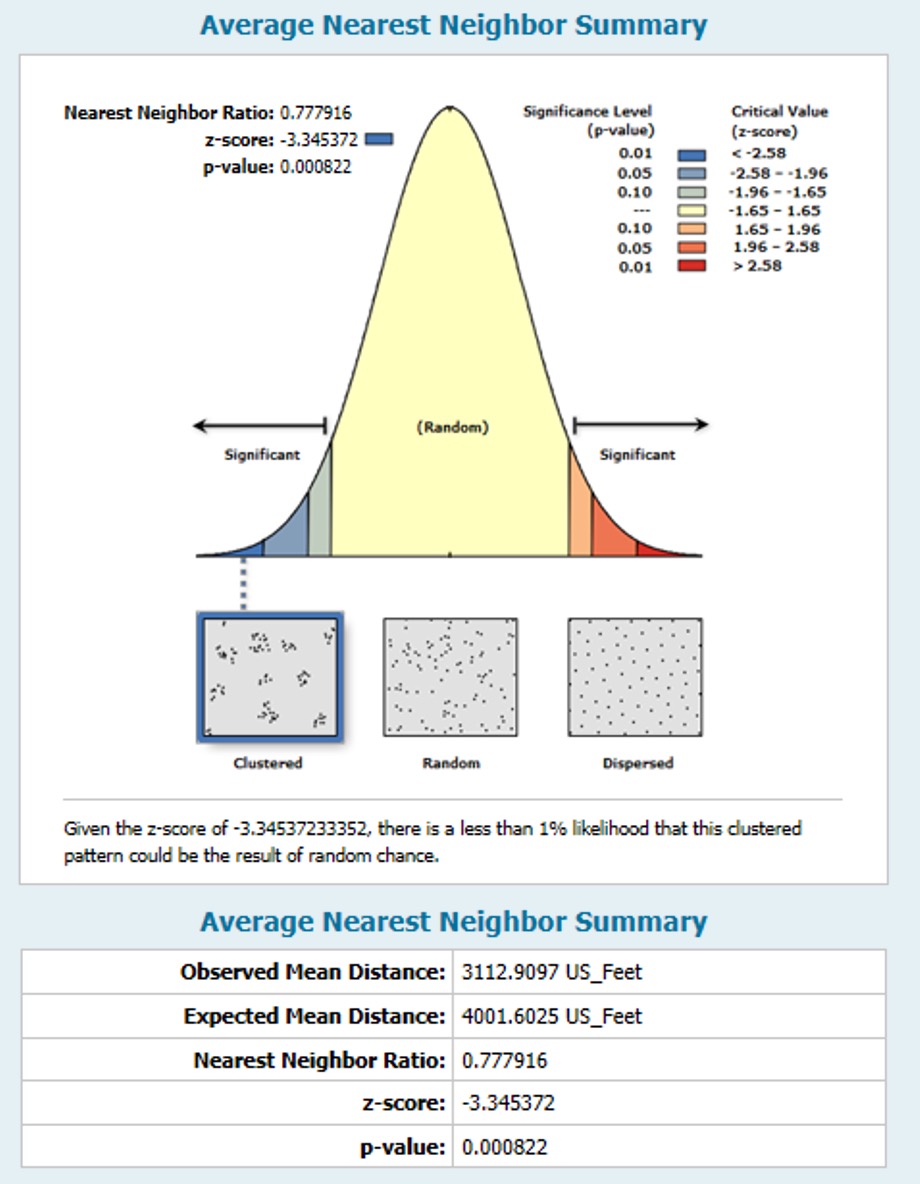
R output:

|  |  |  |
| --- | --- | --- |
| NNI\_Value | p\_value | z\_score |
| 0.9283167 | 0.1401147 | -1.079804 |

First we analyze the ArcMap output. When using the area of the minimum enclosing rectangle, the Nearest Neighbor Index strongly suggests a random pattern. With a z-score of -0.07, we fail to reject , indicating that the observed average distance (3113 ft.) is not significantly different from the expected average distance (3127 ft.), representative of an NNI value very close to 1. However,we can see on the map that there are no farmers markets in the northeast or south of the region. They are mostly clustered in middle of the map. If this point pattern were completely random, as suggested here, we would expect some farmers markets to appear in all areas of the region. Therefore, using the area of the minimum enclosing rectangle is not likely to yield results representative of the Philadelphia region as a whole.

Now looking at the output in R, we see that the NNI is slightly higher at 0.92. The z-score is -1.07 and associated p-value is 0.14, so we still fail to reject of complete spatial randomness. So the ArcMap and R results mostly coincide.

Because of the problems with using the minimum enclosing rectangle, we run the NNI analysis again using the area of the Philadelphia region, 3,971,179,944 sq. ft., as calculated in ArcMap using Zonal Geometry. We also run the analysis in R for comparison.

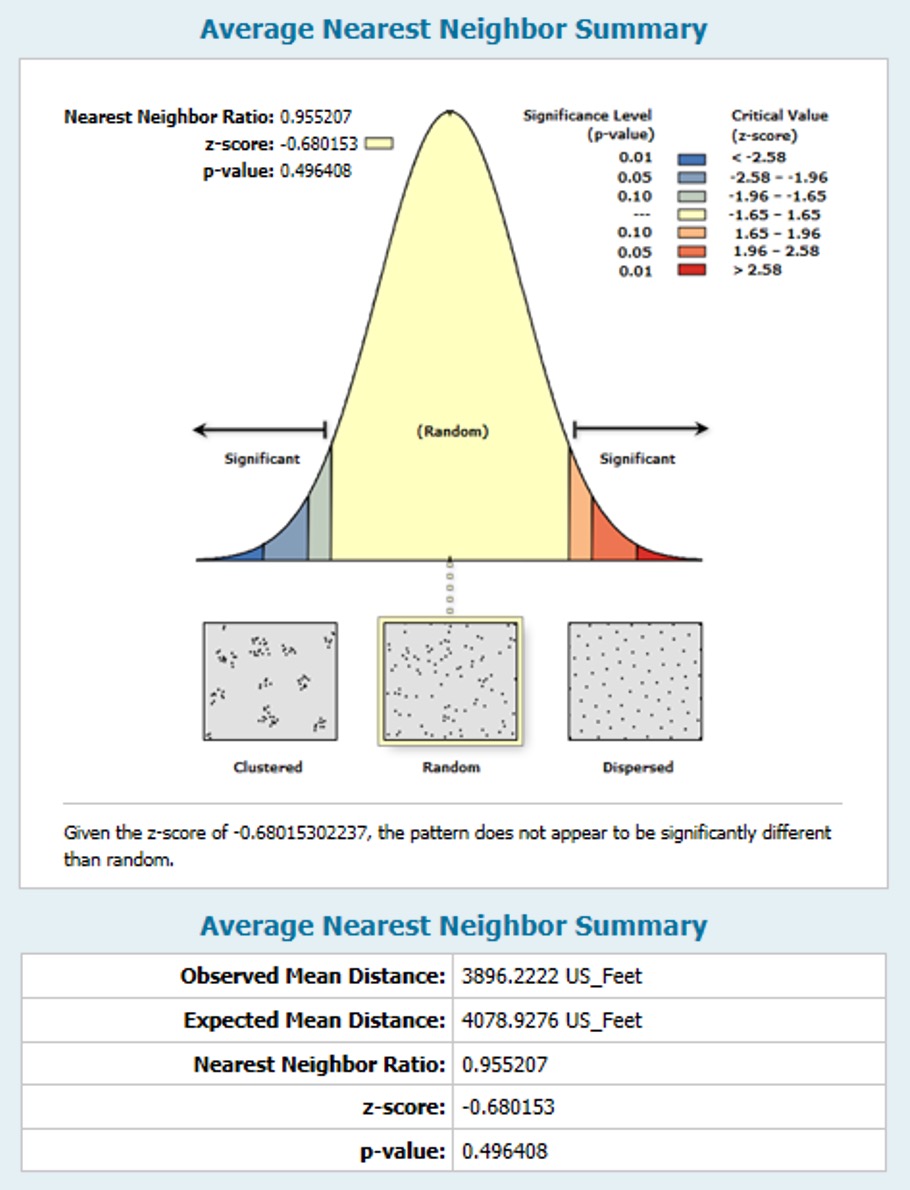


R output:

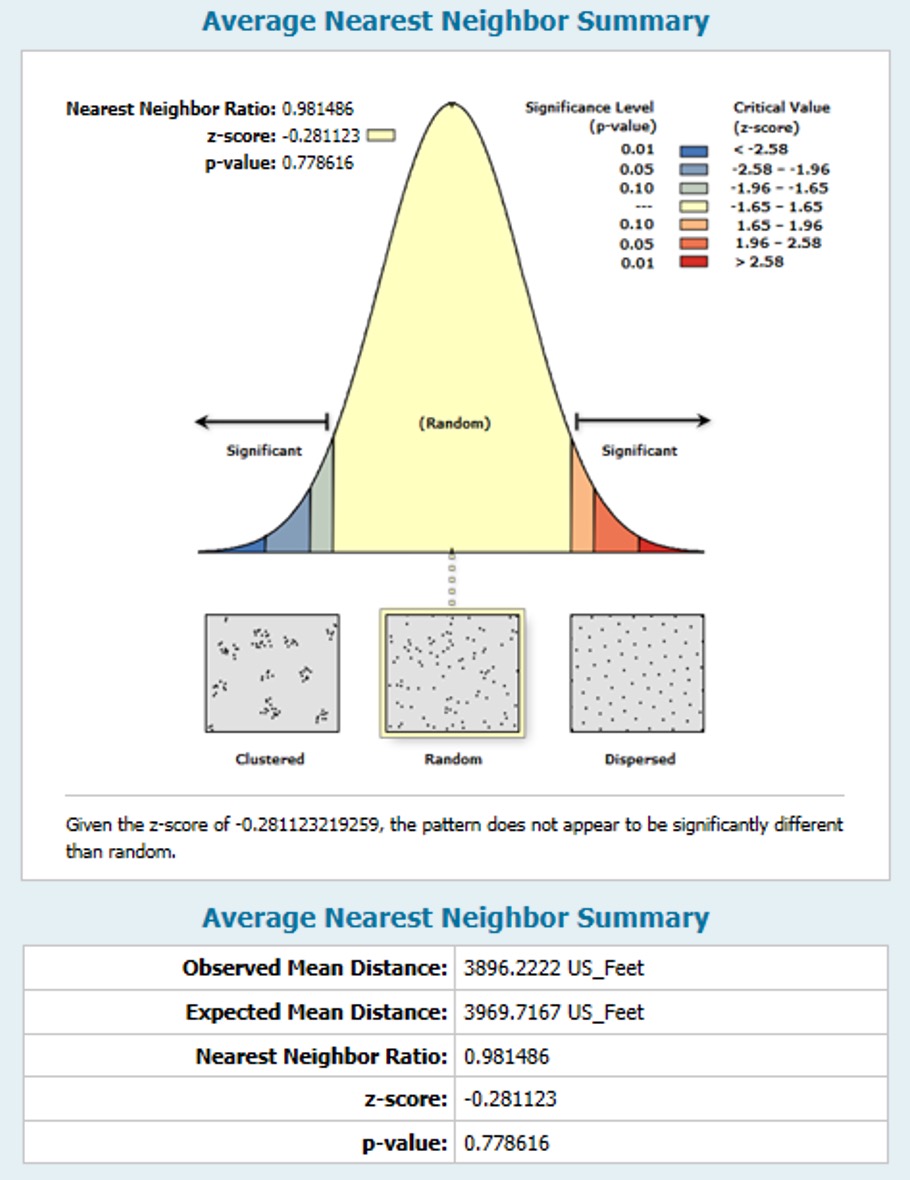
|  |  |  |
| --- | --- | --- |
| NNI\_Value | p\_value | z\_score |
| 0.7779648 | 0.000412 | -3.344634 |

First, we examine the ArcMap output. As expected, using the larger area of the region yields a very different result. Here, the Nearest Neighbor Index strongly suggests a clustered pattern, which is more in line with what we observe on the map. With a z-score of -3.35 (z < -1.96) we reject in favor of , indicating an observed average distance (3113 ft.) that is significantly less than the expected average distance (4002 ft.). The R output corroborates this as the NNI is 0.77 and the z-score is -3.34. So we again reject and see that there is statistically significant clustering.

We also performed Nearest Neighbor Analysis in ArcMap using more recent data from 2016.



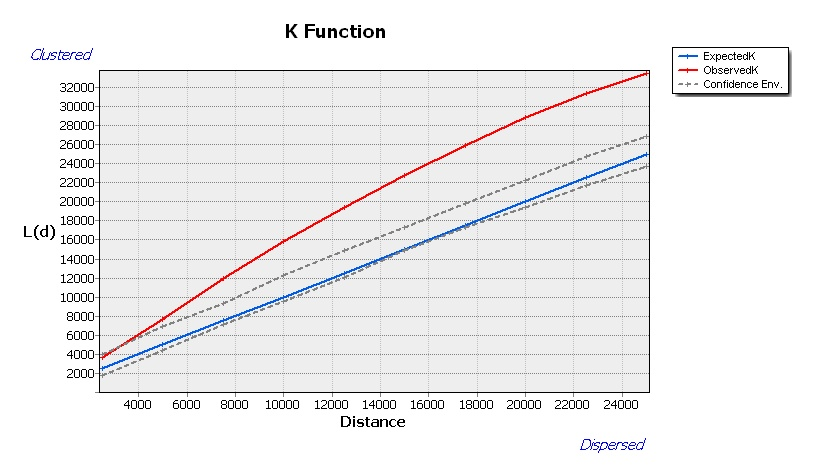
Using the minimum enclosing rectangle in this analysis, we again fail to reject , suggesting a random distribution of farmers markets. The NNI in 2016 is slightly lower (0.96) than it was is 2013 (0.995). When taking the area of the region into account, we see the following results in 2016:



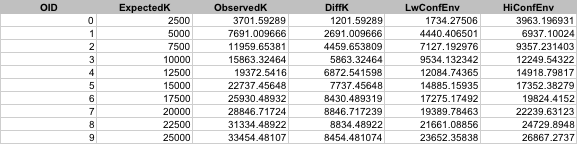
Here we see a very different result in 2016 compared with 2013. In 2016, the pattern appears to be random with a z score of -0.28 and a p-value of 0.78. The addition of two new markets in the northeast are likely the reason for this difference. So, when the size of the region is taken into account, we see that farmers markets tend to be clustered overall in 2013, but the addition of one or more outliers in 2016 introduces significant randomness.

## K-function Analysis

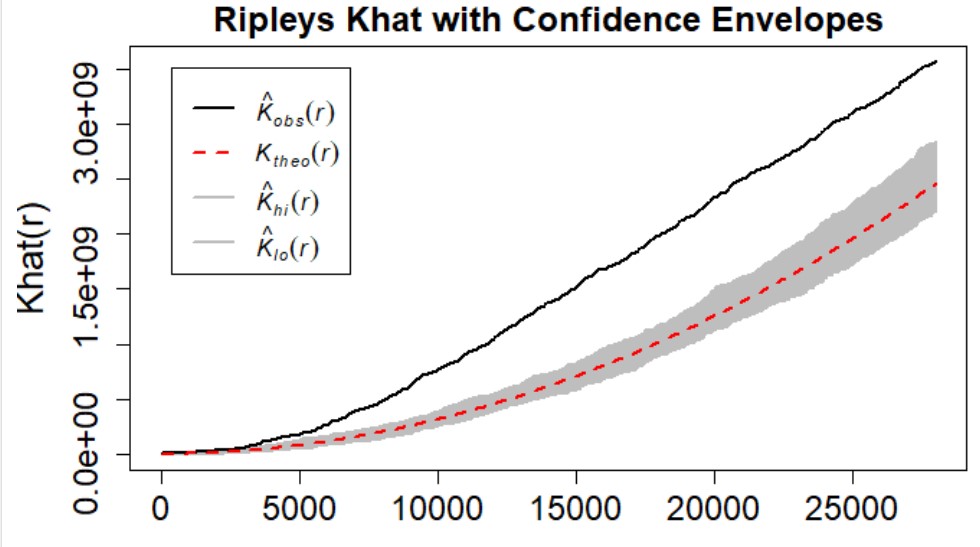
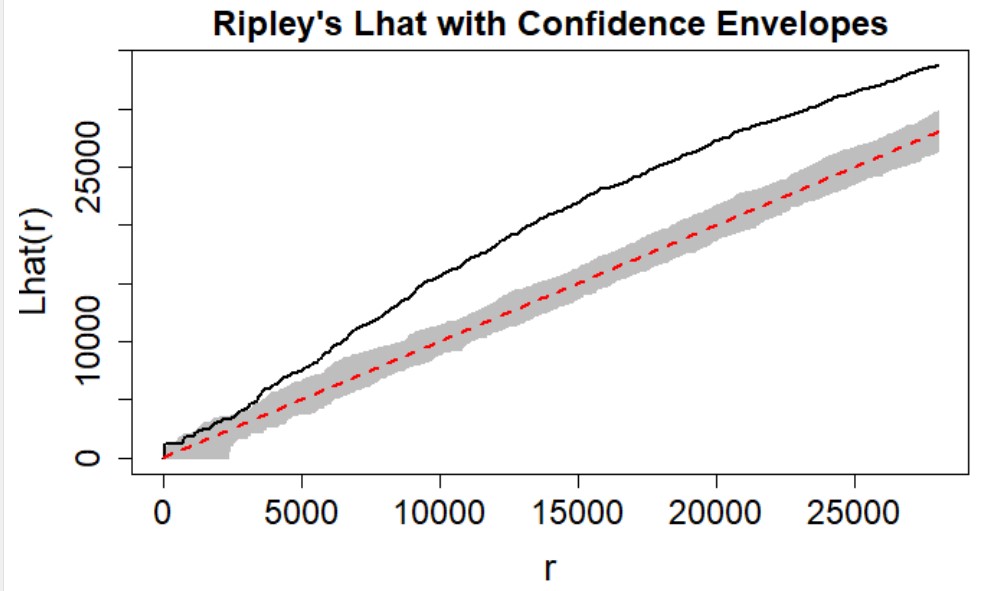
First, we performed a K-function analysis on the location of Philadelphia farmers markets in 2013 using 10 distance bands of radius where increases by 2,500 ft. for each consecutive band. 99 permutations of randomly generated point patterns were used to construct the confidence envelope shown by the gray checked lines in the graph below.



We see that except for very small distances, observed values of L(d) exceed both the expected values under CSR and the upper envelope. This indicates that for distances greater than a few hundred feet, we can reject in favor of , suggesting significant clustering with 99% confidence. This is reflected in the associated table of values at each distance :

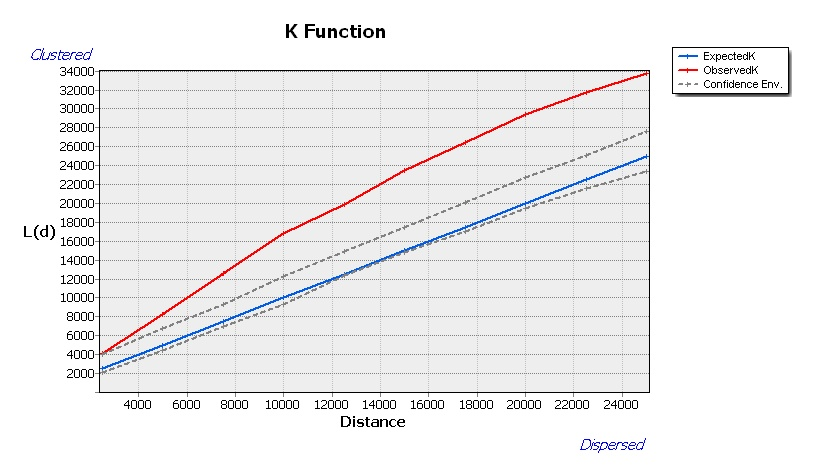


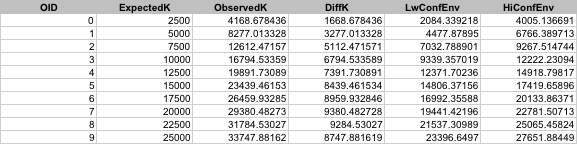
We also replicate the results in R. Below are plots of the k functions and L functions with their associated confidence intervals in gray.

 `

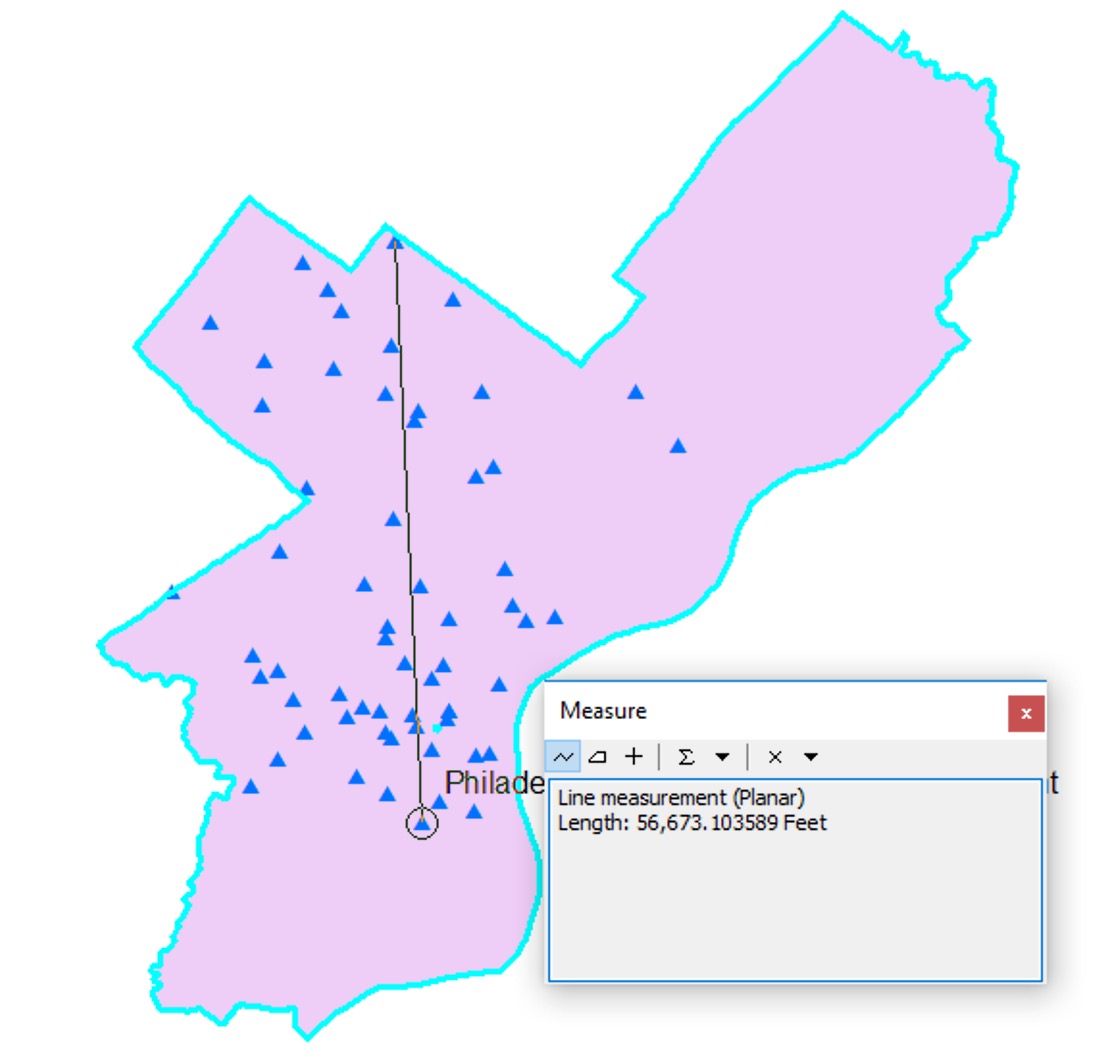
These plots look very similar to the ArcMap output and corroborate our conclusion that for distances greater than a few hundred feet, there is significant clustering.

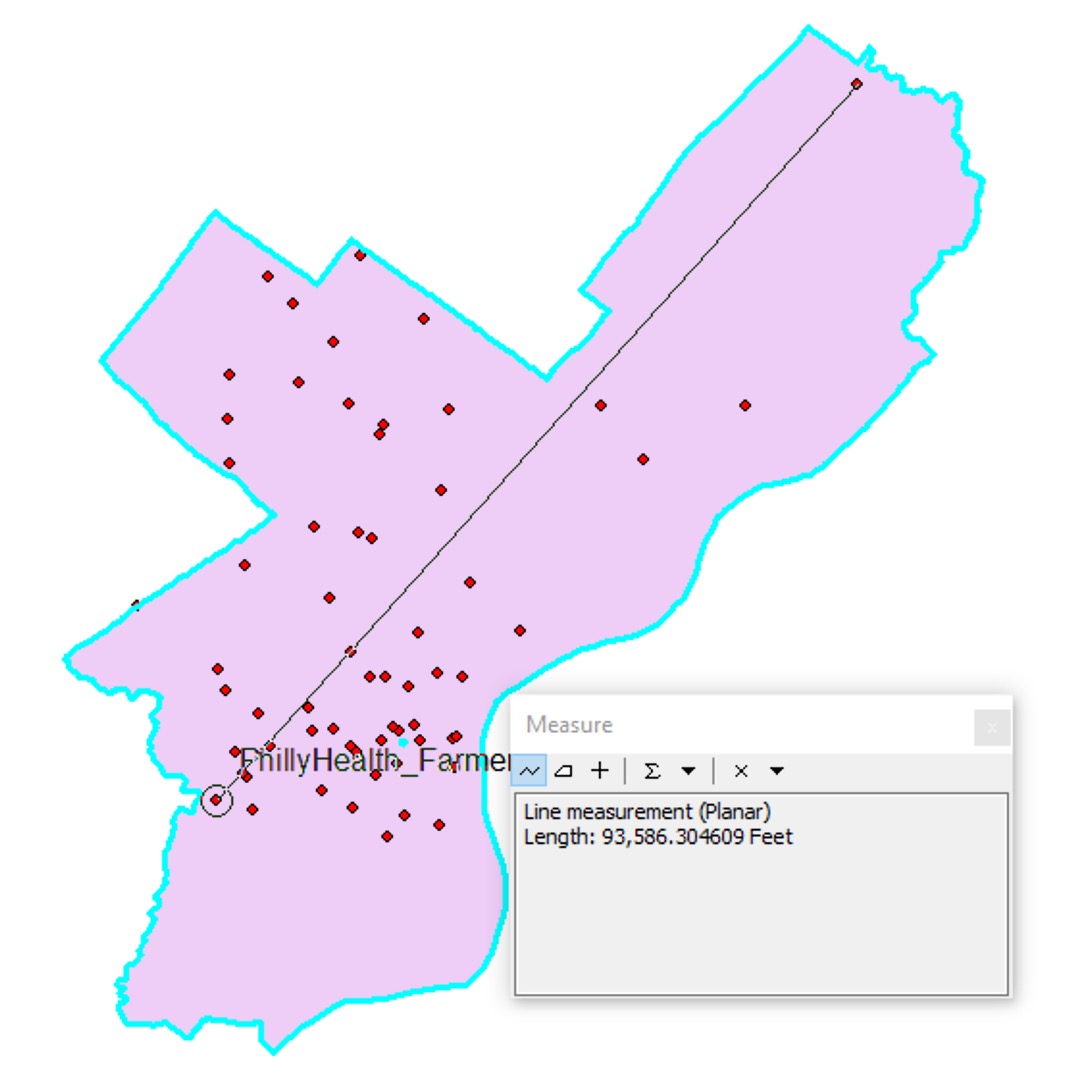
We compare these results to the 2016 data, using the same parameters for neighborhoods and confidence envelope. We find that despite the addition of a market near the northeastern border of the study region, the analysis still indicates significant clustering for almost all distances:





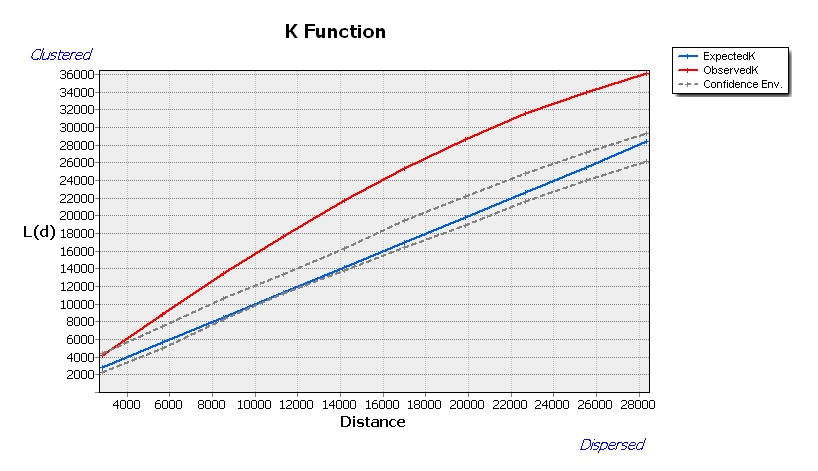
Typically the largest distance between two points in the study region is used to calculate the increments used for neighborhood size. The maximum euclidean pairwise distance between two markets in 2013 was approximately 56,700 ft., while in 2016 it was approximately 93,500 ft., as measured in ArcMap:





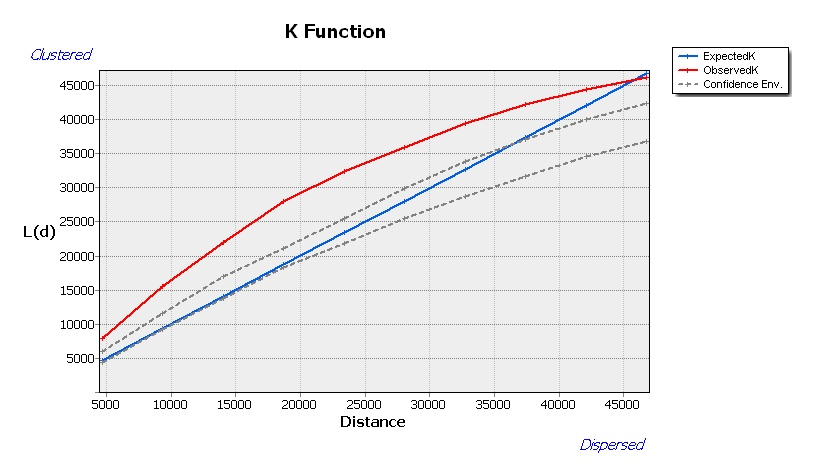
We ran the K-function analysis again using distance band increments customized for each data set (half of the maximum pairwise distance divided by the number of distance bands, in this case 10).

In 2013, the small difference in increment size (2,835 - 2,500 = 335 ft.) does not noticeably impact the results.



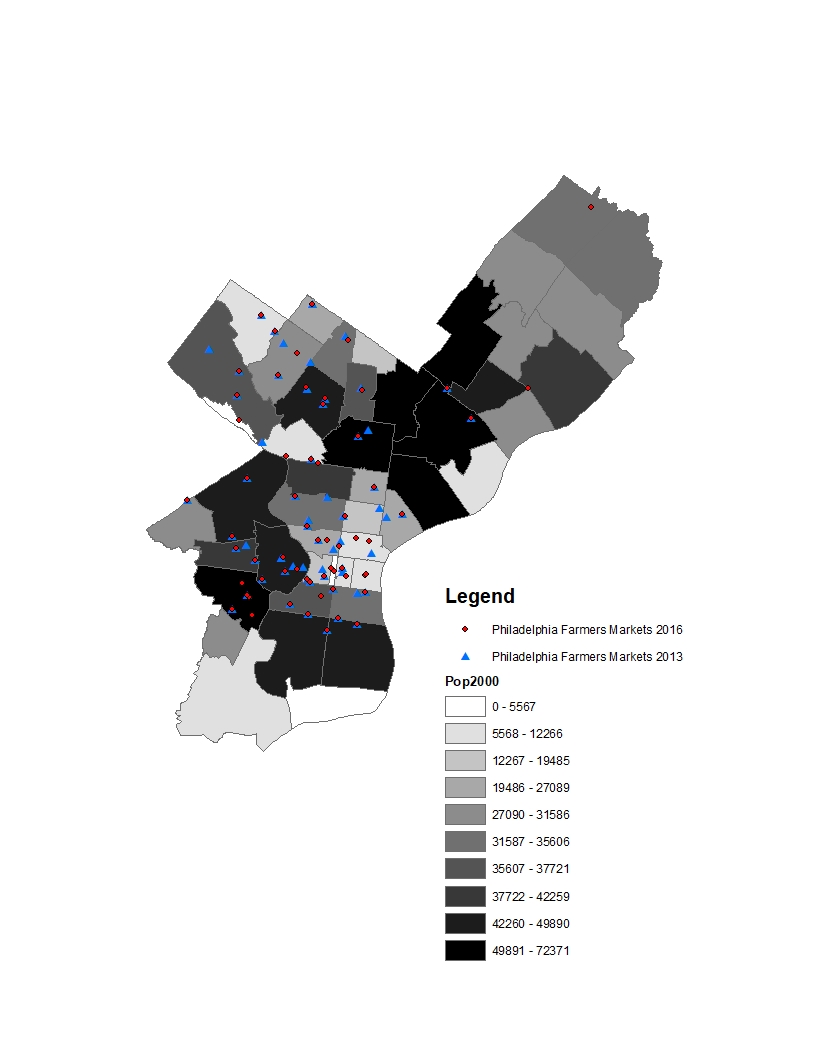


In 2016, the difference in the increment size when calculated using the maximum pairwise distance is much larger than the previous analysis (4,675 - 2,500 = 2,175 ft.). The analysis now predicts some randomness at the largest distances. In the previous analysis, the 10 increments of 2,500 ft. had a maximum neighborhood radius of 25,000 ft. and therefore did include the most distant points, 93,500 ft. apart, in some of the analysis.





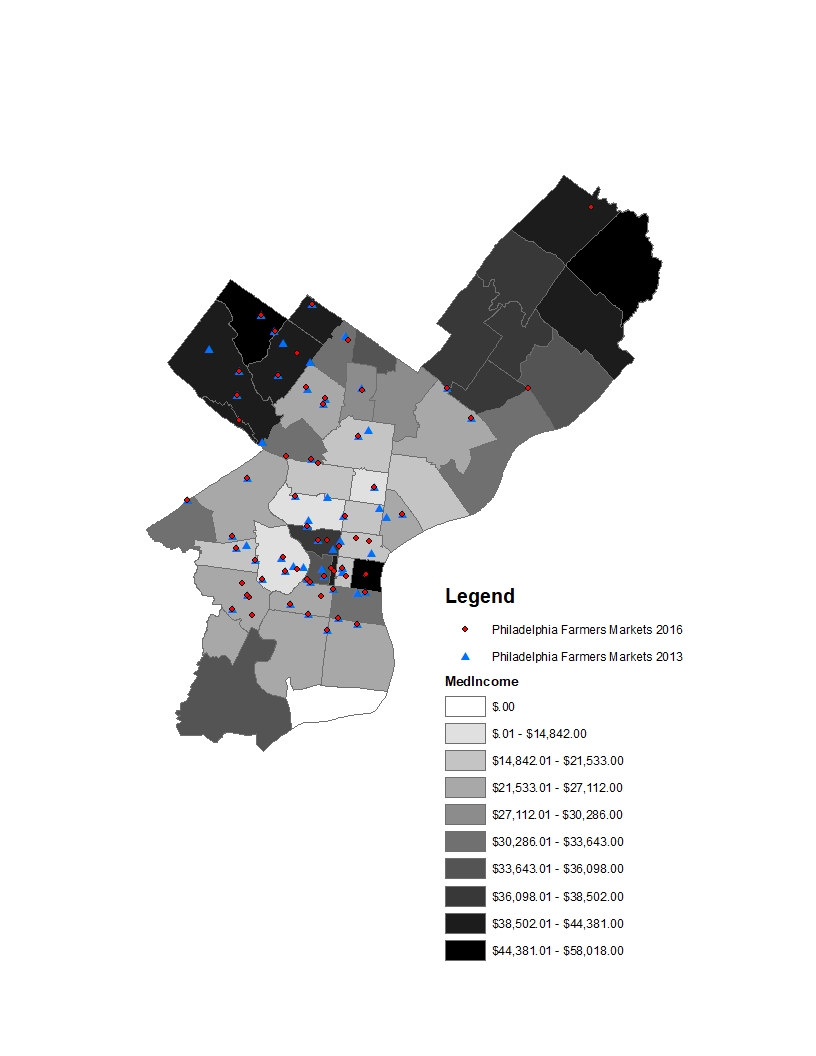
One potential explanation for the lack of markets in certain parts of the city may be low population. If we compare market locations to population at the zip code level (from the 2000 census), we see some support for this. However there are large pockets of North Philly with low population but high number of farmers markets.



## Discussion

Both the Nearest Neighbor Index, when considering the full area of Philadelphia, and the K-function Analysis indicate that the location of Philadelphia farmers markets in 2013 are significantly clustered. The addition of two new markets in the northeast in 2016 caused the NNI to characterize the pattern as random rather than clustered, when considering the full area of the region. However, the K-function Analysis was not significantly changed in 2016, still suggesting a clustered point pattern for all but the largest distances. So, these results are stable over time and not greatly impacted by the outlying new markets. This is generally consistent with our expectations as we think that farmers markets tend to be in parts of the city with similar types of people who are fond of farmers markets. This is also consistent with the visual examination of the point data on a map. The results from R and ArcMap are mostly consistent, with the only discrepancy being marginally different NNI values.

We considered population as a factor correlated with the location of farmers markets in Philadelphia. Income may also be a predictor of farmers markets. We have reason to believe that high income households are more likely to shop at and thus attract farmers markets. Again using 2000 census data, we look the spread of farmers markets in 2013 and 2016 overlaid on a choropleth maps median household income. There does seem to be some visual correlation between farmers market location and income level but further analysis is needed to confirm this.



Assuming that we can trust the outputs of the K functions, we can conclude that at all but the shortest of distances, the placement of farmers markets is significantly clustered. This is backed up by the results of the Nearest Neighbor Index when using the adjusted area of all of Philly. The policy implications of this analysis are that the benefits of farmers markets are concentrated among small pockets of Philadelphia. So, policies should encourage the opening of farmers markets in untapped areas of the city so that all can benefit. This could take the form of subsidies to open up farmers markets in under served areas of the city, or limitations on the amount of farmers markets within a certain radii from each other. These measures would encourage farmers markets to pop up in traditionally under served parts of the city.

##### Code Appendix

knitr::opts\_chunk$set(echo = FALSE)  
library(spatstat)  
library(rgdal)  
library(raster)  
library(sp)  
library(tidyverse)  
library(scales)  
library(gtools)  
library(maptools)  
library(geosphere)  
library(data.table)  
library(knitr)  
  
setwd("C:/Users/Ajjit/Google Drive/Documents/CPLN/CPLN HW 4")  
  
  
Philly = shapefile("Philadelphia.shp")  
farmers\_markets = shapefile("Philadelphia\_Farmers\_Markets201302.shp")  
zip\_codes = shapefile("Philadelphia\_ZipCodes.shp")  
  
  
  
 #NNI\_min\_enclosing\_rectanlge}  
  
  
#NEAREST NEIGHBOR ANALYSIS (The data have been slightly edited from the data used in ArcGIS)  
#Computes the distance from each point to its nearest neighbour in a point pattern.  
  
farmers\_marketsppp = as.ppp(farmers\_markets)  
  
  
  
nnd <- nndist.ppp(farmers\_marketsppp)  
#Using the formulas on the slides, we calculate Mean Observed Distance, Mean Expected Distance and SE.  
MeanObsDist <- mean(nnd)  
#The area.owin command calculates the area of the study area that you use. Here it's the minimum enclosing rectangle, but it doesn't have to be - it could be any shapefile you import from ArcGIS (or generate in R) that corresponds to the study area.  
MeanExpDist <- 0.5 / sqrt(nrow(farmers\_markets) / area.owin(as.owin(farmers\_marketsppp, fatal = TRUE)))  
SE <- 0.26136 / sqrt(nrow(farmers\_markets)\*nrow(farmers\_markets) / area.owin(as.owin(farmers\_marketsppp, fatal = TRUE)))  
  
# # If you want to use area of Philly, use: area.owin(as.owin(Philly))  
# MeanExpDist <- 0.5 / sqrt(nrow(farmers\_markets) / area.owin(as.owin(Philly)))  
# SE <- 0.26136 / sqrt(nrow(farmers\_markets)\*nrow(farmers\_markets) / area.owin(as.owin(Philly)))  
  
#Calculating the z-score  
zscore <- (MeanObsDist - MeanExpDist)/SE  
#Statistical test  
#Here, if the z score is positive, we do an upper-tailed test and if the z score is negative we do a lower-tailed test to come up with the p-value. In our case, z=5.02 and p<0.0001  
pval<-ifelse(zscore > 0, 1 - pnorm(zscore), pnorm(zscore))  
#Calculating the NNI. Here, NNI = 1.26 again, slightly different from the ArcGIS results.  
NNI <- MeanObsDist / MeanExpDist  
  
  
kable(tibble(NNI\_Value=NNI, p\_value =pval, z\_score = zscore))  
  
##p-value is 0.1401147, and we fail to reject the null of spatial randomness  
##NNI is 0.928,  
  
  
# If we use area of Philly instead of area of minimum enclosing rectangle, results show significant small amounts of spatial clustering:  
#pval = 0.0004119561  
#NNI = 0.7779648  
#NEAREST NEIGHBOR ANALYSIS (The data have been slightly edited from the data used in ArcGIS)  
#Computes the distance from each point to its nearest neighbour in a point pattern.  
  
farmers\_marketsppp = as.ppp(farmers\_markets)  
Window(farmers\_marketsppp) = as.owin(Philly)  
pp = farmers\_marketsppp  
  
  
nnd <- nndist.ppp(farmers\_marketsppp)  
#Using the formulas on the slides, we calculate Mean Observed Distance, Mean Expected Distance and SE.  
MeanObsDist <- mean(nnd)  
#The area.owin command calculates the area of the study area that you use. Here it's the minimum enclosing rectangle, but it doesn't have to be - it could be any shapefile you import from ArcGIS (or generate in R) that corresponds to the study area.  
  
# # If you want to use area of Philly, use: area.owin(as.owin(Philly))  
MeanExpDist <- 0.5 / sqrt(nrow(farmers\_markets) / area.owin(as.owin(farmers\_marketsppp, fatal = TRUE)))  
SE <- 0.26136 / sqrt(nrow(farmers\_markets)\*nrow(farmers\_markets) / area.owin(as.owin(farmers\_marketsppp, fatal = TRUE)))  
  
#Calculating the z-score  
zscore <- (MeanObsDist - MeanExpDist)/SE  
#Statistical test  
#Here, if the z score is positive, we do an upper-tailed test and if the z score is negative we do a lower-tailed test to come up with the p-value. In our case, z=5.02 and p<0.0001  
pval<-ifelse(zscore > 0, 1 - pnorm(zscore), pnorm(zscore))  
#Calculating the NNI. Here, NNI = 1.26 again, slightly different from the ArcGIS results.  
NNI <- MeanObsDist / MeanExpDist  
  
  
kable(tibble(NNI\_Value=NNI, p\_value =pval, z\_score = zscore))  
  
#K-Functions  
#Double click on the khat data set. It will have 513 observations and 5 variables. We are interested in 2 variables:  
#-- r, which is the distance that goes in increments of 138.8693  
#-- iso, which is the k-function calculated with Ripley's edge correction  
khat = Kest(pp , rmax=28000)  
#Plots K function calculated with Ripleys isotropic edge correction, line width 2, axis labels, and title.  
plot(khat$r,khat$iso,xlab="r", ylab="Ripleys K",  
main="Ripleys Estimated K-Function",  
cex.lab=1.6,cex.axis=1.5,cex.main=1.5,lty=1,lwd=2)  
# Overlays the theoretical K-function under CSR with a dashed (lty=8) line.  
lines(khat$r,khat$theo,lty=8, lwd=2)  
#Computes confidence envelopes using n=199 simulations. Here, nrank=1 means we're looking at the lowest and highest values of the simulated envelopes. Here, alpha = 2 \* nrank/(1 + nsim) = 2\*1/200 = 0.01  
#spatstat::envelope is to specify that the envelope command is in the spatstat library and not the boot library.  
Kenv <- spatstat::envelope(pp,fun="Kest", rmax=28000, nsim=99, nrank=1)  
# Plots Ripleys K function with 99% simulation # envelopes, axis labels, and a title.  
plot(Kenv,xlab="r",ylab="Khat(r)", cex.lab=1.6,cex.axis=1.5,main=  
"Ripleys Khat with Confidence Envelopes",cex.main=1.5,lwd=2)  
  
  
  
lhat <- Lest(pp, rmax=28000)  
#Plots L function with line width 2, Ripleys isotropic edge correction, axis labels, and title.  
plot(lhat$r,lhat$iso-lhat$r, xlab="r",ylab="Ripleys L",cex.lab=1.6,  
 cex.axis=1.5,cex.main=1.5,lty=1,lwd=2, main="Ripleys Estimated L-Function")  
#Overlays the theoretical L-function under CSR with a dashed (lty=8) line.  
lines(lhat$r,lhat$theo-lhat$r,lty=8,lwd=2)  
  
#Code to compute the Ripleys Simulation Confidence Envelopes  
#Computes confidence envelopes using n=199 simulations. Here, nrank=1 means we're looking at the lowest and highest values of the simulated envelopes. Here, alpha = 2 \* nrank/(1 + nsim) = 2\*1/200 = 0.01  
 Lenv <- spatstat::envelope(pp,fun="Lest", rmax=28000, nsim=199,nrank=1)  
  
 # Plots Ripleys L function with 99% simulation envelopes, axis labels, and a title.  
 plot(Lenv,xlab="r",ylab="Lhat(r)", cex.lab=1.6,cex.axis=1.5, main= "Ripley's Lhat with Confidence Envelopes", cex.main=1.5, lwd=2,legend=F)