Point pattern tests: Andresen's Spatial point pattern test

Martin A. Andresen
m.andresen@griffith.edu.au
School of Criminology and Criminal Justice
Griffith University, Gold Coast Campus
Southport, Queensland 4215

INTRODUCTION

A key question in crime and place research is whether or not there are differences in the spatial patterns of two crime incident datasets at the same level of analysis. For example, an analyst might hypothesize that different crime types have different spatial opportunity structures (commercial and residential burglary) or that the opportunity structure of one crime type changes over time (Hodgkinson & Andresen, 2019), and thus a statistically significant spatial differences in two crime data sets may be present. The question is: how does one test for these differences in spatial point patterns?

One method would simply to plot the point patterns to see if there are any differences. However, as shown by Andresen (2009), this could lead to misleading conclusions. Alternatively, using areal units (street segments, grids, census tracts, etc.) one could calculate counts or percentages in each area or create kernel density maps and visually inspect any differences; this also could be problematic because there is no simple way to test for statistically significant differences.

If one wishes to use inferential tests, either distance-based or area-based tests can be used; distance-based tests use the distance between points to determine a random, clustered, or uniform spatial pattern, whereas area-based tests count the number of points within defined areas, most often a pre-defined zones (street segments, grids, census tracts, etc.) (Bailey & Gatrell, 1995). Though local tests (spatial statistics that measure statistical significance for spatial relationships for each spatial unit of analysis) do exist in area-based application, they test for spatial clustering at the local level, not the similarity or difference in spatial patterns, a subtle but important difference (Anselin, 1995; Getis & Ord, 1992). Andresen's (2009) spatial point pattern test (SPPT) allows for the identification of similarities or differences in spatial point patterns at the local level with statistical significance testing.

SPATIAL POINT PATTERN TEST

Conceptual Aspects

Andresen's SPPT has been used in a variety of contexts, dominantly criminological:

- the similarity of spatial crime patterns (Andresen, 2009),
- changing patterns in international trade (Andresen, 2010),
- crime pattern stability (Andresen & Malleson, 2011),
- the spatial impact of the aggregation of crime types (Andresen & Linning, 2012),
- seasonality and spatial crime patterns (Andresen & Malleson, 2013a; Linning, 2015),
- spatial patterns and the modifiable areal unit problem (Andresen & Malleson, 2013b),

- crime displacement (Andresen & Malleson, 2014),
- comparing open source crime data and actual police data (Tompson et al., 2015),
- the changing spatial patterns of crime with regard to the crime drop (Hodgkinson et al, 2016; Hodgkinson & Andresen, 2019; Pereira et al., 2016),
- the spatial dimension of police proactivity (Andresen & Hodgkinson, 2018; Wu & Lum, 2017),
- and the stability of crime concentration at micro-places (Andresen et al., 2017a; Vandeviver & Steenbeek, 2019).

Andresen's SPPT identifies differences/similarities in the spatial patterns of two or more point data sets, using an area-based methodology. In general, the SPPT includes a global statistic, Andresen's *S,* that measures the percentage of spatial units have are similar for two (or more) data sets. Additionally, being a locally-based test, the output of the test can be mapped so the researcher can see *where* differences in spatial patterns occur.

There are a few different varieties of SPPT, described in Wheeler et al. (2018), but we will focus the example here, and the step-by-step, below, using the full bootstrap on all data sets. The general steps of SPPT are as follows, using two point data sets as an example, with greater detail available in Andresen (2009, 2016) and Wheeler et al. (2018):

1. Identify one of your two data sets as the base and the other as the test. If considering change over time the identification of the base is easy because it is the earlier time period.
2. Run a full bootstrap (100% sampling with replacement) on both the base and test data sets a number of times (200 is the default).
3. For each area in the data sets (each street segment, census tract, neighborhood, etc.) and each iteration of the bootstrap, calculate the percentage of events.
4. Create a 95 percent nonparametric confidence interval for each area in both data sets by ranking all of the percentages of events in Step 3 and removing the top and bottom 2.5 percent of percentages (5 from the top and bottom when using 200 simulations).
5. Identify similarities using the confidence intervals. If the base and test confidence intervals overlap for a given area, the spatial patterns for that area are deemed similar and can be given a value of 1; if the confidence intervals do not overlap that area can is deemed dissimilar and can be given a value of 0.

The output of this test is a global index of similarity, *S*, that ranges between 0 (no similarity) and 1 (perfect similarity), calculated as follows:

$$S = \frac{\sum_{i=1}^{n} s_i}{n} \tag{1}$$

- where s_i is equal 1 if the pattern of two datasets are similar within an individual spatial unit of analysis and 0 otherwise (Step 5), and
- n is the number of areas.

As such, the similarity index measures the percentage of areas (street segments, census tracts, neighborhoods, etc.) that share a similar spatial pattern. There are no strict guidelines when S achieves a threshold to indicate "similarity", but the general practice is to use 0.80 or 0.90, based on the variance inflation factor (VIF). The VIF relates to multicollinearity in a regression context with a value greater than 5 or 10 to be considered problematic (O'Brien, 2007). In a bivariate context, VIF values in this range would coincide with correlation coefficients ranging from 0.80 to 0.90, hence the S-Index value of 0.80. However, it should be noted that two point patterns labelled as similar in this context could lead to a lot of places being different when the overall patterns is deemed similar. For example, in the crime and place literature that uses street segments or micro-places it is not uncommon to have 10,000 or more places in the data set; with a S-Index of 0.80 (similarity), that is 2000 places that exhibit dissimilarity. Because of this, it is important to use the micro-level output from SPPT to map the results and see where the two spatial point patterns are different.

One issue that arises with the S-Index is that it may overstate the degree of similarity, particularly when there are more areal units than there are events. As a result, there are more places than there are events for them to occur within. Consider a rare event that only had 1000 cases in each of 2 years and a set of 10,000 street segments. Even if the two years of data occur in completely different places the S-Index would still be 0.80, indicating a high degree of similarity. In order to control for this situation, there is also the Robust S-Index that uses n^* in the denominator, where n^* represents the number of places that actually have events in any of the point data sets. When using n^*, the situation described above would lead to $S = 0.0$, as it should be.

In the end, the researcher has an indication of how similar/different two spatial point patterns are. This can allow the researcher to make choices about aggregating data such as different crime types (Andresen & Linning, 2012), if a policy initiative has been effective by measuring the presence/absence of crime displacement (Andresen & Hodgkinson, 2018; Andresen & Malleson, 2014), or if crime patterns are stable over time (Andresen & MAlleson, 2011; Andresen et al., 2017a). Moreover, and in the spirit of local indicators of association (Anselin, 1995), the output of SPPT can be mapped to be able to identify where spatial patterns are the similar or different.

SPPT Data Requirements

As should be clear from the discussion above, there are three data sets that are necessary to undertake SPPT:

1. a base point data set
2. a test point data set
3. an area data set

For example, as used below, residential burglary 2003, residential burglary 2016, and dissemination areas—dissemination areas are areas defined by Statistics Canada that are similar in size to the US Census Block Group, approximately 400-700 persons.

When using areas such as census tracts, grids, and neighborhoods that cover the entire study area there are no concerns for SPPT. However, when using street segments that are lines, not areas, a small buffer must be applied around each street segment in order for SPPT to know if a point falls "inside" of a street segment. When doing this, however, overlapping areas are created where different street segments intersect. This is problematic because of double counting. In order to correct for this these overlapping areas need to be identified as intersections and considered separate spatial units of analysis or generate buffers around the street segments that do not overlap. The number of these intersections can be high, even in a city that is not geographically large. For example, in Vancouver, Canada (115 square kilometres) there are approximately 12,000 street segments plus 6,000 intersections; this increases the number of spatial units of analysis by 50 percent and has implications for the global *S*-Index, showing the importance of considering both the *S*-Index and the Robust *S*-Index.

With regard to the point data sets, because SPPT is an area-based test, the precise location of the event is not important, only that the event is placed in the correct areal unit. As such, if you only have the centroid of a census tract for geographic coordinates (the geographic centre of the census tract), SPPT can still be performed. This is the case for all units of analysis. For example, the street segment data used below use the centroid (mid-point) of the street segment for all events that occur on that street segment regardless of the precise location on the street segment.

Limitations of the SPPT

As with any statistical analysis, SPPT has its limitations.

First, in its most common usage, it is only for comparing the similarity of two data sets: two different years of the same phenomenon or two different crime types (e.g. Andresen & Malleson (2011) and Andresen & Linning (2012)). However, SPPT has been extended into a longitudinal version that allows for the identification of spatial trajectories (Andresen et al., 2017a). This methodological extension allows for the trajectory type to be identified at the micro-place level rather than other data reduction techniques such as group-based trajectory modelling.

The second limitation of SPPT is that it is exploratory in its application. Though SPPT is inferential, it is not explanatory. This is a similar limitation in other locally-based spatial statistical techniques. However, the local output from SPPT can be used as a dependent variable for explanatory analyses, similar to that undertaken using other spatial statistics (Andresen, 2011).

STEP-BY-STEP EXAMPLE

In order to perform SPPT in an analysis, one must first install some other software. SPPT is operationalised in the statistical program R. As such, if you do not have R installed you need to install it to be able to apply SPPT using the data sets provided: residential burglary 2003, residential burglary 2016, and dissemination areas (all data are for the City of Vancouver, Canada in 2016).[1] SPPT is implemented as a library within R, but it is not part of the Comprehensive R Archive Network (CRAN). Because of this, as shown below, SPPT needs to be "built" within R (all automatically) so Rtools (the recommended version) also needs to be installed: https://cran.r-project.org/bin/windows/Rtools/. Once all of this has been installed you are ready to undertake the analyses.

Data Set

The data sets included for this example and available for free download, as indicated above, are commercial burglary, residential burglary, and dissemination areas. These data are all ready to go to work your way through the example. The hypothesis for this

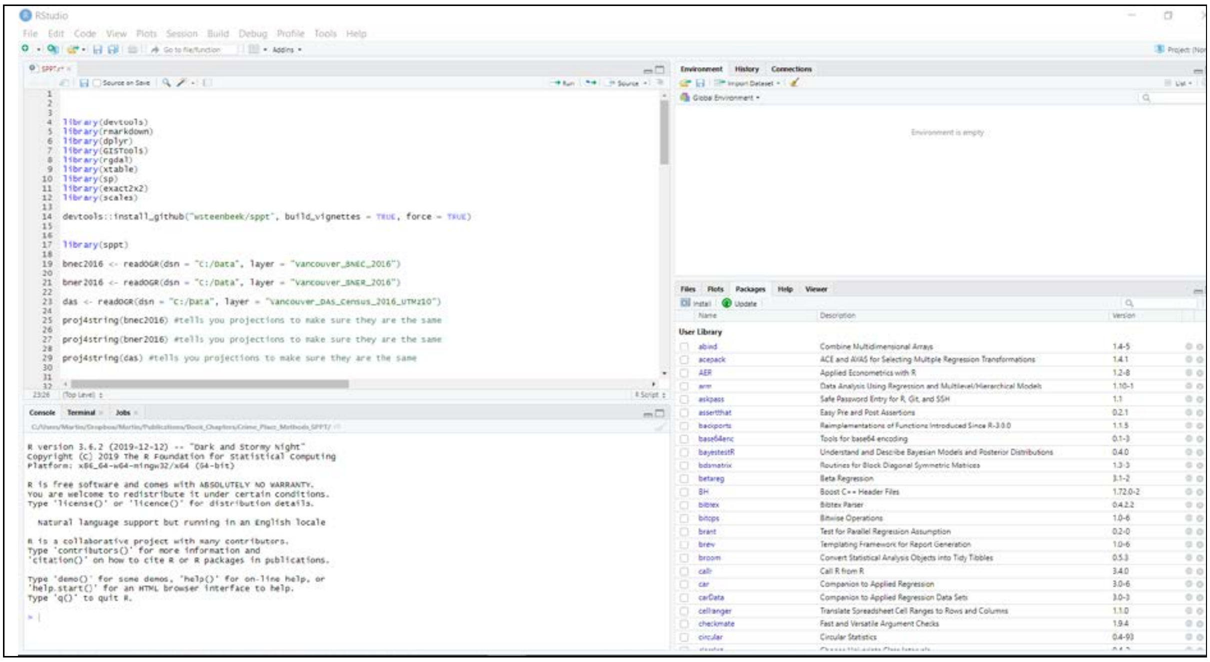
[1] R for Windows can be downloaded and installed here: http://cran.r-project.org/bin/windows/base/ with Mac and Linux versions also available. Once R is installed, it is recommended that you install R Studio https://www.rstudio.com/products/rstudio/download/. Though R Studio is not necessary is a an excellent (and free) environment that makes using R much easier than the default R Console.

example is as follows: residential burglary 2003 (6883 events) and residential burglary 2016 (2994 events) have the same spatial pattern when considering dissemination areas (991 areas). The R code for this example is in the appendix at the end of this chapter, but also available for download with the other data for this book.

An Application of the Spatial Point Pattern Test

Step 1: Start R and load the SPPT library

You can begin the application of SPPT by either opening R Studio and pasting/typing in the code provided here, or downloading the R Code file from the book resources and double-clicking on that file that will open R Studio. If you open the R code file provided with the book (SPPT.r) your screen should look something like this:



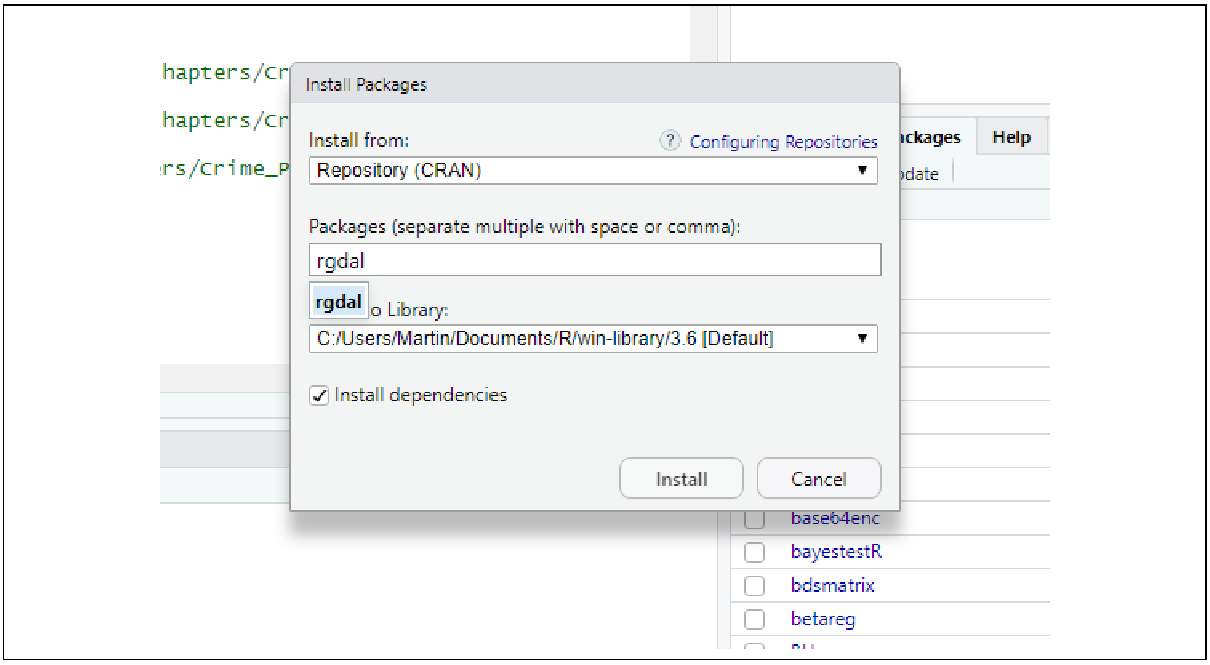
The top left section of R Studio is where your code is placed. In that section you can make changes to the code as you need and it won't run until you click on "Run" in the top-right portion of that section. The output from R (including any warnings) is at the bottom left, the environment (data, output, etc.) is listed in the top-right section, and the bottom-right section lists your packages/libraries as well as graphical output from R (graphs, figures, maps, etc.).

The first set of commands you will find is a number of libraries that you need to load in R for SPPT to work:

```
library(devtools)
library(rmarkdown)
library(dplyr)
library(GISTools)
library(rgdal)
library(xtable)
library(sp)
library(exact2x2)
```

library(scales)

You can highlight all of them, click on "Run", and they will be loaded if they are installed, or you can place your cursor at any point in the line for each library and click "Run" to make sure they each load properly—there can be a lot of messages in R sometimes. If any of these libraries are not installed, click on the "Packages" tab in the bottom-right section of R Studio and then click on "Install". A small window will pop-up and you type in the name of the library/package and it will auto complete for you:



Make sure "Install dependencies" is checked (this installs any other packages you will need) and then click on "Install". Once this has been done properly for all necessary packages, you will be able to load all of the necessary libraries for SPPT to be downloaded and installed.

The next line of code installs the SPPT library:

devtools::install_github("wsteenbeek/sppt", build_vignettes = TRUE, force = TRUE)

Dr. Wouter Steenbeek's GitHub page <https://github.com/wsteenbeek/sppt>, has instructions for doing this with the pre-installed data available. This line of code will install SPPT in R as well as vignettes that provide a few other examples of SPPT. Very early in the installation you will be asked something that looks like the following:

```
Building the package will delete...
  'C:/Users/Martin/AppData/Local/Temp/Rtmpcf9wsv/remotes1ffc51c67dca/wstee
nbeek-sppt-25ae963/inst/doc'
Are you sure?

1: Yes
2: No
```

Type "1" in the bottom-left section where the cursor is and press enter. SPPT will then be installed. Now run the following line of code, library(sppt), and you are ready to begin.

Step 2: Load all necessary data and test for data compatibility

The first thing to do at this point is to load the data. The following lines of code load residential burglaries 2003 (bner2003), residential burglaries 2016 (bner2016), and dissemination areas (das) into R that are included in the sppt library:

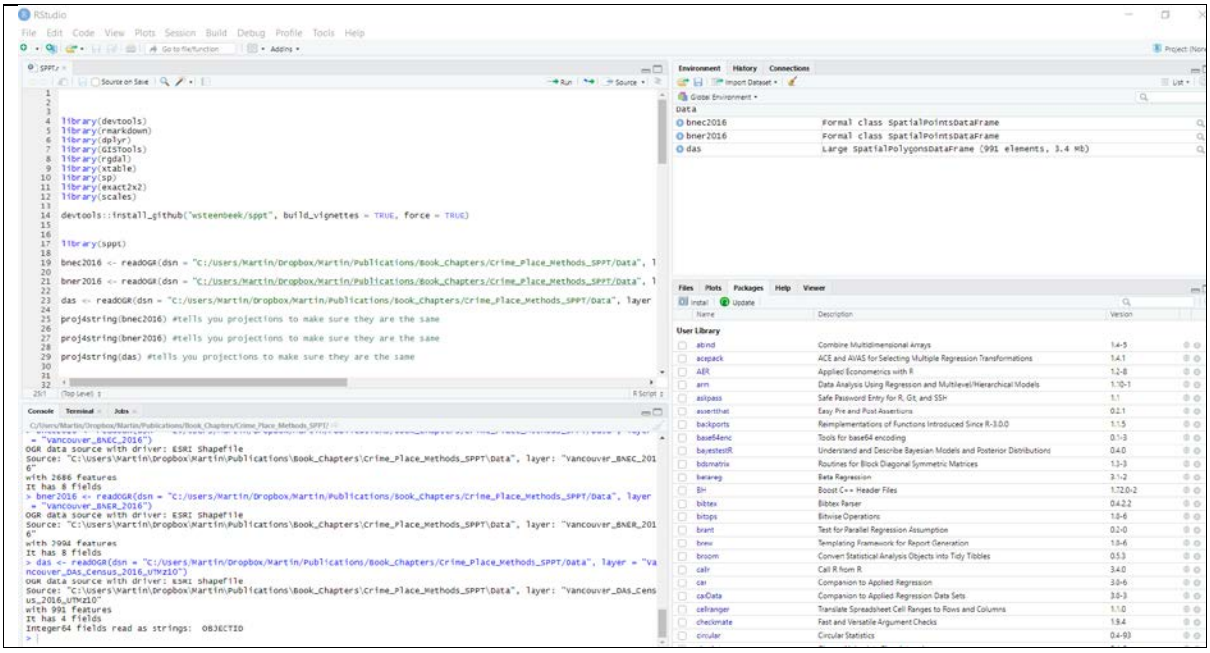
```
bner2003 <- readOGR(dsn = "C:/Users/Martin/Documents/R/win-library/3.6/sppt/extdata", layer = "Vancouver_Residential_Burglary_2003")
bner2016 <- readOGR(dsn = "C:/Users/Martin/Documents/R/win-library/3.6/sppt/extdata", layer = "Vancouver_Residential_Burglary_2016")
das <- readOGR(dsn = "C:/Users/Martin/Documents/R/win-library/3.6/sppt/extdata", layer = "Vancouver_DAs_2011_Census_UTMz10")
```

The first part of the code names the data set, bner2003, for example, is the point data set of residential burglaries in Vancouver during 2003.

"C:/Users/Martin/Documents/R/win-library/3.6/sppt/extdata" is the folder they are installed in, and the names of the files (they are called shapefiles) are listed last—you will see in the screenshots, below, that I have abbreviated the file path name.

Once these data successfully load, you will get a brief description of the data (number of observations, number of variables, and so on.

You will also see these data sets listed in the top-right section of R Studio:



The next set of code checks the projection of the different data sets used to make sure they each show up on a map in the same way:

```
proj4string(bner2003) #tells you projections to make sure they are the same
proj4string(bner2016)
```

proj4string(das)

After running the above code, you can see from the results that each of the three data sets have the same numbers for their respective projections:

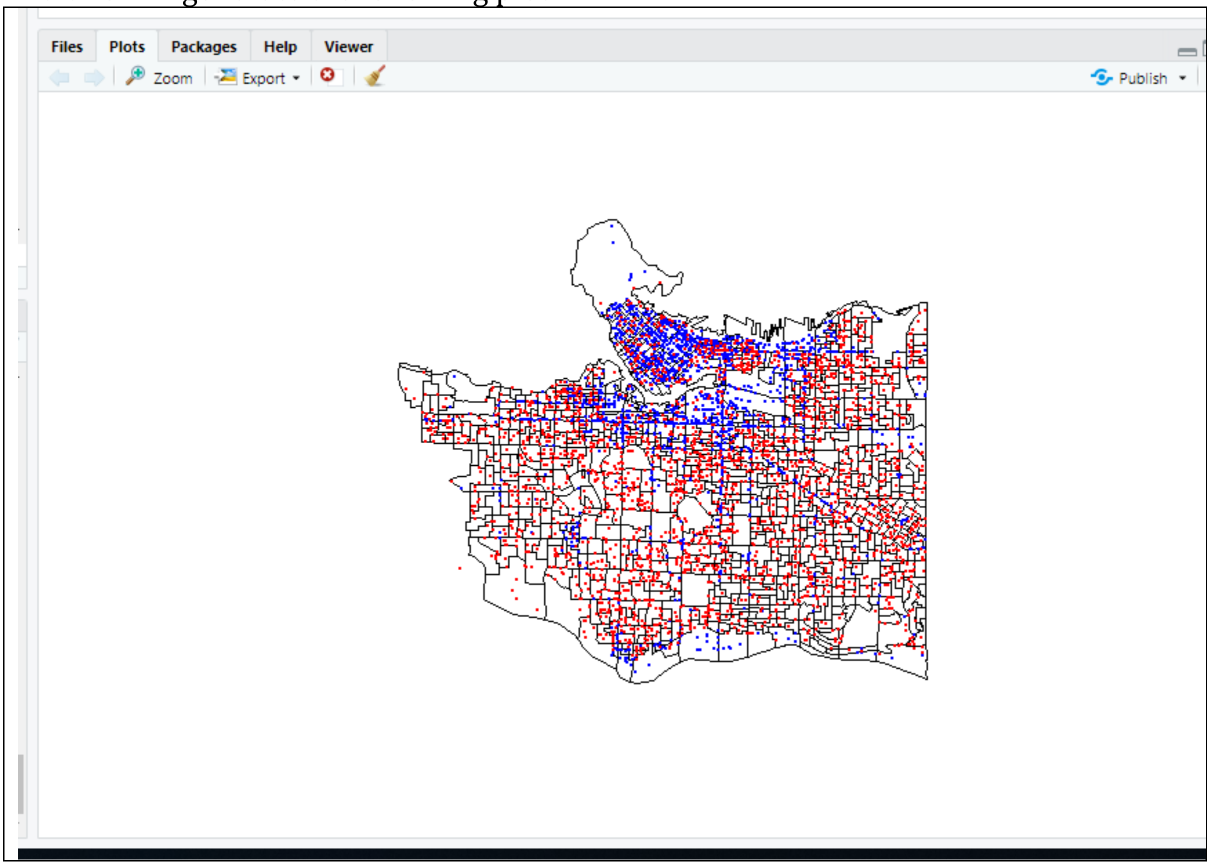
```
> proj4string(burglary2003) #tells you projections to make sure they are t
he same
[1] "+proj=utm +zone=10 +datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs
84=0,0,0"
> proj4string(burglary2016) #tells you projections to make sure they are t
he same
[1] "+proj=utm +zone=10 +datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs
84=0,0,0"
> proj4string(das) #tells you projections to make sure they are the same
[1] "+proj=utm +zone=10 +datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs
84=0,0,0"
```

If this is not the case, when you run SPPT you will get an error message and this will need to be fixed using a spatial data management software of your choice (probably a GIS).

At this point, you can generate a quick map of the data using the following code to make everything is working properly:

plot(das)
points(bner2003, col="blue", pch=16, cex=.3)
points(bner2016, col="red", pch=16, cex=.3)

And this will generate the following plot:



At this point it is difficult to tell if there are any differences between residential 2003 (red) and residential 2016 (blue) burglary, primarily because there are almost 10,000 points on this map.

The R file provided has some code for help files and the vignettes if you are interested, as well:

```
?sppt
browseVignettes("sppt")
vignette("sppt_intro", package = "sppt")
vignette("sppt_diff", package = "sppt")
```

However, we will not go over those here.

Step 3: Run the SPPT

Now it is time to run the test, using the following code:

```
myoutput_boot <- sppt_boot(bner20003, bner2016, das)
```

This tells R to use the full bootstrap version of the test (100 sampling with replacement for both base and test data), you are calling the output from this code "myoutput_boot", commercial burglary in the base data, residential burglary is the test data (this distinction is only important for interpreting the output), and Das are the spatial units of analysis. The test should only take a few seconds to run for these data (more data, whether they be points or areas) increases the computational time.

The Global *S*-Index can be obtained using the following code:

```
summary.sppt(myoutput_boot)
```

with the following output:

```
> summary.sppt(myoutput_boot)
$globalS.standard
[1] 0.764884

$globalS.robust
[1] 0.76417
```

This tells us that the standard *S*-Index is 0.76, indicating that 76 percent of the areas have similar spatial patterns for commercial and residential burglary. The Robust *S*-Index, however, is also 0.76, showing little impact of having smaller areas that do not have any data in them for the base or test data sets.

It is important to note here that because this test is a Monte Carlo simulation, the *S*-Index values will change slightly every time you run the test.[2] See if we run the test again, we get slightly different results:

```
> summary.sppt(myoutput_boot)
$globalS.standard
[1] 0.7598385

$globalS.robust
[1] 0.7591093
```

Step 4: Visualize the results

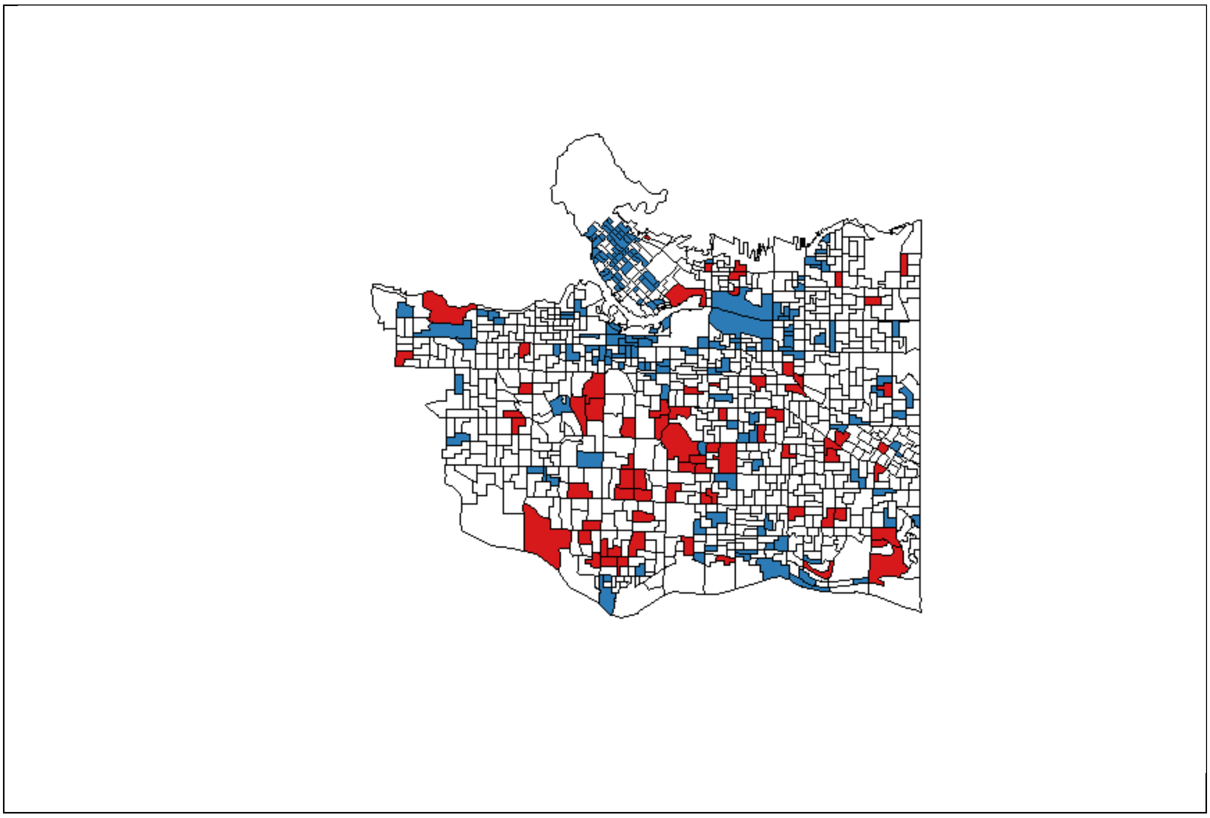
These changes are small, but they do emerge, especially when there are more areas. Now the output from SPPT can be mapped using the following code and with the following map:

```
plot(myoutput_boot)
plot(myoutput_boot[which(myoutput_boot$localS == -1),], col="#2c7bb6", add = TRUE)
#shows where base > test
plot(myoutput_boot[which(myoutput_boot$localS == 1),], col="#d7191c", add = TRUE)
#shows where test > base
```

The blue areas are the areas where the percentage of residential burglaries in 2003 is greater than the percentage of residential burglaries in 2016. The red areas are the areas where the percentage of residential burglaries in 2016 is greater than the percentage of residential burglaries in 2003. The white areas are areas where the percentages of commercial and residential burglaries are statistically the same; these are the areas that are counted in the numerator for the *S*-Index calculations. In the current context, this can be understood with regard to the changing opportunity structures changing in Vancouver; during this time, changes in security for apartment buildings led to a shift in the spatial patterns of residential burglaries to areas with more single-family homes (Hodgkinson & Andresen, 2019). Generating these maps in R is particularly useful for visualizing the results immediately, making any initial observations of change patterns before you have to make a more aesthetically pleasing map in a GIS—it can also save you time if your results end up not being interesting and worth pursuing.

[2] If you wish to maintain consistency in results, you could use R code to set a seed, in the following way, such that you tell R which seed to use every time you re-run the same test:

```
set.seed(93255)
summary.sppt(myoutput_boot)
```



Step 5: Export the output in order to visualize in GIS software

The last line of code:

writeOGR(obj=myoutput_boot, dsn="C:/Data", layer="Burglary_2003_2016_DAs_SPPT_Boot", overwrite_layer=TRUE, driver="ESRI Shapefile") # export shapefile of output

This exports the output from SPPT into a shapefile so you can make better maps in a GIS. It is important to note that you will always get the following warning:

```
Warning message:
In writeOGR(obj = myoutput_boot, dsn = "C:/Users/Martin/Dropbox/Martin/Publ
ications/Book_Chapters/Crime_Place_Methods_SPPT/Data",  :
  Field names abbreviated for ESRI Shapefile driver
```

This is no cause for concern as it only means that variables names in this shapefile may be different from the files that you loaded into R in the beginning.

MOVING FORWARD

SPPT has been applied in a variety of contexts and an increasing number of research projects over the past decade. At this time, it has a few variations (see Wheeler et al., 2018) but the fundamental aspects of SPPT are unchanged. As such, there are no obvious steps or omissions in the basis for the test to alter moving forward, though there will most likely be further refinements. The longitudinal version of SPPT, however, is underdeveloped. Andresen et al. (2017a) discussed how sequential pairwise tests may lead to spurious results over the longitudinal time period. Specifically, even if the *S*-Index is >= 0.90 for a number of consecutive pairwise comparisons, if that 10 percent of non-similar areal units changes every time a successive temporal comparison is made (2010-2011, then 2011-2012, then 2012-2013, and so on), the actual spatial similarity be quite low after only 10 years.

In order to address this, Andresen et al. (2017) extended the original SPPT into a longitudinal version of the test in order to identify spatial trajectories in the context of the crime and place literature. They were able to show that when stability over time was assessed at the micro-level, spatial crime patterns are not as similar as previous research has indicated. Though this longitudinal SPPT can be undertaken using the output from analyses similar to what was shown above, it has not yet been automated in the SPPT R library. Moving forward, this longitudinal version of SPPT (and any others) will be incorporated into the R library.

Lastly, SPPT has only been used in an exploratory fashion. Moving forward the local output from SPPT should be used as a dependent variable in models that can be used to assess why that spatial change has occurred. This is particularly important if that spatial change has occurred from a (crime) prevention intervention.

REFERENCES

Andresen, M. A. (2016). An area-based nonparametric spatial point pattern test: The test, its applications, and the future. *Methodological Innovations*, 9, Article 12.

Andresen, M. A. (2009). Testing for similarity in area-based spatial patterns: A nonparametric Monte Carlo approach. *Applied Geography*, 29(3), 333 - 345.

Andresen, M.A. (2010). Canada – United States interregional trade: Quasi-points and spatial change. *Canadian Geographer*, 54(2), 139 – 157.

Andresen, M. A. (2011). Estimating the probability of local crime clusters: The impact of immediate spatial neighbors. *Journal of Criminal Justice*, 39(5), 394 – 404.

Andresen, M. A., & Hodgkinson, T. (2018). Evaluating the impact of police foot patrol at the micro-geographic level. *Policing: An International Journal*, 41(3), 314 - 324.

Andresen, M. A., & Linning, S. J. (2012). The (in)appropriateness of aggregating across crime types. *Applied Geography*, 35(1-2), 275 - 282.

Andresen, M. A., & Malleson, N. (2011). Testing the stability of crime patterns: Implications for theory and policy. *Journal of Research in Crime and Delinquency*, 48(1), 58 - 82.

Andresen, M. A., & Malleson, N. (2013a). Crime seasonality and its variations across space. *Applied Geography*, 43, 25 - 35.

Andresen, M. A., & Malleson, N. (2013b). Spatial heterogeneity in crime analysis. In M. Leitner (Ed.), *Crime modeling and mapping using geospatial technologies* (pp. 3 – 23). New York, NY: Springer.

Andresen, M. A., & Malleson, N. (2014). Police foot patrol and crime displacement: A local analysis. *Journal of Contemporary Criminal Justice*, 30(2), 186 - 199.

Andresen, M. A., Linning, S. J., & Malleson, N. (2017a). Crime at places and spatial concentrations: Exploring the spatial stability of property crime in Vancouver BC, 2003-2013. *Journal of Quantitative Criminology*, 33(2), 255 - 275.

Andresen, M. A., Curman, A. S. N., & Linning, S. J. (2017b). The trajectories of crime at places: Understanding the patterns of disaggregated crime types. *Journal of Quantitative Criminology*, 33(3), 427 - 449.

Anselin, L. (1995). Local indicators of spatial association – LISA. *Geographical Analysis*, 27(2), 93 – 115.

Bailey, T. C., & Gatrell, A. C. (1995). *Interactive spatial data analysis*. Harlow, UK: Prentice Hall.

Cohen, L. E., & Felson, M. (1979). Social change and crime rate trends: A routine activity approach. *American Sociological Review*, 44(4), 588 – 608.

Getis, A., & Ord, J.K. (1992). The analysis of spatial association by use of distance statistics. *Geographical Analysis*, 24(3), 189 – 206.

Hodgkinson, T., & Andresen, M. A. (2019). Changing spatial patterns of residential burglary and the crime drop: The need for spatial data signatures. *Journal of Criminal Justice*, 61, 90 – 100.

Hodgkinson, T., Andresen, M. A., & Farrell, G. (2016). The decline and locational shift of automotive theft: A local level analysis. *Journal of Criminal Justice*, 44(1), 49 - 57.

Linning, S. J. (2015). Crime seasonality and the micro-spatial patterns of property crime in Vancouver, BC and Ottawa, ON. *Journal of Criminal Justice*, 43(6), 544 – 555.

O'Brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity*, 41(5), 673 – 690.

O'Brien, D. T., & Winship, C. (2017). The gains of greater granularity: The presence and persistence of problem properties in urban neighborhoods. *Journal of*

Quantitative Criminology, 33(3), 649 – 675.
Pereira, D. V. S., Mota, C. M. M., & Andresen, M. A. (2016). The homicide drop in Recife, Brazil: A study of crime concentrations and spatial patterns. *Homicide Studies*, 21(1), 21 - 38.
Schnell, C., Braga, A. A., & Piza, E. L. (2017). The influence of community areas, neighborhood clusters, and street segments on the spatial variability of violent crime in Chicago. *Journal of Quantitative Criminology*, 33(3), 469 – 496.
Steenbeek, W., & Weisburd, D. (2016). Where the action is in crime? An examination of variability of crime across spatial units in The Hague, 2001–2009. *Journal of Quantitative Criminology*, 32(3), 449 – 469.
Steenbeek, W., Vandeviver, C. Andresen, M.A., Malleson, N., & Wheeler, A. (2018). *sppt: Spatial Point Pattern Test. R package version* (0.1.5, 0.1.6). Retrieved from: https://github.com/wsteenbeek/sppt
Tompson, L., Johnson, S., Ashby, M., Perkins, C., & Edwards, P. (2015). UK open source crime data: Accuracy and possibilities for research. *Cartography and Geographic Information Science*, 42(2), 97 - 111.
Vandeviver, C., & Steenbeek, W. (2019). The (in)stability of residential burglary patterns on street segments: The case of Antwerp, Belgium 2005 - 2016. *Journal of Quantitative Criminology*, 35 (1), 111 – 133.
Weisburd, D., Bushway, S., Lum, C., & Yang, S-M. (2004). Trajectories of crime at places: A longitudinal study of street segments in the City of Seattle. *Criminology*, 42(2), 283 – 321.
Weisburd, D., Groff, E. R., & Yang, S-M. (2012). *The criminology of place: Street segments and our understanding of the crime problem*. New York, NY: Oxford University Press.
Wheeler, A. P., Steenbeek, W., & Andresen, M. A. (2018). Testing for similarity in area-based spatial patterns: Alternative methods to Andresen's spatial point pattern test. *Transactions in GIS*, 22 (3), 760 – 774.

APPENDIX

```
library(devtools)
library(rmarkdown)
library(dplyr)
library(GISTools)
library(rgdal)
library(xtable)
library(sp)
library(exact2x2)
library(scales)

devtools::install_github("wsteenbeek/sppt", build_vignettes = TRUE, force = TRUE)

library(sppt)

bnec2016 <- readOGR(dsn = "C:/Data", layer = "Vancouver_BNEC_2016")

bner2016 <- readOGR(dsn = "C:/Data", layer = "Vancouver_BNER_2016")

das <- readOGR(dsn = "C:/Data", layer = "Vancouver_DAs_Census_2016_UTMz10")

proj4string(bnec2016) #tells you projections to make sure they are the same

proj4string(bner2016)

proj4string(das)

plot(das)
points(bnec2016, col="blue", pch=16, cex=.3)
points(bner2016, col="red", pch=16, cex=.3)

?sppt # help file: follow the example steps to see the function in action

# https://github.com/wsteenbeek/sppt/tree/master/vignettes

browseVignettes("sppt")

vignette("sppt_intro", package = "sppt") #
https://github.com/wsteenbeek/sppt/blob/master/vignettes/sppt_comparison.Rmd

vignette("sppt_diff", package = "sppt") #
https://github.com/wsteenbeek/sppt/blob/master/vignettes/sppt_diff.Rmd

myoutput_boot <- sppt_boot(bnec2016, bner2016, das) # will automatically do a full
bootstrap (100%, with replacement) on both sides # i.e. bootstrap = TRUE is the default

summary.sppt(myoutput_boot)
```

```
plot(myoutput_boot)
plot(myoutput_boot[which(myoutput_boot$localS == -1),], col="#2c7bb6", add = TRUE)
#shows where base > test
plot(myoutput_boot[which(myoutput_boot$localS == 1),], col="#d7191c", add = TRUE)
#shows where test > base

writeOGR(obj=myoutput_boot, dsn="C:/ Data",
layer="Burglary_2003_2016_DAs_SPPT_Boot", overwrite_layer=TRUE, driver="ESRI
Shapefile") # export shapefile of output
```