GIS Final Project

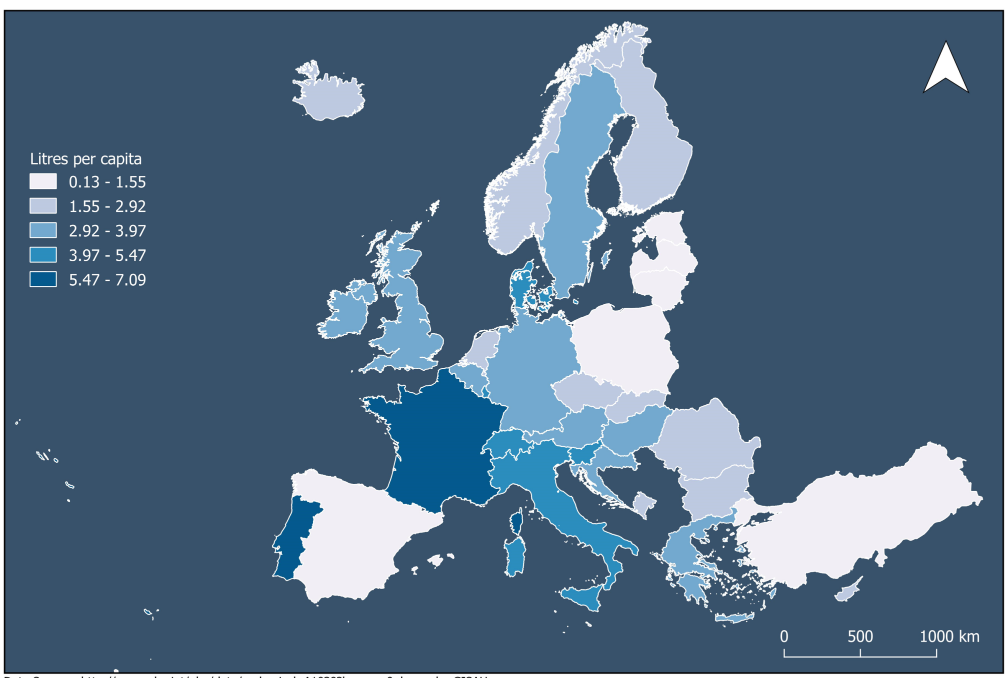
Emily Nagler

January 4, 2019

PART 1

The generated choropleth maps demonstrate two different software processes to create a visualization of the same basic quantitative data from the Global Information System on Alcohol and Health (Global 2018). For this comparative analysis QGIS was chosen as the GUI system and RStudio as the code-based system. Both types of software have similar workflows in theory, but the characteristics and processes come with various benefits and drawbacks between the two. Overall the GUI system is much more user-friendly for beginners and fairly intuitive after using the features only once or twice (Command 2018). Since most tools and commands are already built into the program, once it is known what they can do the process of getting through the steps of creating a map becomes very quick. In this case, after the chosen datasets had been joined with shapefiles to create a map, it became fast and easy to alter the cartographic features towards my final product. This form of direct manipulation is what GUI systems do well and it makes practicing good cartographic skills straightforward for inexperienced users (GUI 2018).

PER CAPITA WINE CONSUMPTION IN EUROPE (2015)



Data Source: http://apps.who.int/gho/data/node.gisah.A1039?lang=en&showonly=GISAH

Figure 1: QGIS Map

Command line interface usability is not as intuitive for a user that has little to no experience with scripting, which was the case for myself when trying to create the second map. The workflow within RStudio assumes that the user is familiar with standard code composition, which is the biggest drawback for beginners. If users are not aware of the proper syntax and tools necessary to create their desired product, this process is more time consuming to get to a finished product (Computer 2018). I also found it to be quite slow when making simple tweaks to the standard map features in R, since the whole code must be re-run each time an adjustment is made. In terms of ease when switching between different types of maps such as interactive, static, or multiples just to name a few, once the specified files are created, libraries downloaded and basic commands written it is very quick. However, It takes much more digging to find the right packages and tools within these packages require specific conditions to be met, all adding up to more work done by user. However, this sort of system is also much better suited to larger amounts of data and repetitive, complex tasks that can be sorted through with just a few bits of code (Soft 2018).

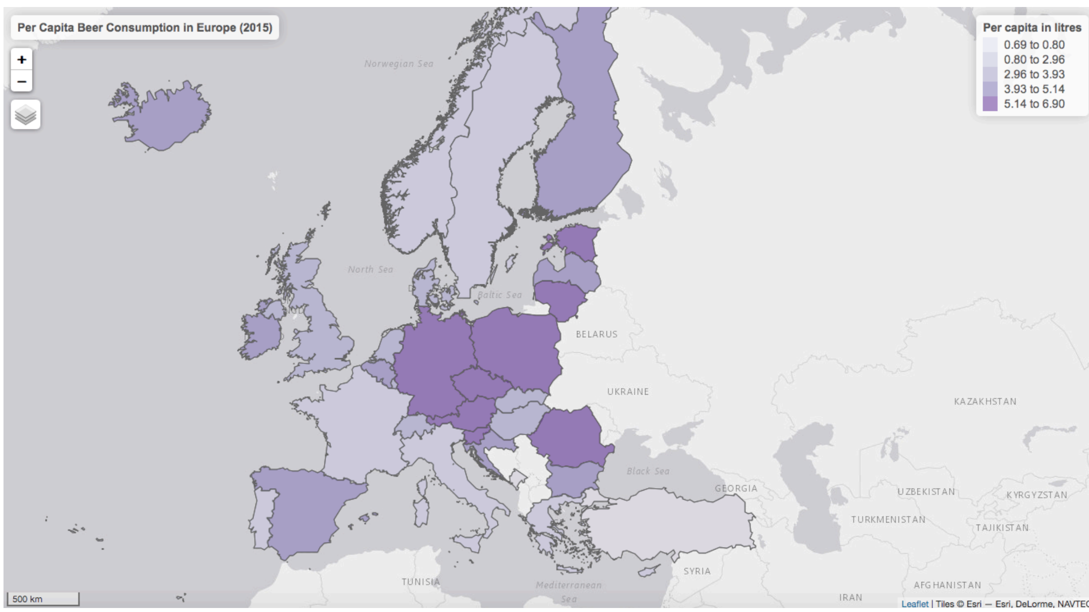


Figure 3: Leaflet Map

In my own experience, I found my map in QGIS to manifest much more quickly than in R, for the simple reason that I have slightly more experience with GUI systems. However, once I had created a static map in RStudio I was able to easily transform that into an interactive version and in my opinion this was a more interesting way of displaying my findings, even though I felt I was able to more easily stylize my map in QGIS. As far as cartographic practice goes, I felt that my QGIS map represented a 'good' map, with all the proper cartographic features such as orientation, scale and an intuitive color palette (Hennig 2016). Even though I preferred the interactive map form of data presentation in R, I felt restricted when adding the same cartographic features. In the context of creating a simple choropleth map, for my experience level it was cleaner to do so with a GUI-based systems. However, the ability to do more with the same basic data and shapefile by changing only a few lines of code is a much more intriguing way of representing one's findings and possibly more efficient when complex computations are involved. **wordcount: 597**

PART 2

In this short analysis we chose to use ArcMap, since the necessary packages are already built in to give a quicker output. To start, a base layer compromised of the London ward boundaries shapefile is uploaded and while the 'Editor' toolbar is active the small ward boundaries within The City of London can be dissolved into one polygon with the 'Merge' tool. After this ward data from the UK Data Service can be joined to the boundaries and then exported as a new geodatabase feature class. By working with a geodatabase there are less restrictive editing locks, making this ideal for our analysis (What 2018). Now we can import the treasure hunt route shapefile into our new geodatabase along with the London tube and rail stations file converted with the 'KML to Layer' tool. Our final layer needed is the treasure hunt locations, which can be converted to an XY layer using the 'Add XY Data' tool, and then projected onto our map use the 'Convert Coordinate System' tool. With the addition of each of these new layers a 'Repair Geometry' tool is used to identify and fix any geometry problems within the feature class before further use. For the first question we can find the length of the route by populating an empty field within the attribute table using 'Calculate Geometry', giving a vector length of 46598 meters.

Figure 3: Question 1 Map

For the second question a 100 meter buffer is created around the route using the 'Buffer' tool. Then the tube and rail station layer with the buffered route is input into the 'Spatial Joins' tool, giving a new output layer highlighting 24 stations. The use of 'Spatial Joins' was chosen over the 'Intersect' tool due to potential point duplication when overlapping more than one polygon (Patterson 2016).

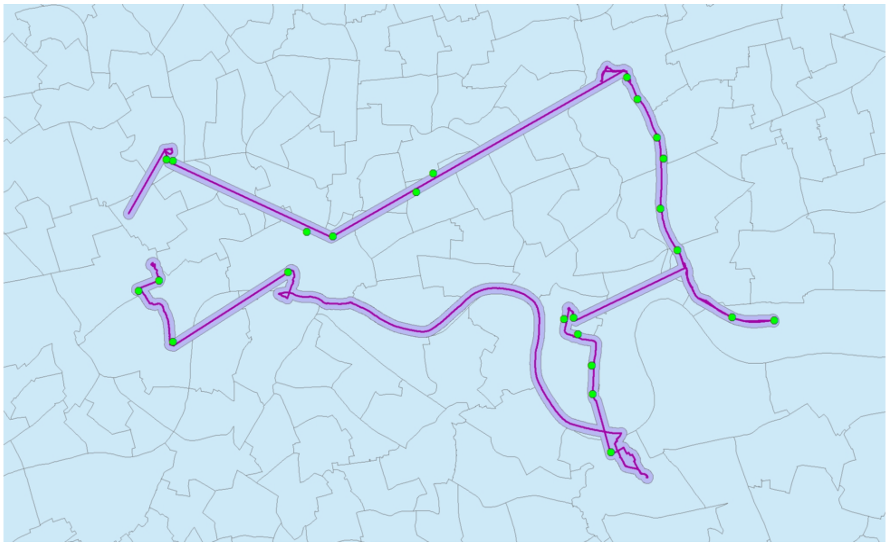


Figure 4: Question 2 Map

In the third question a 300 meter buffer around each treasure hunt location is generated, again with the 'Buffer' tool. Then using 'Spatial Joins', this point buffer layer and our original treasure hunt line give a new layer. By summing the 'Points' field within the new attribute table we get a total of 115 points.

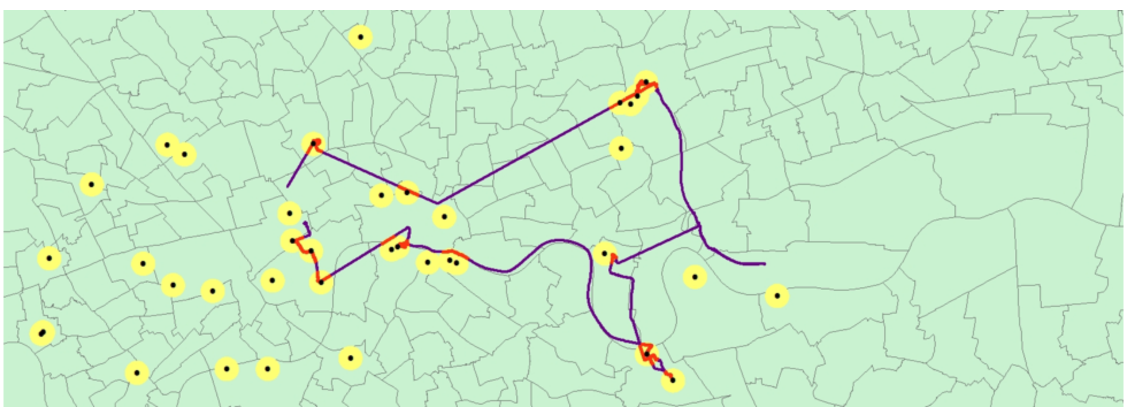


Figure 5: Question 3 Map

For the fourth question we chose to use the 'Intersect' tool on the route and the ward boundaries since there is no chance for duplicated results with these vector classes. Within the attribute table of the output layer we can sort the 'Male Life Expectancy' field in descending order to find that of the wards crossed the highest value is The City of London and the lowest is both Bethnal Green South and Weavers.

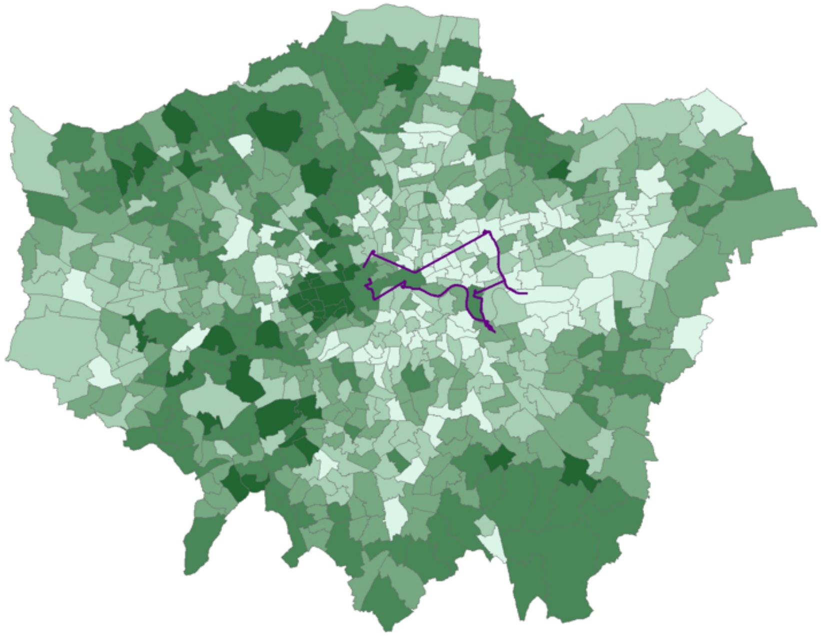


Figure 6: Question 4 Map

From the layer created in the fourth question we can then answer the fifth by using summary statistics within the attribute table to find that the mean value for life expectancy at birth in males is 80 and in females 83. For question six the 'Spatial Autocorrelation' tool was used in order to determine a point pattern for the treasure hunt locations. Before running this tool the coordinate system of the treasure hunt locations must be changed from GCS to PCS using the 'Project' tool. Then running the analysis, Global Moran's I attributes values to feature locations and we can determine from the report that there is in fact no spatial point pattern.

Global Moran's I Summary

Moran's Index:	0.020436
Expected Index:	-0.020408
Variance:	0.000333
z-score:	2.238089
p-value:	0.025215

Figure 7: Spatial Autocorrelation Summary

Uncertainties in the results achieved can be attributed mainly to the different coordinate systems between the layers. The London ward boundaries and subsequent geodatabase is projected with the OSBG 1936 system, whereas all of the other layers are in the WGS 1984 system. While there is no visual evidence of this difference, it could affect certain point and line locations very slightly. **wordcount: 597**

PART 3

Introduction

Within recent years there has been an ever-increasing movement towards health and sustainability within cities, and as a form of urban regeneration many cities have started funding gardening projects within various neighborhoods as a way to increase overall livability for their residents (Obordo 2018). However growing produce within city limits comes with downsides, one of the biggest problems being soil quality and contamination. Many possible garden locations within cities can have poor soil conditions either from current or past land use that creates an unproductive growth environment and more seriously has the potential to transfer harmful chemicals to produce grown in these areas (Reusing 2011). The process of identifying an unfit area is possible, but not entirely obvious since there is a lack of proper guidelines to follow. Therefore, this project will aim to address this issue by creating a tool that can aid cities and communities with better decision making when it comes to allocation of empty land for urban gardening uses.

Literature Review

The emergence of urban gardening can be defined as an attempt to combat what is described as a 'food desert', or location with a lack of fresh produce that is easily accessible and affordable to local residents (Gibson 2017). With more city inhabitants pushing for an increase in urban farms and community gardens, addressing soil quality is the next step that needs to be taken in order to ensure the safe growth of crops. There are many ways that soil can become contaminated within cities, mainly industrial and construction activity lead to heavy concentrations of lead, copper, and zinc to name a few (Galitskova 2016). London funded city-wide soil sampling for lead content from 2005 till 2009 and found that land use was the single greatest identifier of variation in lead found (Lark 2013). Similarly, some groups such as the James Hutton Institute have tried to create an extensive soil quality map for in this case, all of Scotland (Prager 2014). However, most cities don't have an extensive understanding of areas at risk since it is expensive and time consuming to produce surveys, yet we do know what kinds of industries and land use practices can produce these types of heavy metals (Lark 2013). Many cities such as Toronto and governmental organizations such as the EPA have created their own guides to inform the public on safe urban gardening practices with similar parameters of highlighting land use (From 2013, Superfund 2011). Nevertheless, sources of contamination seem to be hard to define universally and often the level of severity can vary over the years with land use change.

Therefore, instead of mapping locations where heavy metals are found today or in the past, we can look at current land use and locations of specific features within a city that are known sources of soil contamination. In order to provide this spatial information within cities we will create an interactive app that will display a map of the urban area with different levels of contamination severity. Most reports are aimed at one city in particular, however these issues can be applied to almost any city that wishes to convert vacant space for gardening. Thus, instead of focusing on a single place this tool will have the capability to be applied to any city input by the user, making it universally useful while promoting a defined set of contamination levels to be used by all. To achieve this we will first discuss our data source, then define index levels of soil quality, and finally develop the interactive map. This process will be carried out using RStudio, since we want to create a shiny app that can be accessed over the web by any city that wants to make more informed decisions in terms of safe urban garden locations.

Data

For this analysis we have chosen three levels of soil contamination risk: low, medium, and high. These categories were defined based on the level of concern guidelines from the Toronto Public Health guide for soil testing in urban areas (From 2013). Our spatial data for this tool will be sourced from OpenStreetMap (OSM). We chose this platform since the main package we will use to create the spatial data frames for our index levels, 'osmdata', draws from OSM (Padgham 2018). This package was chosen above other OSM packages due to the ability it has to download vector data rather than raster tiles (Padgham, 2018). Another essential packages used for this analysis is the 'raster' package, this was chosen due to its ability to bind and buffer data frames either in vector or raster form (Hijmans 2018). The two final packages employed in this analysis are 'tmap' and 'leaflet', which were chosen because they can create an interactive map and output the results on the shiny app dashboard. Using the OSM map glossary we identified three features to be included in each index category with their respective key and element tags, which will be the starting point in order to build queries for data extraction.

Level	Key	Value
LOW	Building	Residential
LOW	Leisure	Park
LOW	Amenity	School
MEDIUM	Building	Commercial
MEDIUM	Highway	Motorway
MEDIUM	Power	Pole
HIGH	Building	Industrial
HIGH	Amenity	Fuel
HIGH	Railway	Rail

Figure 8: Index Table

Methods

The process to create this interactive application can be broken down into two steps, in the first we will define and create our index layers using the 'osmdata' and 'raster' packages. In the second step we will add this code into our app template to output the final map using the 'shiny' and 'leaflet' packages. For step one we must begin by running an overpass API in order to select customs parts of OSM map data for our different levels (Padgham 2018). This function requires three conditions: a bounding box, a key and an element. We will define a bounding box later with user input, but for now we can add the keys and elements. These features are then passed to queries that can be further transformed for various uses, in our case we will convert them to spatial data frames. Once all our features are in this format we can then call a specific vector class for the spatial information to be displayed in, either a point, line, or polygon. It is important to note that all features within an index level need to have the same vector class in order to be bound together, in our case we used points for all features. Next, the three features within each level are combined using the 'bind' function in the 'raster' package to create one large spatial points data frame. To finish creating our layer we will add a buffer to each bound data frame also from within the 'raster' package. The buffer is based on the recommended distance to plant from each feature (From 2013). For low we set the buffer to 50 meters, for medium 30 and for high 100. The low buffer is higher than the medium because it refers to being able to plant within the 50 meter buffer as a safe zone, whereas the other two must be outside of the buffers. After repeating these steps for each index level we have three large spatial polygons that are ready to use in our app.

In step two we begin by setting up our ui where we will create the dashboard layout and define where the inputs and outputs will be shown. The sidebar panel includes the text input box and the main panel includes the leaflet output map. Next we create the server function where we can copy our code from step one and use this as our leaflet output. Within our previous code we can now define a bounding box within the initial queries for each level by calling the text input by the user. After the layers are developed we can then set the tmap mode to interactive and add them to our leaflet map with a legend and corresponding colors. To finish, our script gets the name 'app.R' and we can then publish to shinyapps.io so making this work accessible to anyone as opposed to running locally on a single computer.

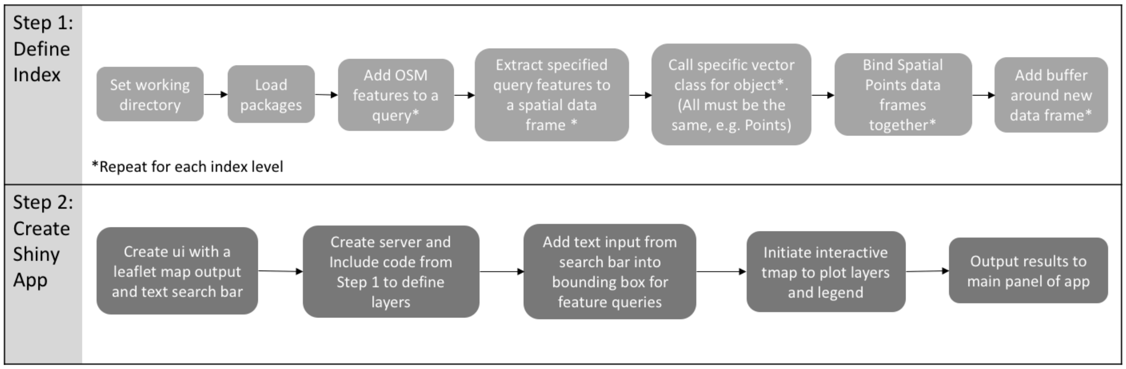


Figure 9: Methods Flow Chart

Results

Based on the outlined indices we can see that for the city chosen by the user three different buffer layers overlap to show the contamination severity. With the interactive leaflet map that is output the user can then zoom in on a particular location of interest within the greater bounding box. This resulting interactive shiny app can be accessed at: https://emilynagler.shinyapps.io/ucl_gis_app/.

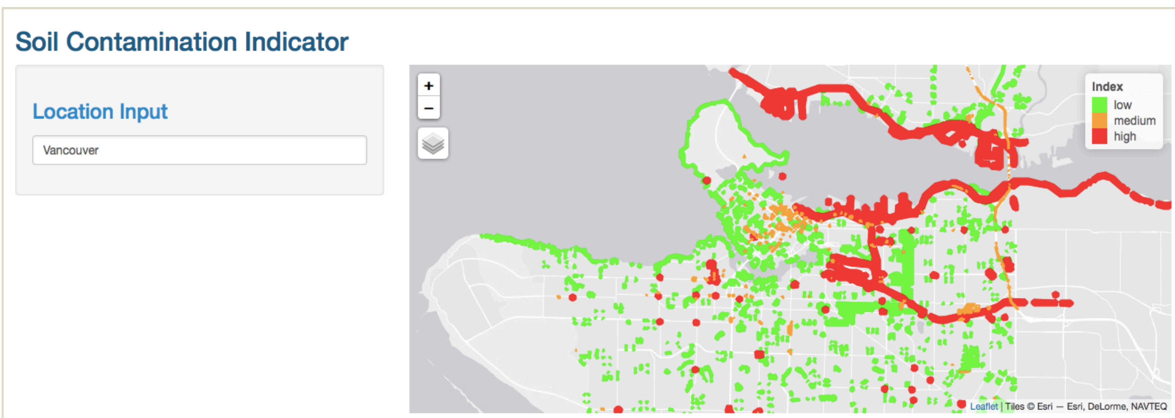


Figure 10: Shiny App Output of Vancouver

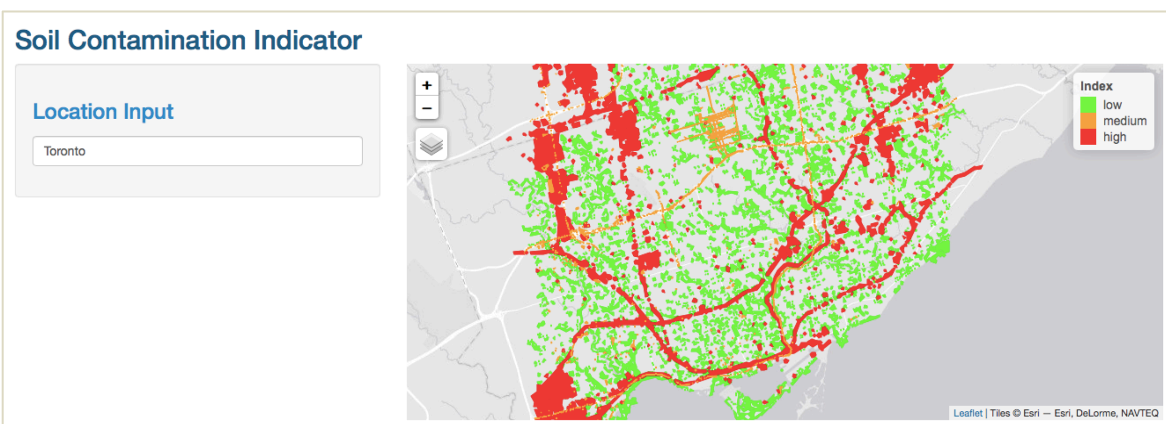


Figure 11: Shiny App Output of Toronto

Discussion

Our results show that we can indeed map a city for different soil contamination zones, however there are a few limitations and risks in the output that must be addressed. The limitations of this tool primarily come from the usage of OpenStreetMap, since it is a community driven tool and draws upon input of spatial knowledge by volunteers (Mandeno 2017). While it is very good for providing free data, the concentration of quality can range between locations throughout the world, giving less thorough results depending on the city we input (Mandeno 2017). Another limitation comes from our chosen indices and the amount of grey space or unknown areas within a chosen city that can be seen on the map output. This means that the map doesn't have an answer for every square inch of the city, however this leaves room for further development that could aim to cover every part of the defined bounding box and place it within an index level. Our assumptions in terms of the defining features for each category while based on literature, is still subjective and therefore also needs to be taken into account. It is also important to note that these index levels are just an estimation, this is a tool that can be used as a starting point for further investigation into a specific location within a city. If a space is located in a red zone it doesn't necessarily mean it is unusable, but merely that there should be further on-site soil testing to determine if it is in fact unsafe.

Conclusion

While this app does have much more room for further development, it provides a good starting point that can be built on with added complexity and features for future use. In the end we can get a general idea of the different contamination zones within a city of our choice, and see this projected on a simple map. The hope is to make this type of information on soil contamination more accessible for any group interested in starting their own garden in whatever space they want, not just large, public open plots. In this way any communities or neighborhoods can participate in urban gardening without the help of larger groups or governmental organizations which often take time to get in contact with. By addressing this key issue of soil quality for urban gardening and keeping people informed with simple tools, it can become safer and easier than ever to add a healthy activity to neighborhoods anywhere in the world. **wordcount: 1800**

Sources

Command line vs. GUI., 2018. Computerhope [Online] Available at: https://www.computerhope.com/issues/ch000619.htm [Accessed 30 Oct. 2018].

From the Ground Up: Guide for soil testing in urban areas, 2013. [Online] Available at: https://www.ruaf.org/sites/default/files/Guide for soil testing in urban gardens.pdf [Accessed December 18, 2018].

Galitskova, Y.M., Murzayeva, A.I. 2016. Urban Soil Contamination, *Procedia Engineering*, 153, Pages 162-166, ISSN 1877-7058, https://doi.org/10.1016/j.proeng.2016.08.097.

Gibson, M. 2017. *History of urban farming, and its 21st century resurgence*. Zoom Out Mycology. Available at: https://www.zoomoutmycology.com/blog/repeating-history-urban-farming-in-america-is-back-and-here-to-stay [Accessed January 3, 2019].

Global Information System on Alcohol and Health (GISAH), Recorded alcohol per capita consumption, from 2010, 2018. World Health Organization. Available at: http://apps.who.int/gho/data/node.gisah.A1039?lang=en&showonly=GISAH [Accessed October 30, 2018].

GUI vs Command line interface, 2018. Softpanorama [Online]. Available at: http://www.softpanorama.org/OFM/gui_vs_command_line.shtml#Advantages_and_disadvantages_of_GUI_vs_command_line [Accessed 30 Oct. 2018].

Hennig, B. 2016. *Mapping Practices in a Digital World*. Available at: http://www.viewsoftheworld.net/data/Hennig_2016_AdvancingGIS_DigitalMapping.pdf [Accessed October 30, 2018].

Hijmans, R.J. 2018. *Geographic Data Analysis and Modeling: R package raster version 2.8-4*. The Comprehensive R Archive Network. Available at: https://cran.r-project.org/web/packages/raster/index.html [Accessed December 26, 2018].

Lark, R.M., Scheib, C. 2013. 'Land use and lead content in the soils of London', *Geoderma*, 209-210, p65-74, ISSN 0016-7061, [online]. Available at: https://doi.org/10.1016/j.geoderma.2013.06.004 [Accessed January 2, 2019].

Mandeno, E. 2017. Pros and Cons to OpenStreetMap. *Digital Mapping Solutions*. Available at: https://digitalmappingsolutions.com/2017/08/25/pros-and-cons-to-openstreetmap/ [Accessed January 3, 2019].

Obordo, R. 2018. 'Fresh, free and beautiful: the rise of urban gardening'. *The Guardian*. Available at: https://www.theguardian.com/world/2018/jun/07/fresh-free-and-beautiful-the-rise-of-urban-gardening [Accessed January 2, 2019].

Padgham, M., Lovelace, R. 2018. osmdata. *The Comprehensive R Archive Network*. Available at: https://cran.r-project.org/web/packages/osmdata/vignettes/osmdata.html#4_recursive_searching [Accessed December 18, 2019].

Patterson, D. 2016. *How to Prevent Intersect from Duplicating* [Online]. Use the Five-Step GIS Analysis Process | GeoNet. Available at: https://community.esri.com/thread/174319 [Accessed December 7, 2018].

Prager, K., McKee, A. 2014. Use and awareness of soil data and information among local authorities, farmers and estate managers [Online]. Available at: http://www.hutton.ac.uk/research/themes/realising-lands-potential/land-manager-attitudes-and- behaviours

REUSING POTENTIALLY CONTAMINATED LANDSCAPES: Growing Gardens in Urban Soils, 2011. Superfund Remediation and Technology Innovation. U.S. Environmental Protection Agency. Available at: https://www.epa.gov/sites/production/files/2014-03/documents/urban_gardening_fina_fact_sheet.pdf [Accessed December 26, 2018].

What are the advantages of file geodatabases?, 2018. Knowledge Base. Available at: https://kb.iu.edu/d/axid [Accessed December 4, 2018].