5-2020

'LMshapemaker': Utilizing the 'Rmapshaper' R Package to Modify Shapefiles for Use in Linked Micromap Plots

Braden D. Probst
Utah State University

‘LMSHAPEMAKER’: UTILIZING THE ‘RMAPSHAPER’ R PACKAGE TO MODIFY SHAPEFILES FOR USE IN LINKED MICROMAP PLOTS

by

Braden D. Probst

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Statistics

Approved:

Jürgen Symanzik, Ph.D. Major Professor	Adele Cutler, Ph.D. Committee Member
Robert Gillies, Ph.D. Committee Member	Richard S. Inouye, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2020

ABSTRACT

‘LMshapemaker’: Utilizing the ‘rmapshaper’ R Package to Modify Shapefiles for Use in Linked Micromap Plots

by

Braden D. Probst, Master of Science

Utah State University, 2020

Major Professor: Jürgen Symanzik, Ph.D.
Department: Mathematics and Statistics

Linked micromap plots (LM plots) allow viewing spatial and statistical data simultaneously. They are helpful in detecting spatial trends that may be hidden in the statistical variables. Freely accessible boundaries from various sources are usually not ideal to be used directly in LM plots and have to be modified. Specifically, the boundaries have to be simplified, some regions have to be resized and shifted in order to be meaningful when used in the final LM plot. Through these kinds of modifications, users can make the maps in LM plots more readable and interpretable. This simplification and modification process has been done in the past for several countries, including but not limited to, the United States, France, Germany, Scotland, China, Korea, Argentina, and Brazil. The resulting modified boundaries are spread out in R packages, various publications, and on web sites. This thesis will discuss how to modify shapefiles for most countries in the world towards future use in LM plots. These modifications to the boundaries are conducted via Mapshaper, specifically through the ‘rmapshaper’ R package and compiled in a single archive under an R package titled, ‘LMshapemaker’. In addition to the archive of recommended modifications provided in the ‘LMshapemaker’ R package, a Shiny app is included allowing users to further modify boundary files in a manner more meaningful to the user.

(248 pages)

PUBLIC ABSTRACT

‘LMshapemaker’: Utilizing the ‘rmapshaper’ R Package to Modify Shapefiles for Use in Linked Micromap Plots

Braden D. Probst

In order to effectively create map-based visualizations, some map modifications need to be conducted to ensure the map is readable and interpretable. There are several issues that need to be addressed to achieve this. The boundaries of a country may be overly complex which is particularly true with coastal areas of countries. Regions may be small and not seen in the final plot, as is the case with many capital cities in the world’s countries such as Washington D.C. and the Federal District of Mexico City. In other countries, regions may geographically lie far away from the rest of the country as is the case with the Galàpagos Islands in Ecuador and the state of Hawaii in the United States. To best use maps with these issues in visualizations, users need to modify the boundaries and regions using software, such as arcGIS and R. Modifications have been done in the past for many countries, including the United States, France, Germany, Scotland, China, Korea, Argentina, Brazil, and others. Before the ‘LMshapemaker’ R software package, there was no single location containing usable maps. Access to the modified boundaries for the countries previously mentioned are spread out across different sources and R packages. ‘LMshapemaker’ is a new R package developed in this thesis that not only provides ready-to-use boundaries for a majority of the countries in the world, but also provides an accessible approach for users to modify boundaries to their preferences through a Shiny R app.

This thesis is dedicated to my wife, Kiana, and my children, Boston and Talia.

ACKNOWLEDGMENTS

To my family, for without their support and inspiration none of this would have been possible. They give me purpose and something to look forward to seeing every day.

I am deeply grateful to my advisor, Jürgen Symanzik, for his mentorship, guidance, and his enthusiasm for understanding data visualization principles. I am grateful that he believed in this research and believed in me.

With special mention to Adele Cutler and Robert Gillies for the time they put in serving on my thesis committee and whose input made this research more meaningful.

Braden D. Probst

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Background

Simultaneously visualizing geographic spatial data and statistical data can be a challenge. While there are many approaches to visualize statistical data, the visualization of geographic data is limited due to the requirement in many cases to provide a representation of the geographic layout in a meaningful way.

1.1.1 What are Linked Micromap Plots?

Linked micromap (LM) plots were first introduced in 1996 at the American Statistical Associations annual meeting (Olsen et al. (1996), Carr and Pierson (1996)). Over the last two decades, LM plots have been developed as a way to visualize potential trends within some spatial geographic data and within some associated statistical variable(s).

LM plots are a versatile plot and are useful for a variety of geographic regions. They can use any data that contains a spatial aspect to it. This makes them useful for plotting environmental, agricultural, medical, health, or economical data. As this is the case, government agencies specializing in these fields use LM plots for reporting their data. This includes agencies in the U.S., such as the U.S. Department of Agriculture (USDA), the National Cancer Institute (NCI) (Wang et al. (2002), Carr et al. (2002)) and the Environmental Protection Agency (EPA) (Rosenbaum et al. (1999)). LM plots have been used for different geographic regions at many different levels. They can be used to compare countries to each other in a large region, such as Europe. They can be used to compare administrative regions in a single country to others, such as comparing the US states to each other. They could also be used at a smaller scale by comparing counties in a given state to other counties. Other uses involve non-political boundaries, such as ecological regions and other geographic partitions (McManus et al. (2016)).

The basic structure of LM plots is an array of several vertical panels with the underlying data being oriented by row, as shown in Figure 1.1. Typically, although not exclusively, the leftmost panel is used to plot spatial data in the form of a map of the geographic regions of interest, along with the labels and names of these regions in panels 2 and 3. One or more columns of statistical data corresponding to the regions in the spatial panel are included in the subsequent panels (panels 4 and 5 in Figure 1.1). These panels plotting the statistical variable can take on a variety of forms, such as box plots and confidence intervals although dot plots and line plots are the most common choices.

The entire LM plot is then sorted by one of the variables that were included. For the sake of aesthetics, sorting is typically done by the first (leftmost) of the statistical panels. When sorted, the micromap plot adjusts all of the rows according the specific variable and orientation of the sorting.

Interpreting the LM plot is easily accessible to even those without a strong statistical background. As the data is row-oriented, a reader can see all of the data for a given region on a single row. The LM plot is typically broken into perceptual groups of five or less regions so on any given map in the LM plot, only five of the regions are filled in with color. The color assigned to each state is unique within each perceptual group and is similarly colored in the statistical panels. This allows a reader to compare any region's statistics to those that are similar, but also compare one grouping of regions to those that are higher or lower in their measure of the selected sorting variable.

The complete LM plot shows whether there is any association among the selected statistical variables and whether or not the data is also spatially correlated, as reflected in the filled in maps found within the spatial panel.

1.1.2 Limited Access to Usable Shapefiles

Shapefiles for various countries and other regions of interest are readily available through the internet using such resources as the Database of Global Administrative Areas, or GADM (Global Administrative Areas (2012)). A shapefile is a collection of files that store the

Fig. 1.1: Example of a multi-panel linked micromap plot for the 29 counties of Utah. For the purpose of demonstration, variables 1 and 2 are arbitrary numbers assigned to each country. Variable 1 is a randomly assigned index from 1 to 29 and variable 2 is a random number between 1 and 9.

information necessary for mapping a given geographic feature. The information that defines a map are the points and lines that make up the boundaries and the coordinates of these features.

Various methods existn for modifying and simplifying shapefiles in R (R Core Team (2019)). This has been done for several countries and regions in the world. In many of these cases, changes had to be made to the shapefiles to allow for meaningful plotting of the maps in the LM plot. Finding simplified, ready-to-use boundaries for a given country poses another challenge as the simplified files that do exist for public use are scattered through various R packages or locations on the web. Before now, there was no single archive that provided access to ready-to-use shapefiles for the creation of LM plots.

1.1.3 Motivation

Wide access to shapefiles does not solve the problem for an individual wanting to create LM plots. Plotting raw shapefiles, as found on GADM, can be a time consuming and computationally expensive process. Plotting even a single shapefile accurately can take time. This problem is compounded even further, given that LM plots require plots of several repeated shapefiles within the spatial panel of the plot. This problem can and will be addressed by 'thinning' the number of points included in a given set of shapefiles. Rather than plotting all of the points in a geographically accurate representation of a region, we can remove a specified proportion of boundary points to plot much less, while still retaining the overall shape of the region of interest, at the cost of a perfect geographic representation.

We are sacrificing the geographic accuracy of a region, however, in the context of LM plots many of the polygons that do exist in such a shapefile carry no value in interpretation when filled with color. One reason for this could be due to the small size of a given polygon. When a sufficiently small polygon is filled with color it is not distinguishable from surrounding areas. Another reason for a given polygon not being interpretable in LM plots is due to the fact that the polygon is so small that it is entirely masked by the larger surrounding regions.

These issues, in some cases, will be fixed by thinning the polygon and simplifying the boundaries. However, it is more likely that further modifications will be required to get a given shapefile into a format that can easily be read and interpreted. Possible modifications are enlarging or exaggerating regions or moving them entirely to be easier seen in comparison with the remaining regions.

In summary, as we attempt to address problems in using a shapefile in LM plots, it is beneficial to consider geography as a feature that is malleable. In order to obtain a shapefile that will produce the most meaningful LM plot we will need to, in some cases, considerably change what a given country or region looks like while retaining the overall shape of that region.

1.2 Overview

The purpose of this thesis is to provide more accessibility to obtain the shapefiles needed for a user to create linked micromap plots for a given country of interest. As stated above, raw shapefiles, as found on GADM, are often not suitable for immediate use and the files need to be prepared in order for the LM plots to be more meaningful.

In Chapter 2, we will review the previous approaches that were taken to modify shapefiles and create LM plots. As new technology has become available, we have more tools available that allow a simpler, modern approach. These specific advancements are detailed in Chapter 3.

Chapter 4 will provide details on the specific modifications that have been performed for a large majority of the countries in the world. Details for the 'LMshapemaker' R package containing these shapefiles and their supporting documentation can be found in Chapter 5.

It is to be understood that the shapefiles that have been created were done with a subjective perspective on what was found to be a meaningful representation. Some readers may have a better understanding of a given region and should be able to either modify the provided recommendations, or modify their own shapefile to fit the exact specifications that they desire. Therefore, an application through R Shiny that allows a more approachable method for modifying and simplifying shapefiles will be introduced in Chapter 6.

Examples showing the process of modifying shapefiles and then being used in LM plots with real data will be given in Chapter 7. In Chapter 8, we will discuss conclusions to the research detailed in this thesis and what further work can be done.

The appendices will detail the recommended modifications for each of the shapefiles that were determined necessary for each country and the documentation for the functions included in the ‘LMshapemaker’ R package and R Shiny app. Visualizations for how the changes affect the regions and boundaries of these shapefiles will be provided in Appendix A along with tables or code documenting the modifications. The help pages for the ‘LMshapemaker’ R package and associated Shiny app will be detailed in Appendix B.

CHAPTER 2

Review of Past Work on LM Plots

Since their introduction in 1996, LM plots have been used to visualize statistical and map-based spatial data and address some of the shortcomings of choropleth maps (Olsen et al. (1996), Carr and Pierson (1996)). In more recent years, users of LM plots have been met with more complications specific to LM plots. Regions may be small and unreadable in the context of a LM plot and countries with many regions can make LM plots challenging to read. In each of these cases, solutions exist.

2.1 Addressing Limitations of Choropleth Maps

The idea of tying statistical and spatial data together is not new to micromaps, however. One such plot is the choropleth map. While these can be meaningful in many cases, there are several advantages to using LM plots over the choropleth map. These advantages all stem from problems with how choropleth maps represent data (Symanzik and Carr (2008)). In a choropleth map, a statistical variable is translated into a color scale with each color representing a certain level or range of values in the statistical variable. A given region on the map is then filled with the color that represents that region's value, in terms of the statistical variable.

The first of these problems is that choropleth maps overemphasize the area of large geographic sub-regions and in many regions there are small sub-regions that will not show the plotted color well. Examples include several countries whose capitals are small geographic regions compared to the whole, such as Washington D.C. in the United States or the capital region of Littoral in Benin. These problems are addressed by scaling regions in different ways that either make regions more visible (Monmonier (1993)) or make the area reflect another factor, such as population (Dorling (1995)). In either case, a reader may end up with regions that are no longer recognizable.

The second problem with choropleth maps is that by converting continuous variables into discrete classes, information is lost. By translating a variable into a discrete color scale, the unique values found in each of the regions are no longer distinguishable and many will be plotted with the same color when filling in the choropleth map. Balancing the number of classes with the readability of the plot is a challenge, however, researchers have found that even using the most advantageous class selection leads to loss in information (Brewer and Pickle (2002)).

The last of these problems associated with choropleth maps is that showing more than one variable is challenging. Attempts have been made to use bivariate color schemes for representing two variables on a single map, but the results were not very successful (Wainer and Francolini (1980)). The solution would be to display two choropleth maps side-by-side with each one representing a different variable. This introduces another issue, however, large areas that are similar can mask the areas of interest where the differences lie. This phenomenon is known as change blindness (Monmonier (1996)).

There are several features of LM plots that help address the problems listed above. The first feature is that the focus of LM plots is on statistical variables rather than the maps. This, in part, allows us to use the unique values of a statistical variable and not convert them into color classes. Another feature used by LM plots is small multiples. Rather than representing the geographic region as one large boundary as in choropleth maps, LM plots use smaller scale multiples of the same region. The benefit here is rather than having all of the data in a single bounded region, it is spread into several smaller and comparable maps. Lastly, in LM plots a user can easily include and compare many variables.

2.2 Maintaining Readability and Interpretability in LM plots

Two notable situations have caused issues with the readability and intepretability of LM plots in the past. These issues both result from using small multiples for representing the maps within the spatial panel of the plot.

2.2.1 Issues with Region Size

The first situation that causes problems is the comparative size of sub-regions within the shapefile. Many countries in the world have small administrative divisions. Capital districts, such as Washington D.C., the Federal District of Mexico City, and the Federal District in Brazil are examples of such regions. When plotting countries that include small sub-regions, the regions become indistinguishable from the surrounding regions.

In order to fix this issue, the small areas need to be enlarged. There are different ways to address the enlargement, each with their own strengths. Different approaches to enlarging areas are detailed in Symanzik et al. (2014), in which two different results are presented showing how to fix this issue for the Federal District of Brazil and Symanzik et al. (2016), in which several regions in China needed to be enlarged. Further details on the process of enlarging regions are provided in later chapters.

2.2.2 Issues with Region Count

Another factor to consider when creating LM plots is the region count in a shapefile. Determining how to partition the regions in a LM plot can be a challenge, particularly for regions that contain many sub-regions. While recommendations for how to partition regions based on the total region count are provided in Symanzik and Carr (2008), Table 2, a shapefile that contains high region counts start to lose interpretability in the context of LM plots because it requires more repetitions on the map within the map panel. While it is not a strict requirement to only include five regions within a single map, including more makes colors harder to distinguish for the reader. In order to create the most meaningful LM plot, a creator should minimize the number of map repetitions and optimize the number of regions in each comparison group.

In an online blog, `freerangestatistics.org`, Peter Ellis discussed his experience with how LM plots represent data for the island nation of New Zealand (Ellis (2017)). While Ellis pointed out several issues that were encountered, the biggest issue has to do with New Zealand having 66 sub-regions at the administrative level he was interested in. In Ellis' construction, there are 11 repetitions on the New Zealand map making the maps small and the regions indistinguishable. A similar issue was documented in Payton et al. (2015), page 10. In Figure 3 of the same publication, the authors presented a solution to plotting the 255 counties of Texas. Rather than plotting in a single LM plot, the data were plotted in a multi-panel LM plot where each panel corresponded to roughly a quarter of the original data.

A similar issue was encountered in preparing a shapefile for use in the JSM Data Expo Challenge of the American Statistical Association 2019. The purpose of this challenge is to look at census data within New York City. New York City is separated into five boroughs which are themselves broken into many sub-boroughs. Combinations of these sub-boroughs make up the 55 Public Use Microdata Areas (PUMAs) in New York City. Figure 2.1 shows a hypothetical LM plot for using the data in a single panel. In this figure, the 11 repetitions of the map combined with the small regions make distiguishing the regions a challenge.

A solution to this is to break the data into two groups, one with 28 of the PUMA districts, and the other with 27. We can then create micromaps for both groups and display them side-by-side. This configuration is shown in Figure 2.2. As seen in the plot, the number of map repetitions drops from 11 to 6 and the regions are much more visible.

2.2.3 Past Examples

In addition to the the New Zealand example introduced above from `freerangestatistics.com`, LM plots have recently been used in The Indian Story Report (Singapore Management University (2017)). The LM plot created was for users to explore graduation rates from the Indian population for different genders and for the population as a whole.

In the past, LM plots were also introduced for Germany (`http://stackoverflow.com/questions/21651985/shapefile-to-produce-a-linked-micromap-in-r`), France (Bonnal et al. (2011)), Korea (Ahn (2013) and Han et al. (2014)), Argentina and Brazil (Symanzik et al. (2014)), and China (Symanzik et al. (2016)).

Unfortunately, there exists no single web archive or R package for users to have access to modified shapefiles.

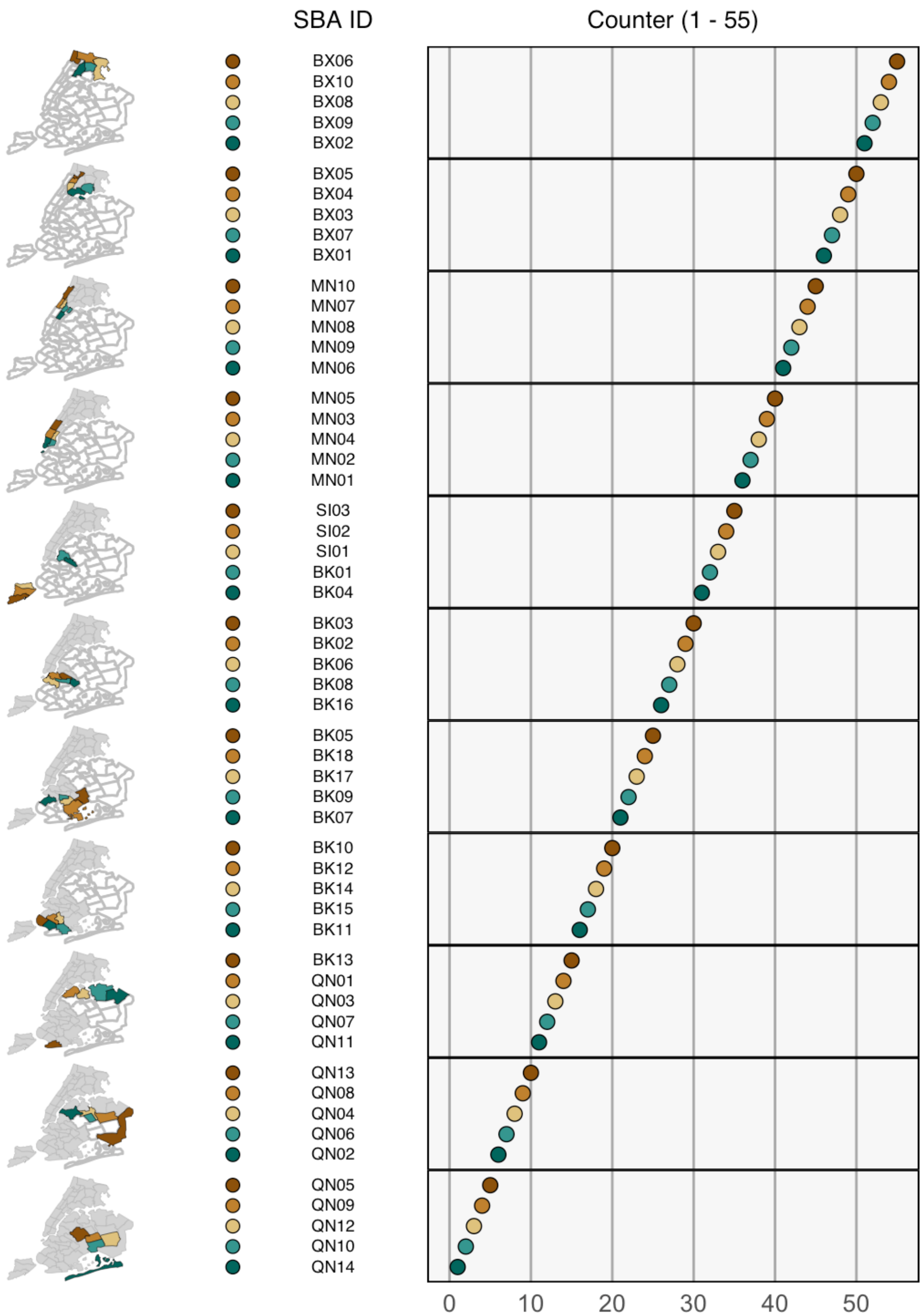


Fig. 2.1: Example of a single-panel linked micromap plot for the 55 PUMA regions of New York City. Boundary simplifications have been applied and several regions have been resized and shifted. By representing the 55 PUMA regions in a single panel, the regions are difficult to distinguish in the map panel.

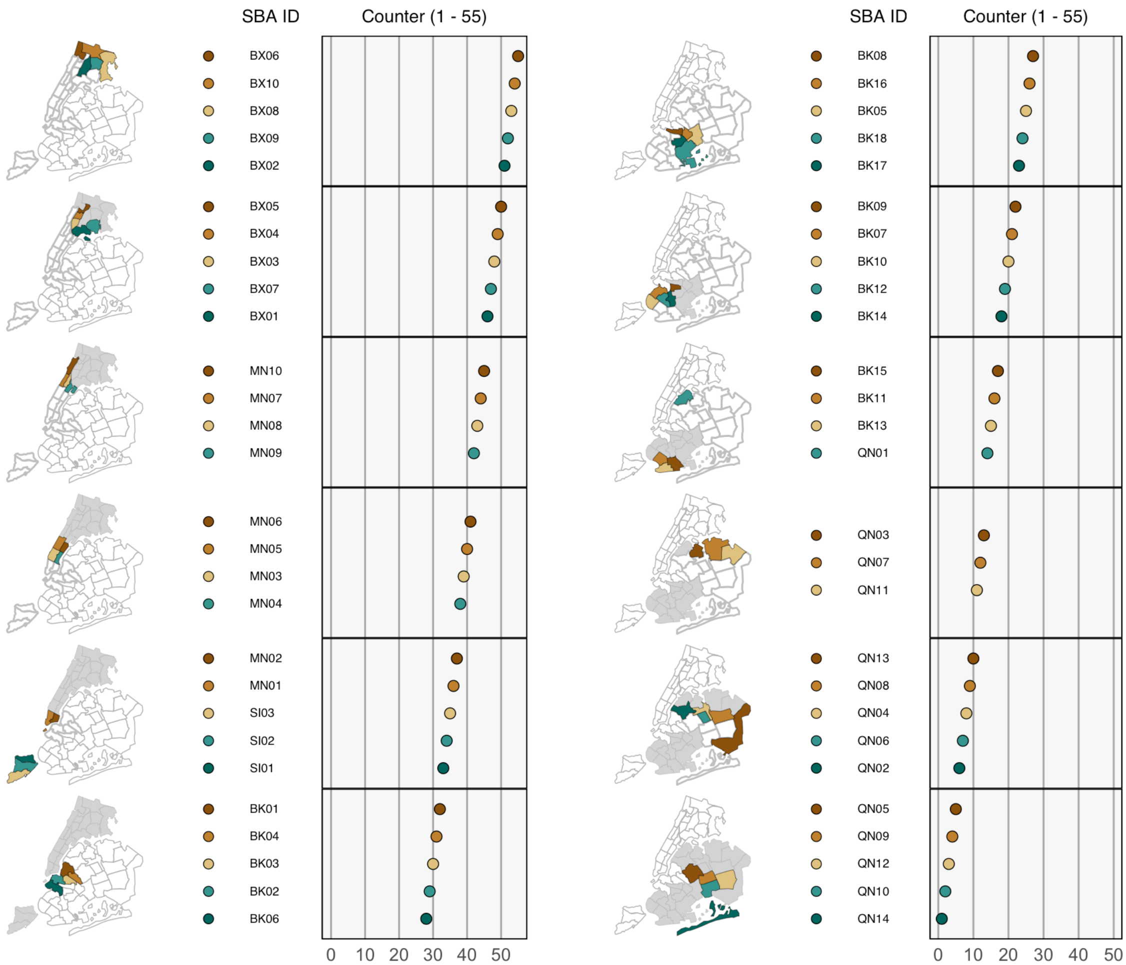


Fig. 2.2: Example of a multi-panel linked micromap plot for the 55 PUMA regions of New York City. Boundary simplifications have been applied and several regions have been resized and shifted. By splitting the data into two roughly equal sized groups (28 in the left LM plot and 27 in the right LM plot), the maps are much more visible and regions more distinguishable.

CHAPTER 3

Advantages of Mapshaper and the ‘rMapshaper’ R Package

Creating a meaningful linked micromap plot in R (R Core Team, 2019), in many cases, requires access to a modified shapefile with adjustments made to ensure readability and interpretability. Modified shapefiles for this specific use are not widely available. Modifications to a shapefile can be made externally in outside software or internally in R using various R packages as described in Symanzik et al. (2014).

In this chapter, we will introduce some software tools that are publicly available to assist in the process of modifying shapefiles. In addition to being available for public use, Mapshaper and the ‘rmapshaper’ R package, in particular, make the modification process more accessible for their users and can be simply done within the same R script that is used to create the LM plot.

3.1 Mapshaper

Mapshaper, and its web browser version, `mapshaper.org`, is software introduced to the public by Matthew Bloch and Mark Harrower in 2006 as an open-source tool to modify and simplify shapefiles (Harrower and Bloch, 2006).

The strength of Mapshaper is that it was the first, free program that provided a WYSIWYG (what you see is what you get) approach to simplify shapefiles for users that are not trained or prepared in using this type of software.

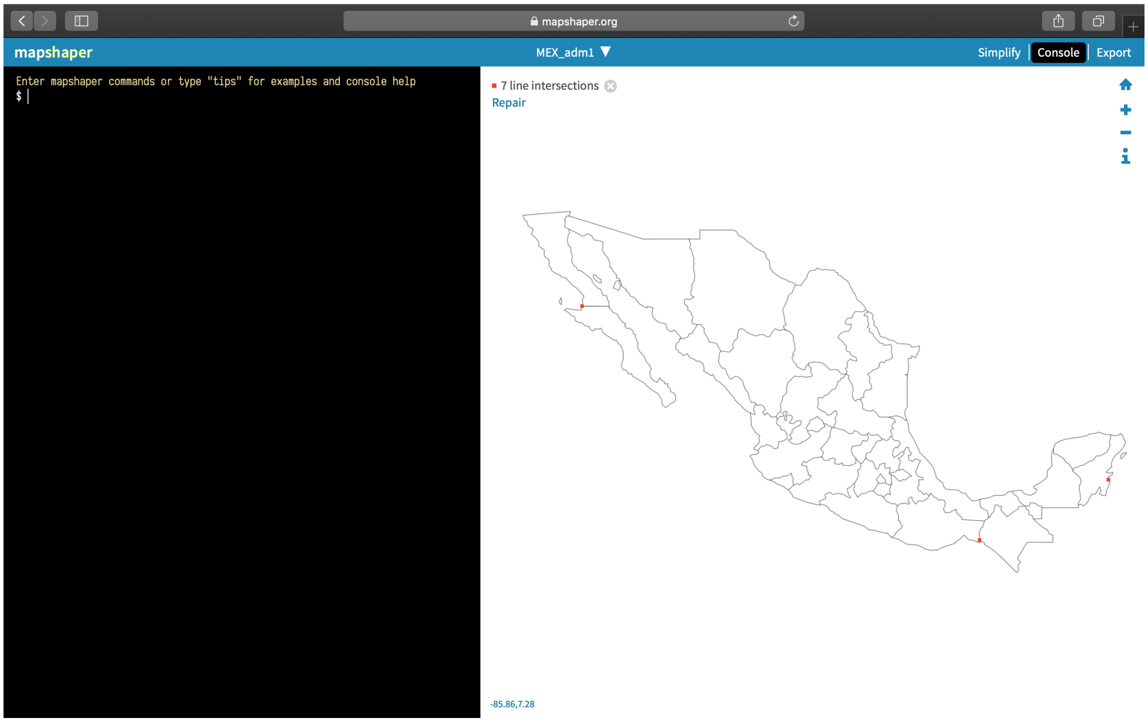


Fig. 3.1: Example of the web browser version of Mapshaper. The console for inputting commands is on the left. The near real-time WYSIWYG modifications are shown on the right. Here the country of Mexico is shown after the boundary lines have been thinned.

While most of the functionality of Mapshaper is done manually through a command line in the console (see Figure 3.1), the syntax of the commands is easy to understand even for beginners. Further, with a shapefile being updated in close to real-time, users are able to see exactly how each modification would appear in the final product.

Upon completion, a user could download a modified shapefile and upload it to R for further use.

3.2 'rmapshaper' Package

Andy Teucher, in 2016, brought even more accessibility to the modification and simplification of shapefiles when he released the 'rmapshaper' R package, (Teucher and Russell (2018)).

At its core, 'rmapshaper' is an R wrapper to the functionality provided within `Mapshaper.org`. As 'rmapshaper' is still fairly new, not every command in Mapshaper has been directly translated into a standalone R function. While 'rmapshaper' is still being updated to include more Mapshaper commands as R functions, Teucher has given access to the entirety of Mapshaper commands through the inclusion of the command *apply_mapshaper_commmad*(). While not necessarily ideal, this still allows a user to use all of the Mapshaper commands to tailor shapefiles to be exactly what is needed.

Historically, using Mapshaper would have required a user to modify their shapefiles prior to constructing the final LM plot. Now, thanks to the 'rmapshaper' R package, the modification and simplification of a given shapefile can be done within the same R script where the LM plot is created. Thus, when users are reviewing a LM plot before its inclusion in a presentation or publication and they notice that they would like to make some changes, they no longer have to externally prepare a separate file to upload. Instead, they may change the code immediately in the R script and see the changes as soon as they run the updated R code.

CHAPTER 4

Modifications of Boundaries and Regions

Several modifications should be considered to prepare a shapefile for meaningful plotting in linked micromap plots. Exterior and interior boundaries of the country and regions can be simplified. Regions, especially islands, can be shifted closer to the rest of the regions and very small or very large regions can be resized. Each of these modifications can improve the readability and interpretability of a LM plot.

4.1 Simplifying Boundaries

In order to understand why the process of thinning needs to be applied to a shapefile, one needs to understand how a shapefile is constructed. At first glance, a shapefile is simply the outline of some region, often a geographic or political boundary. These outlines are created using adjacent polygons, whose edges relate to the real-life boundaries of the given region. The boundaries of these polygons can be considered as a dense collection of points.

On simpler shapefiles that represent boundaries created by humans, thinning may be unnecessary, such as in many of the Midwest states in the United States. The state boundaries of states such as Utah, Wyoming, Colorado, etc. are linear at particular degrees latitude or longitude determined by the government at the time of their addition to the United States of America. Shapefiles get much more complicated when the administrative boundaries are created to follow the geographic features in the area. This is especially true in coastal regions where the boundaries tend to not follow a straight line. When plotting such regions, this can be a computationally intensive process. In order to simplify the shapefile and cut down on computing time in cases such as those mentioned above, thinning the boundary is often the first step.

(a) (b) (c)

Fig. 4.1: Simplification done for Mexico. (a) shows the raw shapefile provided by GADM. In (b) the modified shapefile contains 1% of the original points. The thinning of boundary points is particularly noticeable in the coastal regions. (c) shows further simplification by keeping 0.1% of the original points. As displayed, by thinning, the small islands in the Pacific Ocean have been removed from the map.

4.1.1 Thinning

The process of thinning is as intuitive as it sounds. If we think about the many individual lines that make up a shapefile to be a dense collection of points lying next to each other, then the process of thinning is the process of selecting a proportion of these points to keep in the final shapefile. The thinning function included in Mapshaper (Harrower and Bloch (2006)) is based of the Douglas-Peucker algorithm (Douglas and Peucker (1973)). In the 'rmapshaper' R package (Teucher and Russell (2018)), the Mapshaper command for thinning has been translated into an R function `ms_simplify(shape, keep = p)`. The user has to decide **p**, the proportion of points to keep in the shapefile. For example, if we were to apply this command with $\mathbf{p} = 0.001$, then the final product of this process will be a shapefile that contains 0.1% of the original points, shown in Figure 4.1 (c).

As an example of showing what changing the value of **p** does, Figure 4.1 shows three outputs of a shapefile of the administrative districts of Mexico with different values for the thinning process. Figure 4.1 (a) shows the unmodified shapefile as provided by GADM. Figure 4.1 (b) shows that when **p** is set to a value of 0.01, the resulting shapefile is simpler and contains 1% of the points that made up the original shapefile. In Figure 4.1 (c), **p** has been set as 0.001 and shows the resulting boundaries that contain only 0.1% of the original points. In the latter simplification, thinning has removed several of the smaller islands off the coast and made the boundaries between the different administrative districts more linear.

While it is sometimes the case that thinning is all that is needed to create an adequate shapefile for the creation of a meaningful LM plot, it is likely that additional modifications are needed.

4.1.2 Filtering Islands

In most cases, the step of applying the R function `ms_filter_islands()` would be redundant as the previous thinning step above would remove any regions that would otherwise be affected by this function. This step does exist as a valid option for cases where further thinning would remove regions that should be kept in the final shapefile. This option allows a user to retain more of the original geographically detailed shapefile, but still filter out small islands that are less relevant in the context of LM plots. It should be noted that the option to filter islands is not currently available within the 'LMshapemaker' R package or its attached Shiny R application. However, this is an option at the command level in R.

4.2 Shifting Regions

There are several cases where, even after thinning the boundaries, physically shifting regions around to be in a more meaningful location may be necessary. There are two cases of shifting regions in a shapefile that can improve the the readability and interpretability of LM plots using these shapefiles.

Each of the following modifications, detailed in Sections 4.2 and 4.3, use an 'rmapshaper' function call that allows access to the `mapshaper.org` command line. The actual command used in `mapshaper.org` is "-affine". As this has not been translated directly into its own standalone R function, we will use the function call:

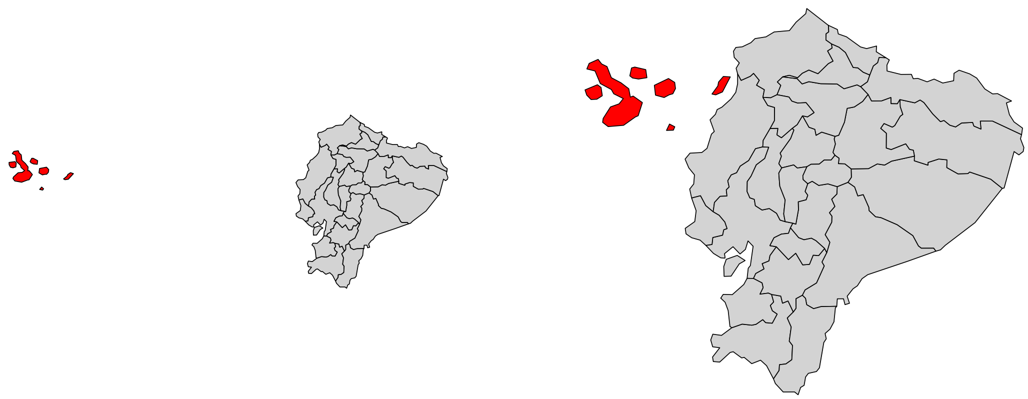
```
apply_mapshaper_commmand("-affine", force_FC = TRUE)
```

Within the "-affine" command, we can specify modifications such as scale, shift, and rotate, in addition to specifying the sub-region these modifications need to be performed on.

4.2.1 Shifting Detached Regions

In cases where an island (island, here, refers to both the geographic definition of a body of land surrounded by water and to a region of a country that may be completely surrounded by a different region or other geographic features) lies far outside the mainland region, the plotting region is stretched and causes the entire region to appear smaller. This method is frequently used for displaying the map of the United States including Hawaii. In most representations, Hawaii is shifted and appears much closer to the West Coast of the United States than it is in reality.

Consider Figure 4.2 (a): Here, the shapefile for Ecuador is plotted, having been previously thinned. Due to the Galápagos Islands (shown in color) being so far west from mainland Ecuador, there are several issues. At the scale shown in Figure 4.2 (a), one could potentially have no issue distinguishing one region from another neighboring region. Remember, however, that if this map were to be used in a LM plot, we would have several repetitions of this map making each of them even smaller given they have to be plotted in a single panel of a LM plot.



(a) (b)

Fig. 4.2: Ecuador is shown in (a) having the boundaries thinned, but otherwise unmodified. In addition to thinning the boundaries, the map on the right (b) has shifted the Galápagos Islands to the east.

In order to fix this, we will use the "shift" option within the "-affine" command. This allows a user to select any region(s) and physically change the coordinates of where this region should be plotted. As a coordinate is an ordered pair of two variables (x, y), we can shift a region both horizontally and/or vertically. In the case of Ecuador, the polygons representing the Galápagos Islands were shifted to the east by changing just the x-coordinate. In doing so, Figure 4.2 (b) shows the Galápagos Islands now appear to be much closer to mainland Ecuador.

The effect of a single horizontal shift is that the entire region appears larger, even when the scale option is not applied to any part of the country.

4.2.2 Shifting Small Regions Outside of the Boundaries

Another instance that may require a user to move a region is where a region is so small compared to the regions surrounding it that in its unchanged state it is either hard to see or not visible at all. Such is the case of many city-states and federal districts within countries.

To fix this issue, a small region may be better represented as an “island" outside of a country’s boundaries. In order to accomplish this in a meaningful way, we will not only enlarge a region but also change the coordinates of a region to have it show up adjacent to the mainland. This has previously been used as a solution for creating a series of LM plots representing all fifty states in the US and the District of Columbia (Carr et al. (1998) and Pickle et al. (2015)). Due to the small area of Washington D.C., the authors shifted the region to lie just off the east coast of the United States. In the “-affine" command, we will need to specify the scale and the shifted coordinates of the region.

The region of Littoral is the main urban area in Benin, a country in western Africa, shown in Figure 4.3. When plotting without modifications as in Figure 4.3 (a), the region is barely visible along the southern coast. Further, if we were to simply enlarge the region in this case, by the time is was large enough for plotting, it would be masking the surrounding regions. By representing the region as an “island" as in Figure 4.3 (b), we are able to make the region much more visible without intruding on the visibility of surrounding regions.

(a) (b)

Fig. 4.3: The map in (a) shows the African country Benin plotted with some thinning done to the boundaries. The issue is the capital region, Littoral, on the southern coast is not visible. (b) on the right shows that by enlarging and shifting the region, the capital region can be made visible.

As this modification may not be ideal, particularly to locals in the region, this is meant to be viewed as one possible solution addressing the fact that simply enlarging the region would cause other issues that may not be as easy to fix.

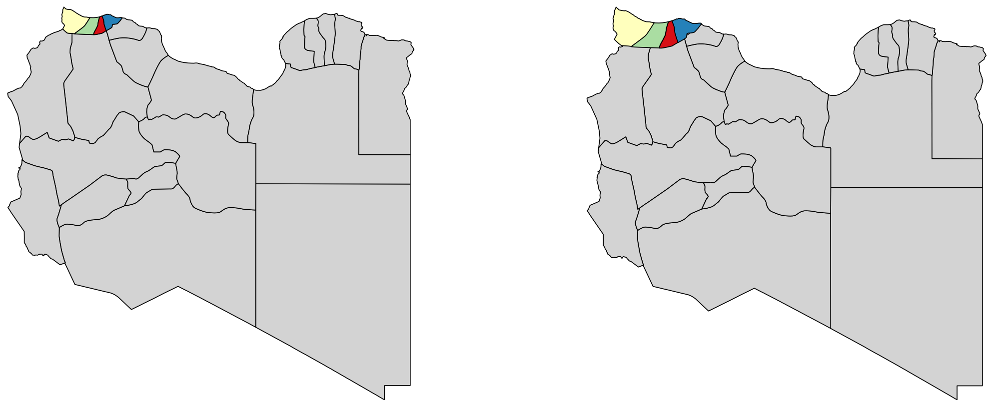
4.3 Resizing Regions

Resizing regions helps to solve two issues caused by a country's geography and plotting a map representation in R. The first issue comes from a region being too large that it dwarfs the remaining regions. This is the case for the US state of Alaska. Not only is this region far away from the mainland United States, but it is by far the largest state. When viewing maps of the US, Alaska often appears much smaller than it is in reality. In some countries, there may be some regions that are small and hard to see when plotted in LM plots. Both of these issues may be remedied by resizing regions.

When resizing a region in Mapshaper, a user must specify the scale of resizing trough the "scale" option within the "-affine" command. The default value is equal to 1. By making this value larger, a user can scale a region to be larger, while choosing a value less than one scales a region to be smaller.

The scaling of a region is centered at the midpoint of the original region. This process does not save the existing borders between two neighboring regions and instead, the enlarged region and all of its neighbors now have overlapping borders. In order to correct this, we first shift the enlarged region to the location we desire and then cut out a hole in the existing region, effectively removing parts of the surrounding polygons and, by extension, the overlap caused by the enlarging of a region. We can then put our modified enlarged region in the hole that was previously cut out.

Prime examples of where fixes such as these are appropriate are the countries surrounding the Saharan Desert of Africa. Often, the area of an administrative district in a given country is inversely proportional to the population in the area. On one hand, heavily populated regions are zoned into small districts, in regards to the physical area the district covers. On the other hand, rural or unpopulated regions are given much larger regions of governance.



(a) (b)

Fig. 4.4: (a) shows that the four regions along the northwest coast of Libya are small enough that when plotting in a LM plot, distinguishing the unique colors of each region would be challenging. (b) shows enlarged and slightly shifted regions that better show the filled color of each region.

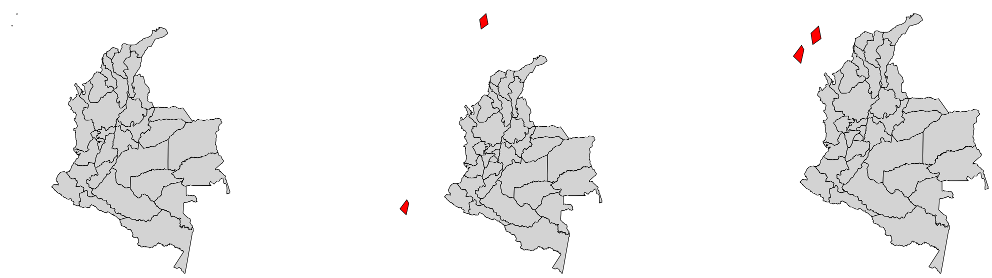
To illustrate this, Figure 4.4 shows a potential modification performed on the country of Libya, whose southern border is located in the Saharan Desert. There are four smaller regions on the northwest coast of Libya. By simultaneous enlarging all four of them and shifting the new, larger boundaries, they are more clearly visible. If we were to use the shapefile from Figure 4.4 (a) in a LM plot, it becomes a challenge to distinguish the differing colors of the four northwestern regions. By enlarging each of these regions and then correcting the alignment by shifting all of the four regions we can obtain the map in Figure 4.4 (b) which makes the four regions more visible.

Resizing can also be used to reduce the size of a region. This would be helpful for regions that are proportionally large compared to the other regions. An example where this is often used is for the state of Alaska in the United States. In Carr et al. (1998) Figures 2 and 3 show this type of modification and represent Alaska a small “island" off the southwest coast of California. In reality, Alaska is much larger.

4.4 Moving Disjoint Areas of the Same Administrative District

In some cases, it may not be enough to employ any single method from above for a given sub-region within the area of interest. These cases are rare, but employ multiple of the previously described methods. Unfortunately, the execution of these steps is not always as simple as performing them in order. Take the South American country of Colombia, for example, shown in Figure 4.5. After thinning, we are left with tiny islands to the northwest of mainland Colombia shown in Figure 4.5 (a). Upon enlarging these regions, the geographic area around and between the islands is also proportionally enlarged, making the map seen in Figure 4.5 (b) not a good solution. In order to create the representation in Figure 4.5 (c), an additional step has to be applied. Within the R package 'sp' (Bivand et al. (2013)) there exists a function, `disaggregate()` for dissolving features that contain disjoint polygonal regions. In essence, when applying the `disaggregate()` function to the islands, we are considering each island as its own distinct region, even though they still have the same name. Now that they are separate, we can apply the `apply_mapshaper_command("-affine")` function and shift the islands separately. When applying the "-affine" command, a user will reference another variable in the shapefile that is unique to each polygon. In the case of the shapefile used for Figure 4.5, the variable 'geo_id' was used. After shifting each island to the desired location a user does not need to aggregate the polygons together. This is because the 'micromap' R package (Payton et al. (2015)) uses the shared name of the polygons, in this case 'NAME_1'.

However, note that while doable at the command level in R, this specific modification is not currently available within the Shiny R app of the 'LMshapemaker' R package.



(a) (b) (c)

Fig. 4.5: When plotting the unmodified shapefile for Colombia shown in (a), the island region of San Andrés and Providencia is not visible in the Pacific Ocean. By resizing and shifting, the islands are enlarged, but so is the area between them (shown in (b). In order to produce the map in (c) where the islands are closer together, the polygons need to be shifted separately.

CHAPTER 5

The ‘LMshapemaker’ R Package

In order to create a meaningful and interpretable linked micromap plot, shapefiles often need to be modified. This has been done in a case-by-case basis for previous work using LM plots. Before now, there has not been a single archive that contains ready-to-use shapefiles for use in LM plots and the process for modifying shapefiles is not as accessible to inexperienced users. The ‘LMshapemaker’ R package has been created as a solution to both of these issues. This package is available to download as a repository from GitHub at `https://github.com/LMshapemaker/LMshapemaker` and can be installed using the ‘devtools’ R package (Wickham et al. (2019)).

5.1 Eligibility for Country Inclusion

In determining which countries were eligible for inclusion in the ‘LMshapemaker’ R package, we applied several filters. The initial goal was to include all of the 193 member states of the United Nations (United Nations (2019)) from the 231 shapefiles included in the Database for Global Administrative Areas (GADM) (Global Administrative Areas (2012)). Shapefiles were obtained via GADM and modifications were done at the first administrative level. GADM provides shapefiles for several administrative levels for each country. The 0-level represents a country, such as the US, and only has one administrative district. The next administrative level (Level 1) represents the individual states and districts. In the example of the US, this consists of 51 administrative districts (50 states and 1 district). Level 2 represents counties of the U.S., etc. For the purpose of the initial build of ‘LMshapemaker’, only the first administrative level of each country was considered. For a full list of the 135 countries included in the ‘LMshapemaker’ R package, see the provided tables in Appendix A.

5.2 Exclusions from the 'LMshapemaker' R Package

Due to a variety of reasons, some countries were excluded from the 'LMshapemaker' R package. Modifications could be made to the countries excluded, but for the scope of this R package at its initial release, they were not considered.

The first filter from excluding countries was whether or not a country is fully recognized as a sovereign nation by the UN. Examples include countries such as Greenland, the Faroe Islands, and Puerto Rico. These countries are all territories of other nations, specifically Denmark and the USA.

Another reason for excluding a country from its inclusion in the R package was due to it having a small number of Level 1 administrative areas. This decision was made because a LM plot with only five or fewer regions would only require a single repetition of the map in the map panel. In this case, a LM plot would not be meaningful or interesting. Due to this, any country that had five or less regions at the first administrative level was excluded. A notable example of an excluded country is the United Kingdom. The first administrative level for the UK consists of the four constituent countries that are part of the United Kingdom, namely: England, Northern Ireland, Scotland, and Wales. Thus, at the first administrative level of the United Kingdom, there are not enough regions to make a LM plot advantageous.

Similar to the problem of too few Level 1 administrative districts, issues arise from countries having many administrative districts or complex administrative organizations. A few examples that represent these issues are Uganda, India, and Russia. Uganda has over one hundred districts. This would require potentially many modifications and the use of multi-panel LM plots to meaningfully show the data. The administrative regions of India involve both states and territories within the GADM shapefile. Russia is a unique case that involves controversial borders, a unique system of administration that involves districts, republics, federal cities, and other federal subjects. In addition to these problem, there are more than 80 regions.

Complex geography is an additional limiting factor in modifying shapefiles. Some countries, such as the Bahamas, are archipelagos made up of many small islands that are each their own administrative district. While meaningfully modifying countries similar to this to be visible within a LM plot is doable at the command level in R, this is not possible in the Shiny app (see Sections 4.2.1 and 4.4) and would require many more lines of code within the `apply_mapshaper_command()` function.

5.3 'LMshapemaker' Functionality

The primary features of the 'LMshapemaker' R package are as follows: (i) to be an archive for users to find ready-to-use shapefiles for creating meaningful LM plots and (ii) to allow users to easily tailor shapefiles to their preferences. While there are some exceptions that were discussed in Section 5.2, users can find modified shapefiles for most of the countries in the world with an overview of the recommended modifications in the 'LMshapemaker' R package. In addition to the shapefiles, for each country, there exists a modification table that details the modifications performed in the shapefile. Whether a user chooses to use the provided modified shapefile or to create their own based on their own modifications in table format, the 'LMshapemaker' app will allow users to obtain the shapefile they need.

5.3.1 Shiny Shapemaker

The first feature to point out in 'LMshapemaker' is the Shiny R app (Chang et al. (2018)). This app allows a user to upload a shapefile, previously obtained from GADM, and apply modifications to the boundaries and regions of the shapefile. This Shiny app is run externally from R (R Core Team (2019)), and instead launches in the computer's preferred web browser. In order to launch the 'LMshapemaker' Shiny app, a user needs to use `runShinyShapemaker()` in R.

Upon running this function, users will be directed to a web browser page where they can upload their shapefile and work with a previously created modification table or create their own table. Users can specify the level at which the boundaries will be thinned and how the regions will be modified. After uploading a shapefile, the app will display the shapefile in its unmodified state. Upon specifying modifications, the app will plot the modified map and highlight the regions that were affected. As further changes are made, the app will update, allowing the user to see the changes made in real-time. When finished, the user can export the modification table that contains the details of the changes made.

The benefit of using the Shiny app is that it is accessible to users with little experience using R or working with shapefiles. More details for understanding and using the Shiny app can be found in Chapter 6.

5.3.2 Ready-to-Use Shapefile Archive

After installing ‘LMshapemaker’, users will have access to shapefiles for most of the countries in the world. For a full list of included countries, see Appendix A.1. The data for each country is stored as a list that contains 2 objects. The first object in the list is a data frame detailing the modifications that were done. The second object is a recommended modified shapefile.

Accessing the data for countries included in the package is done using a `data()` call. For example, to obtain the data for Mexico, a user would input `data("mexico")` into R. This will create a list object called “mexico” in the global environment. The first element of the list is a data frame that shows the level at which the boundaries were thinned and which regions were modified. An example is shown in Figure 5.1 (a). In this example, the first row specifies the level at which the boundaries were simplifed. 0.1% of the original boundary points remain in the modified shapefile. The following three rows relate to the modifications to specific regions, with ‘V1’, ‘V2’, ‘V3’, and ‘V4’ representing the scale value, the rotation, longitudinal shift, and latitudinal shift, respectively. The second object returned by the `data()` call is a modified shapefile using the modifications in the data frame, shown in Figure 5.1 (b). The three modified regions specified in Figure 5.1 (a) are manually colored in red.

	Command	Region	V1	V2	V3	V4
1	simplify		0.001	NA	NA	NA
2	affine	Distrito Federal	2	0	0	0
3	affine	Tlaxcala	1.5	0	0	0
4	affine	Morelos	1.5	0	0	-0.4

(a)

(b)

Fig. 5.1: The following are provided by the data function call in 'LMshapemaker': (a) a data frame that details the modifications done to Mexico, and (b) a modified shapefile where the modifications have been applied. The map in (b) shows a visualization of this shapefile with modified regions manually colored in red.

5.3.3 Custom Modifications to Shapefiles

An additional function is provided in the 'LMshapemaker' package allowing users to apply their own modifications to shapefiles. It is to be understood that the shapefiles and modifications provided in the package are objective recommendations for optimal shapes to create LM plots. Users more familiar with local culture and customs may have a different perspective on how to best represent the country of interest. If this were the case, users could use the Shiny app to create a modification with their specifications and make use of the `mod_shape(mod_table, shape)` function to create the shapefile they desire. In this function, **mod_table** is a modification table provided by the Shiny app. **shape** is a shapefile provided by GADM.

	Command	Region	V1	V2	V3	V4
1	simplify		0.001	NA	NA	NA
2	affine	Distrito Federal	2	0	5	1
3	affine	Tlaxcala	1.5	0	0	0
4	affine	Morelos	1.5	0	0	0

(a)

(b)

Fig. 5.2: This figure shows a hypothetical alternative to the recommended shapefile provided by the 'LMshapemaker' R package. (a) displays a data frame detailing the alternative modifications done to Mexico. (b) shows the map with the modifications applied (coloring manually applied), in which the Federal District of Mexico City is being represented as an "island".

Figure 5.2 shows a hypothetical situation, in which a user has different preferences for what the map of Mexico should look like. In this example, the hypothetical user wants to emphasize the Federal District of Mexico City and represents it as an "island". The data frame in Figure 5.2 (a) shows the modifications necessary to produce such a map as obtained through the Shiny app. For the sake of demonstration, it will be referred to as `alt_table`. The user is using a shapefile, downloaded from GADM, called `mex_shape`. By running `mod_shape(alt_table, mex_shape)`, the user can produce the shapefile with their desired modifications and use it for the creation of a LM plot.

5.4 Implementation Details

The `mod_shape()` function call is what makes the 'LMshapemaker' package unique. It takes a shapefile and a data frame as its arguments and produces a new shapefile with the specified modifications. The code that creates the `mod_shape()` function is the following:

```
mod_shape <- function(shape, mod_table){
  shape_json <- geojson_json(shape, geometry = "polygon",
                             group = "group")
  shape_sp <- geojson_sp(shape_json)
```

```
  if (mod_table[1, 1] == "simplify"){
    shape_mod <- ms_simplify(shape_sp, mod_table[1, 3])
  } else {
    shape_mod <- shape_sp
  }

  ms_command <- vector()

  for(i in 1:nrow(mod_table)){
    ms_command[i] <- ifelse(mod_table[i, 1] == "affine",
                            paste("-affine",
                                  paste0("scale=",
                                         mod_table[i, 3]),
                                  paste0("rotate=",
                                         mod_table[i, 4]),
                                  paste0("shift=",
                                         mod_table[i, 5], ",",
                                         mod_table[i, 6]),
                                  paste0("where='NAME_1 == \"",
                                         mod_table[i, 2], "\"'")),
                            "")
  }

  ms_input <- paste(ms_command,  collapse = " ")
  shape_mod_json <- geojson_json(shape_mod)
  mod <- apply_mapshaper_commands(shape_mod_json,
                                  ms_input, force_FC = TRUE)
  mod <- geojson_sp(mod)

  mod_clean <- mod
  for (i in which(shape$NAME_1 %in% mod_table[, 2])) {
    mod_clean <- ms_erase(mod_clean, mod[i, ])
    mod_clean <- rbind(mod_clean, mod[i, ])
  }
  return(mod_clean)
}
```

Upon running the `mod_shape()` function with the shapefile and data table specified, the function will prepare the GADM shapefile and convert it into the correct format to be compatible with the 'rmapshaper' commands described earlier in this chapter and in Chapter 5. To do this, `mod_shape()` will first convert the shapefile into a Java Script Object Notation (.json) file and from there into a Spatial Polygon (.sp) file. In this .sp format, the shapefile is compatible with the `ms_simplify()` function. The table obtained from the Shiny app (discussed in detail in Chapter 6) is structured so that if simplification is a desired modification it will be located in the first row of the table. `mod_shape()` will identify that 'simplify' has been specified and will search under the 'V1' column for the values that boundary thinning will be necessary. If the first row did not specify 'simplify' under the command step, then the thinning modification would be skipped.

After thinning the boundaries (or not in the case of this step being skipped), the `mod_shape()` function will then use the remaining rows in the data table to build a string that will be passed through the `apply_mapshaper_commands()` function from the 'rmapshaper' package. Each row with 'affine' specifed in the 'Command' column will be used in building this string. Within the `mod_shape()` function code, there are several `paste()` commands that are used to build differnt parts of the string based on the values in the 'V1' through 'V4' columns of the data table. Each of these values represent different types of modifications that can be applied to regions in the shapefile. With the string compiled, the `apply_mapshaper_commands()` function will modify each of the regions with the values and modifications specified.

The final function of the `mod_shape()` function is to clean up the new boundaries. This is done using the `ms_erase()` function from the 'rmapshaper' package to "cut out" the old boundaries that are being overlapped by any region that was enlarged and then filling in the "hole" with the new modified region. This last step is done internally in the function and requires no specifications or input from the user.

As an example, the following is the data table associated with recommended modifications to Mexico. The same table can be seen in Fig 5.1 (a).

```
> mexico[1]
[[1]]
   Command            Region     V1  V2  V3    V4
```

```
1 simplify                  0.001 NA NA   NA
2   affine Distrito Federal 2.000  0  0  0.0
3   affine         Tlaxcala 1.500  0  0  0.0
4   affine          Morelos 1.500  0  0 -0.4
```

Upon running the `mod_shape()` function, the shape file will be converted into the necessary file format for modifications. Upon reading the row with 'simplify' in the 'Command' column, the function will look to 'V1' for the specified level of thinning. In this case, it is 0.001 or 0.1%. After thinning. The `mod_shape()` function will use the remaining three rows, each with 'affine' in the 'Command' column to build a string with the desired specifications. The first section of this string will specify that the Distrito Federal region will be enlarged by a scale factor of 2.0. The second piece of the string will specify that the Tlaxcala region will be enlarged by a scale factor of 1.5. The final piece of the string will specify that the Morelos regions will be enlarged by a scale factor of 1.5 and shifted south 0.4 degrees latitude. With the string compiled, the `mod_shape()` function will pass it through the `apply_mapshaper_commands()` function and make the necessary changes to the coordinates defining these regions.

Lastly, the function will cut out regions in the existing shapefile and fill in the holes with the new modified regions.

CHAPTER 6

Shiny Application

In order to assist those who are interested in using linked micromap plots as a visualization but who do not have access to a suitable shapefile, an app was developed through Shiny R (Chang et al., 2018). This app is designed to allow users to start with a shapefile and in near real-time see the changes made to the boundaries and regions included in the shapefile. This will allow users to tailor the shapefile to ensure interpretability and meaningful plotting when creating a LM plot.

6.1 Layout and Functionality

This app, done through a web interface, is split into two side-by-side panels. The side panel located on the left side of the app window is where a user will upload files and specify modifications to be made. The main panel, initially blank (shown in Figure 6.1), will display the table detailing the modifications made to a shapefile and maps comparing the original and modified shapefiles to each other.

6.1.1 Side Panel

The side panel, as displayed in Figure 6.1, is the interactive part of the application. It is here that a user will upload a shapefile by using the file search option under 'Input Shapefile'. When selecting a shapefile, a user will need to select the four files that make up the boundaries of interest. These files have to be named as follows: .shp, .shx, .dbf, and .prj.

The .shp, .shx, and .dbf files are three mandatory files that make up functional shapefiles (ESRI (1997) and Dempsey (2016)). The main file (.shp) stores the information describing the geometry of the shapefile. This includes information on the shapes and their vertices. The index file (.shx) stores the indexing of the attributes and the shapes inluded in the .shp

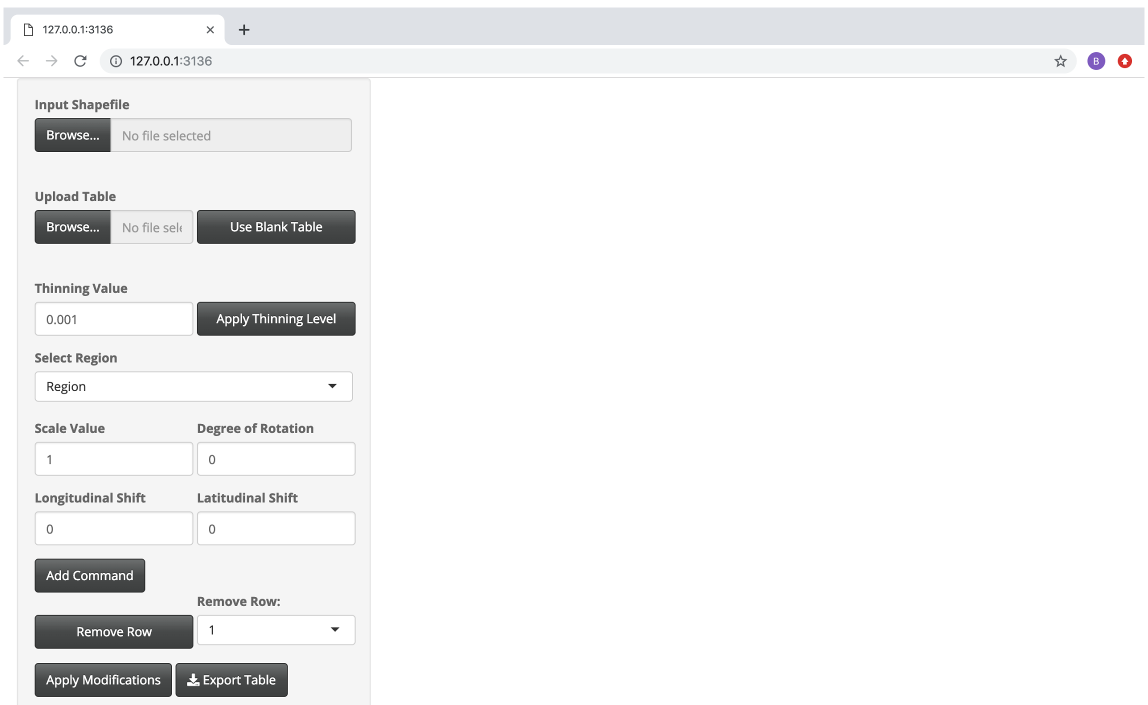


Fig. 6.1: Example of the web browser Shiny R app upon loading. The side panel, on the left, has options for uploading and specifying modifications. The main panel will display the table detailing modifications and a comparison of the shapefile before and after applying modifications.

file. The dBASE table file (.dbf) stores the infomation on the attributes included in the shapefile, such as name and id of each included region. While not a mandatory file for using shapefiles, the Projections Definition file (.prj) is recommended and stores information for the projections and associated coordinate system to be used for the shapefile.

Upon selecting these four files, the map associated with the shapefile will be displayed in its raw, i.e., unmodified, state.

For the creation of the table detailing the modifications, the user will either upload the table of suggested modifications from the 'LMshapemaker' R package (these tables are stored as .csv files) or start from a blank table with no modifications specified. In either case, the table will have the same format with the column headers being, 'Command', 'Region', 'V1', 'V2', 'V3', and 'V4'. Depending on what is stored in the command field, the values stored in the 'V' columns will represent different parameters associated with the command.

Upon selecting a shapefile to upload and a modification table to use, the app will update its main panel and display both the table and the unmodified map for the uploaded shapefile. Figure 6.2 displays both the modification table and unmodified map for the country of Libya in northern Africa.

6.1.2 Main Panel

As soon as a user has specified a shapefile and table for the modifications, both will be rendered in the main panel of the app. This is shown in Figure 6.2. The top section of the main panel will show the table and then update with each added or removed command, thus, showing an up-to-date record of the modifications that should be kept.

The bottom section displays the map detailed in the original shapefile on the left and, upon applying modifications, will display the map with the specified modifications in the modification table. Regions that were affected by any modification (an exception being the thinning modification as it is a global modification) will be highlighted in red. The maps being displayed side-by-side allow the user to easily see the changes that were specified, shown in Figure 6.3.

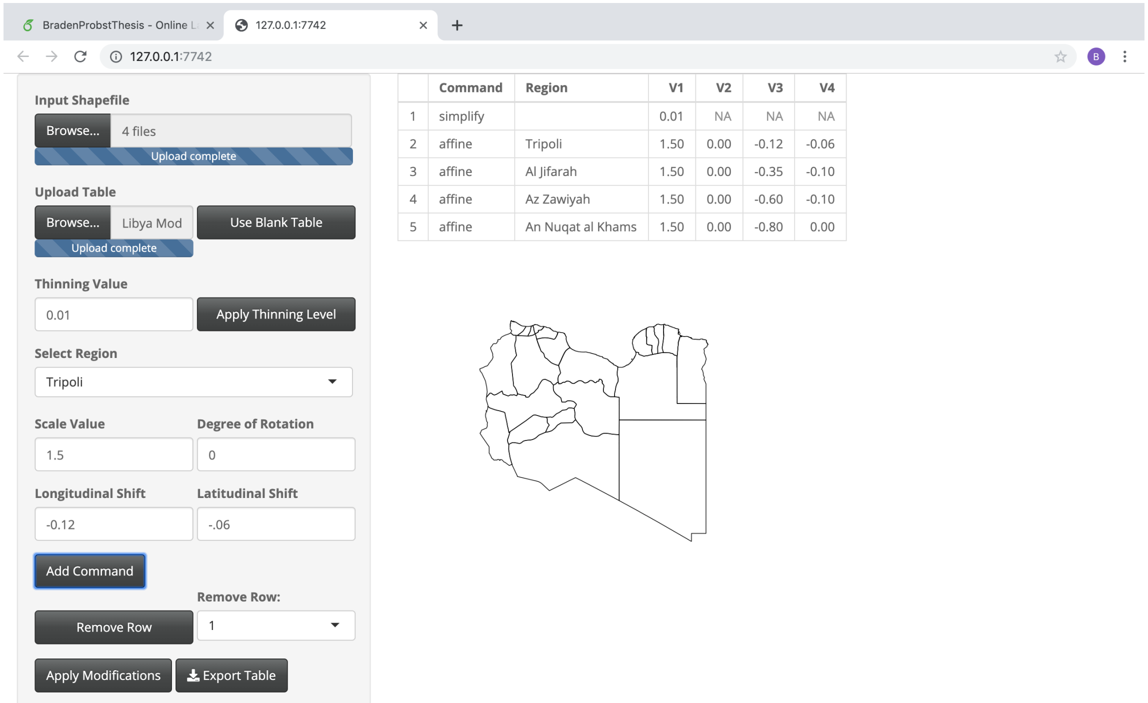


Fig. 6.2: Example of the web browser Shiny R app upon uploading a shapefile and modification table for the country of Libya.

6.1.3 Using the Side Panel

Within the side panel, the various modifications detailed in Chapter 4 can be specified and added to the table in the main panel. The first of these modifications that should be considered is the value at which a user wants to thin the boundaries. The default value is set to 0.001, however, users can specify in the 'Thinning Value' box a more appropriate level for their shapefile. Recall from Chapter 4 that the value specifying the thinning level represents the proportion of points to be kept in the boundary lines. Thus, values closer to 1 will produce more complex, accurate maps while values closer to 0 will produce simpler, more abstract maps. By selecting the 'Apply Thinning Level' button, the app will add a row to the modification table with this specification. The command is classified as 'simplify' and the value selected by the user will be stored under 'V1'. In the case that a previous thinning level had been specified, the 'Apply Thinning Level' will update the value stored in 'V1'.

When a shapefile is uploaded, the different regions will populate a drop down list

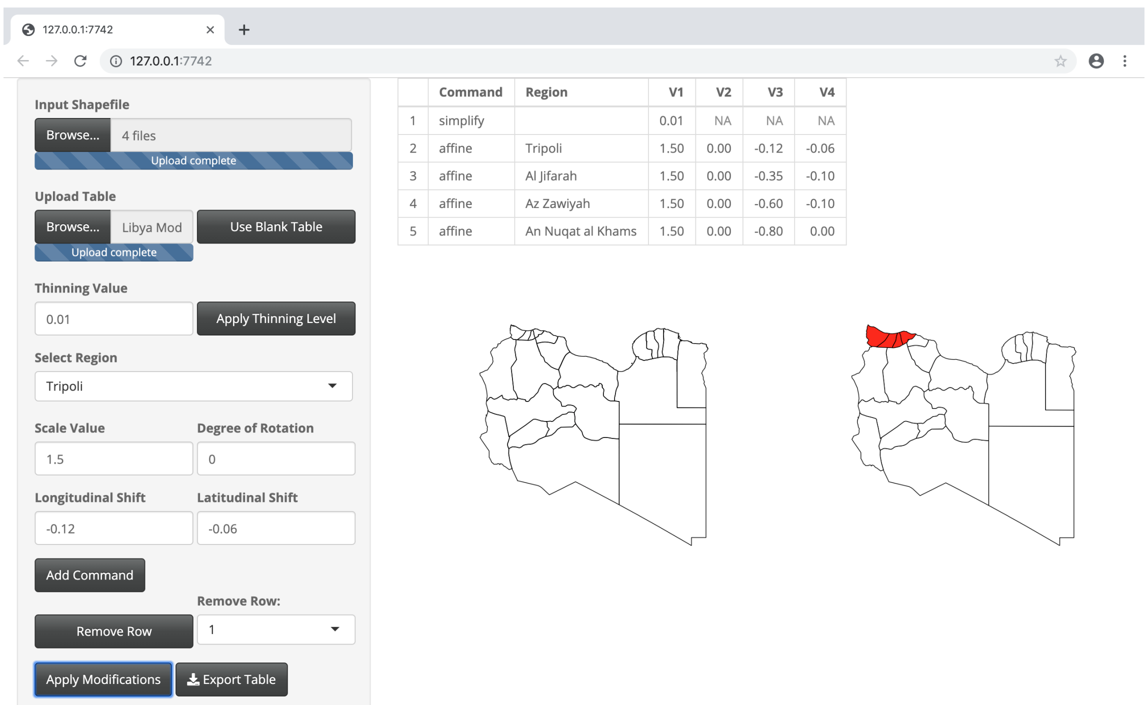


Fig. 6.3: Example of the web browser Shiny R app upon applying modifications detailed in the modification table located in the main panel. The map of Lybia on the left displays the unmodified shapefile. On the right, the modified map of Libya is shown with thinning having been applied to the boundaries and four regions being scaled and shifted in the northwest portion of Libya. These regions are highlighted for easy comparison of the two maps.

labelled ‘Select Region’ and be available for selection. The four modifications that can be done to each region are entered in the four numeric input boxes labelled ‘Scale Value’, ‘Degree of Rotation’, ‘Longitudinal Shift’, and ‘Latitudinal Shift’. These modifications are all classified as options in the ‘affine’ Mapshaper (Harrower and Bloch (2006)) command that was described in Section 4.2.

For the ‘Scale Value’, the default value is 1. To enlarge a region a user would specify a value greater than 1. To make a region smaller, the user should specify a rational number between 0 and 1. The value will be stored in the column ‘V1’ in the modification table.

The ‘Degree of Rotation’ modification is measured in degrees. In order to rotate a region, a user should specify any number between -180 and 180. The value will be stored in the column ‘V2’ in the modification table.

The ‘Longitudinal Shift’ and ‘Latitudinal Shift’ values are measured in degrees around the surface of the earth. Thus, the values specified in these boxes will not directly translate to linear distances, such as miles and kilometers. This step, due to its unit of measurement, may require trial an error to get a region in the right location. These values will be stored in the columns ‘V3’ and ‘V4’, respectively, in the modification table.

After specifying each of these values, the ‘Add Command’ button will either update a row with the selected region or add it to the list of modifications to be done. This will consolidate all modifications to a given region into a single line of code.

If a user decides that one of the modifications is unnecessary or unwanted, selecting the row number from the ‘Remove Row:’ drop-down list and clicking the ‘Remove Row’ button will delete the specified row and any associated modifications from the modification table.

By selecting the ‘Apply Modifications’ button, the application will read through the compiled modification table and re-plot the map associated with the shapefile with the indicated changes to the right of the original plotted map. Regions that were modified will be highlighted in red. An example of this can be seen in Figure 6.3 where the regions of Libya in the northwest are highlighted red after being scaled and shifted for more visibility.

The final button of the side panel is the ‘Export Table’ button that will allow a user to download the table specifying the different modifications as a .csv file. This table can then be read into R for creating and using the modified shapefile needed in a spatial visualization for LM plots.

6.2 Limitations of the Shiny App

While the Shiny app is an accessible tool for users wanting to customize shapefiles, there are some limitations. These limitations include compatibility with shapefiles and which of the modifications, detailed in Chapter 4, are possible in the Shiny app.

The initial launch version of the app will only be able to work with shapefiles that are formatted similarly to those provided by GADM (Global Administrative Areas (2012)). While GADM provides shapefiles at different administrative levels for each country, the focus of the 'LMshapemaker' and its associated Shiny app is to provide recommendations for shapefile modifications specific to first level administrative divisions (variable 'NAME_1' in the shapefile data). Due to the maximum upload size allowed in the shiny app, large countries with long coastal boundaries will not work within the Shiny interface. Canada and China are examples of such countries, whose large geographies include many small islands and coastal variations.

The GADM shapefile for Nicaragua includes a region called Lago Nicaragua (Lake Nicaragua). Due to this region being included, when plotting in a LM plot, the island regions located in the lake become masked and not visible. In order to properly and meaningfully view Nicaragua as a LM plot, this seeming extraneous region was manually removed (see Section A.4, page 183).

Part of 'rmapshaper' functionality is to modify the various levels of a shapefile. The 'LMshapemaker' Shiny app only allows users to access the 1st administrative level. Meaningful modifications can be done at the command level in R using different levels. In the case of Japan, the final product features the mainland being rotated and the Oninawa and Hokkaido islands shifted slightly above and below the rotated landmass. The rotation is applied at the 0-level administration and only affects the largest of the land masses. See Section A.2, page 112 for details.

Throughout Chapter 4, we detailed the different modifications that a user can apply to customize shapefiles. Among these modifications were two that, while doable at the command level in R, are not currently available as options in the Shiny app. These modifications that are excluded from the app are filtering islands and shifting disjoint polygons within the same region. As for the first of these modifications, the step of thinning will function as a valid substitute in most cases. Allowing the second mentioned modification is more of a challenge. It is the intent to make this shiny app more robust and include functionality in the future that can address these modifications.

CHAPTER 7

Examples

In this chapter, we will look at examples for two countries using the modified shapefiles provided in the 'LMshapemaker' R package. For each of these countries, data was collected from the web from wikipedia.org, specifically the 'Providences of Ecuador' data for Ecuador from https://es.wikipedia.org/wiki/Provincias_de_Ecuador; and for Mexico, the 'Mexican States by GDP per capita - 2016' data from https://en.wikipedia.org/wiki/List_of_Mexican_states_by_GDP_per_capita, the 'By GDP per capita (2007)' data from https://en.wikipedia.org/wiki/Ranked_list_of_Mexican_states, the 'Mexican States' and 'time series' data from https://en.wikipedia.org/wiki/List_of_Mexican_states_by_homicides#cite_note-mexst17-11, and the 'Mexican States' data from https://en.wikipedia.org/wiki/List_of_Mexican_states_by_Human_Development_Index.

7.1 Ecuador

Ecuador, a country in South America, poses an interesting challenge in preparing its shapefile for use in a meaningful LM plot. While the regions on mainland Ecuador don't require too much attention, the issue lies with the Galápagos Islands region, a province that is situated 851 miles (1,369 kilometers) west of the mainland in the Pacific Ocean. In order to best visualize Ecuador in a LM plot, the Galápagos Islands need to be shifted closer to the mainland.

Figure 7.1 shows the the modifications made via the 'LMshapemaker' R package Shiny app. The changes made are to thin the boundaries so that 0.4% of the original boundary points remain in the shapefile and to shift the Galàpagos Islands to be closer to the main land. Specifically, the islands have been shifted 8.5 degrees to the east and 3.5 degrees to the south. Using this modified shapefile, we can construct a LM plot that more optimally uses the space available in the map panel of the plot.

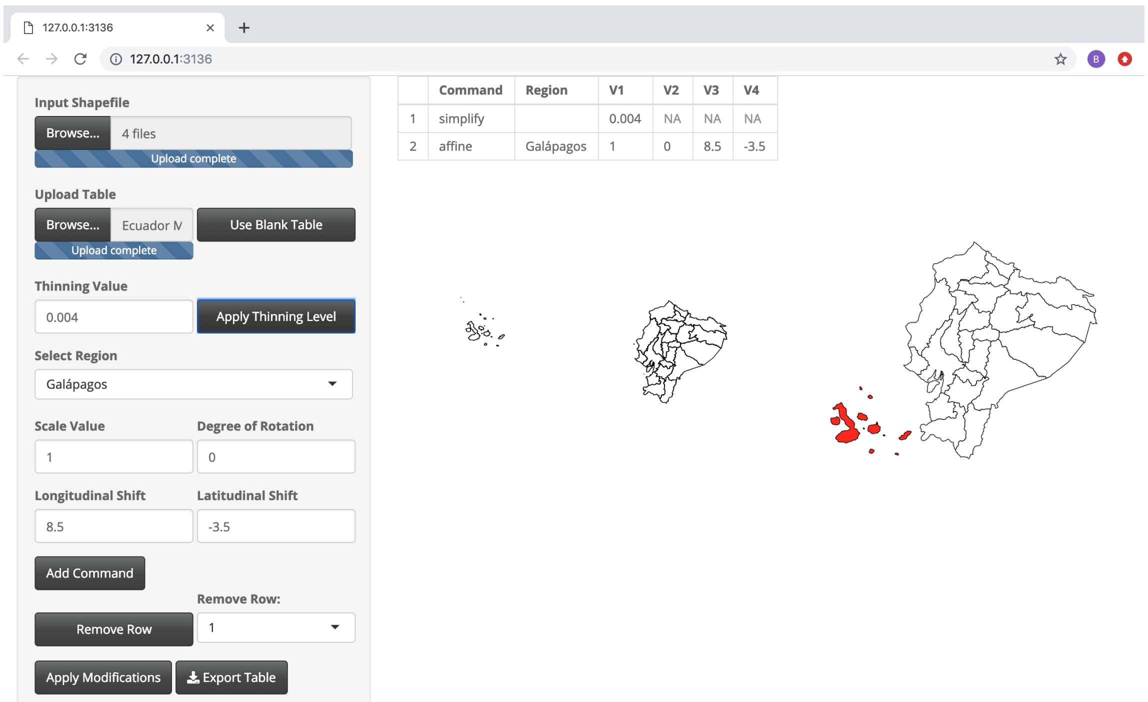


Fig. 7.1: Specifications detailing the modifications done to the country of Ecuador. Thinning was done so that 0.4% of the original boundary points were retained. The province of Galápagos was shifted to the east by 8.5 degrees and to the south by 3.5 degrees. The shifted Galápagos Islands are shown in red. The shapefile as a whole appears larger due to minimizing the white space between the islands and the mainland.

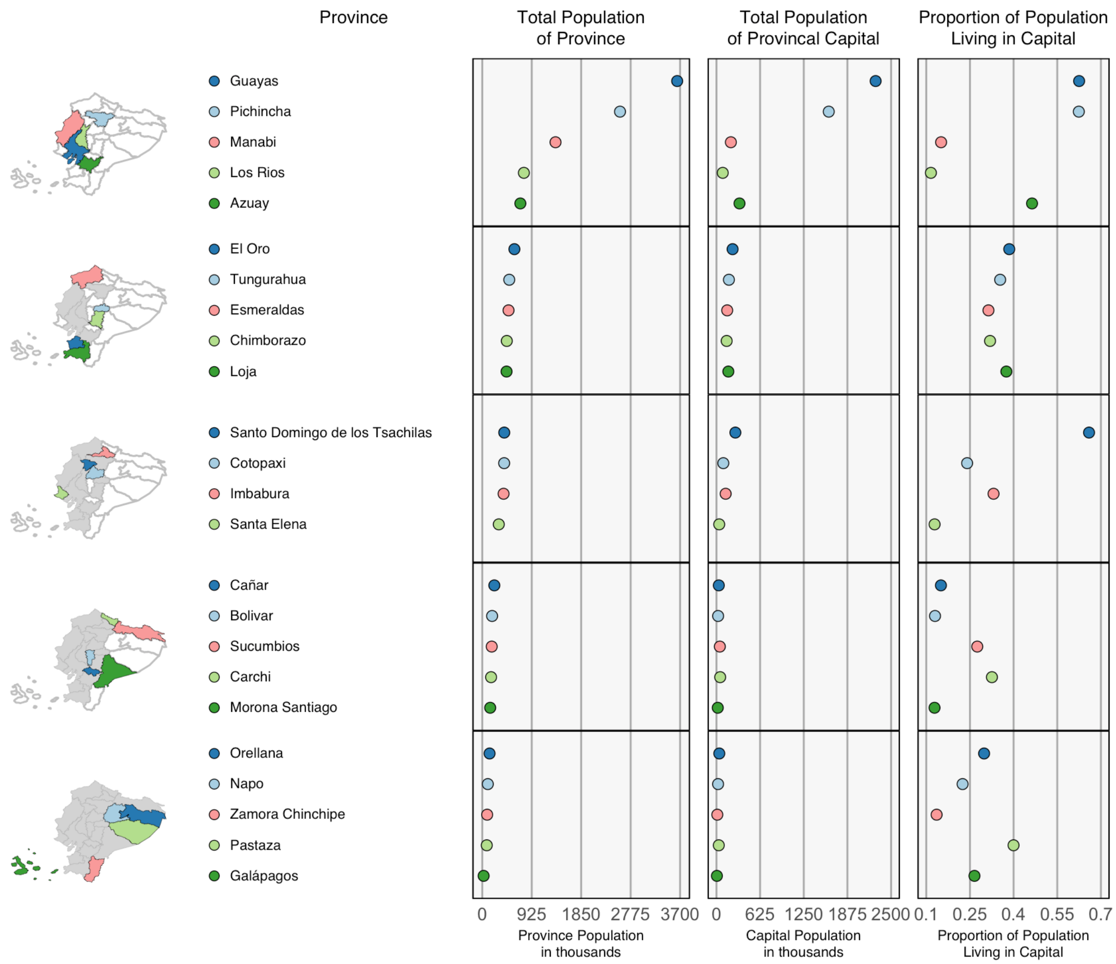


Fig. 7.2: Linked micromap plot for the country of Ecuador. The statistical panels show dot plots representing the population of the province, the population of the provincial capital, and the proportion of each province's population that live in the province's capital city.

Figure 7.2 shows a completed LM plot using the modified shapefile (shown in Figure 7.1) which is accessible in the 'LMshapemaker' R package. The LM plot is sorted by the total population for each province and compares each provincial population to the capital population in each province as well as the proportion of the population that lives in the capital of each province.

The LM plot shows a strong spatial relation to the provincial and capital populations. The most populous regions both in terms of total population and capital population are the western regions and more coastal with the jungle regions in the east and the Galápagos islands being the less populous provinces. Further, the proportion of the population living in the provincial capitals has a moderately strong correlation with both the total provincial population and provincial capital population. The correlation coefficients are 0.57 and 0.67, respectively. This suggests that the more populous a province is, it is likely to have a larger capital city. Of note are three provinces whose provincial capitals have a proportion of the population that does not follow the pattern the others follow. These provinces are the Manabi, Los Rios, and Santo Domingo do los Tsachilas provinces.

7.2 Mexico

In order to make a shapefile suitable to best represent data associated with Mexico in a LM plot, several changes had to be made to some smaller states and to the Federal District of Mexico City (Distrito Federal). After thinning the boundaries, there were issues with viewing three of the regions within the country. Firstly, the Distrito Federal was enlarged by a scale factor of 2. Additionally, the two neighbor states of Tlaxcala and Morelos were sufficiently small that viewing them in the context of a LM plot was a challenge. To fix this, they were enlarged by a scale factor of 1.5. Finally, to avoid overlapping the enlarged Distrito Federal, the enlarged state of Morelos was shifted south 0.4 degrees latitude. By using the shapefile shown on the right in Figure 7.3, each of the 32 states of Mexico are visible and readable in a LM plot (the regions that were modified are highligted in red).

A completed linked micromap plot using the modified shapefile (shown in Figure 7.3) which is available through the 'LMshapemaker' R package is shown in Figure 7.4. This LM

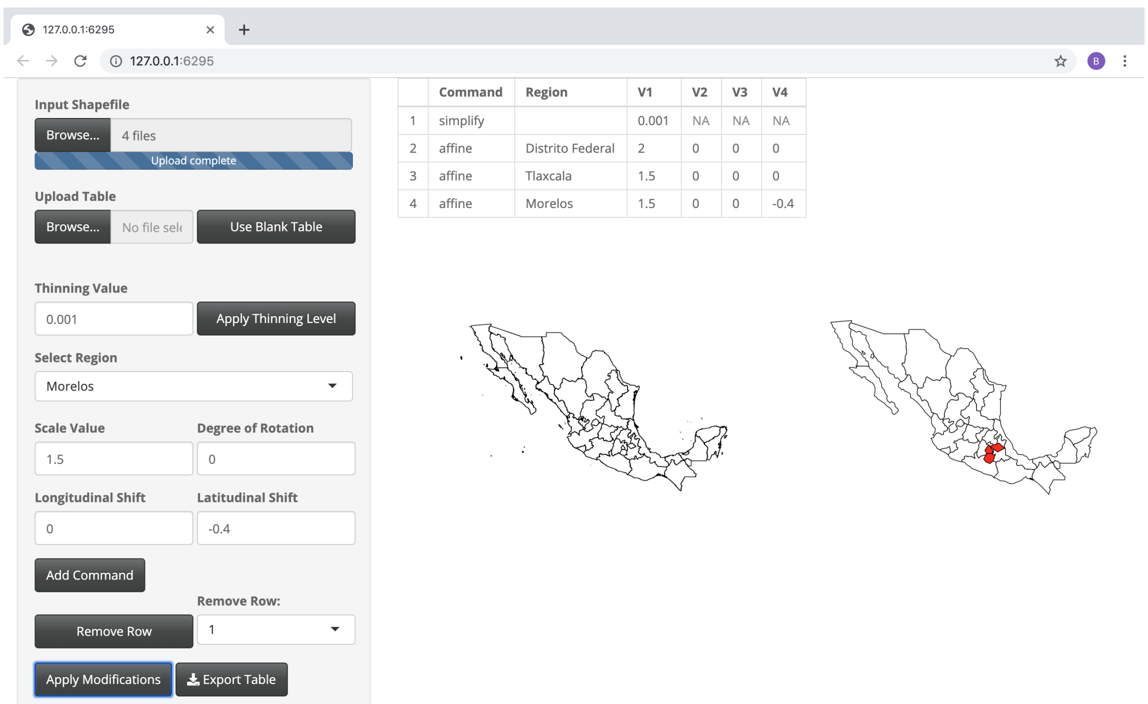


Fig. 7.3: Specifications detailing the modifications done to the country of Mexico. Thinning was done so that 0.1% of the original boundary points were retained. The states of Tlaxcala and Morelos, and the Federal District of Mexico City were all enlarged. Additionaly, the state of Morelos was shifted south 0.4 degrees latitude. Each modified region is highlighted in red.

plot shows relations between Human Development Index (HDI) scores (a measure of life expectancy, GDP per capita, and education), Gross Domestic Product (GDP) per capita, and the percent change in murder rate across similar periods of time in Mexico. The plot is sorted by the state's HDI scores in 2017.

Both of the statistical panels for HDI and GDP per capita are represented using arrow plots showing changes in values over the time period specified in each panel. The tail and tip of the arrows indicate the earlier and most recent values for each variable, respectively. The third panel shows the percent change in murder rate for each state over the specified time period.

There is a noticeable spatial trend within the map panel on the left. The more north a state is, the higher its HDI score. The obvious spatial outlier is the Federal District of Mexico City (Distrito Federal). As it is the most populous and most developed city in the country, it makes sense that it also has the highest HDI score and, until recently, the highest GDP per capita. The spatial pattern somewhat extends to GDP per capita as well, as more northern states typically have higher GDPs per capita. The correlation between HDI and GDP per capita has slightly weakened over over time ($r \approx 0.74$ for the earlier (arrow tails) data points has decreased to $r \approx 0.56$ for the most recent (arrow tips) data points). Even with the decrease, there is still a moderate positive correlation between the two variables. The one major outlier with respect to GDP per capita is the state of Campeche, whose GDP per capita grew from 15,000 to 53,000 and now has the highest GDP per capita in the country.

The final statistical panel, while not strongly correlated with either of the other variables ($r \approx -0.002$ for the difference in HDI scores and the difference in murder rate and $r \approx 0.1$ for the difference in GDP per capita and difference in murder rate), tells its own interesting story, in addition to the apparent spikes in murder rate that are shown.

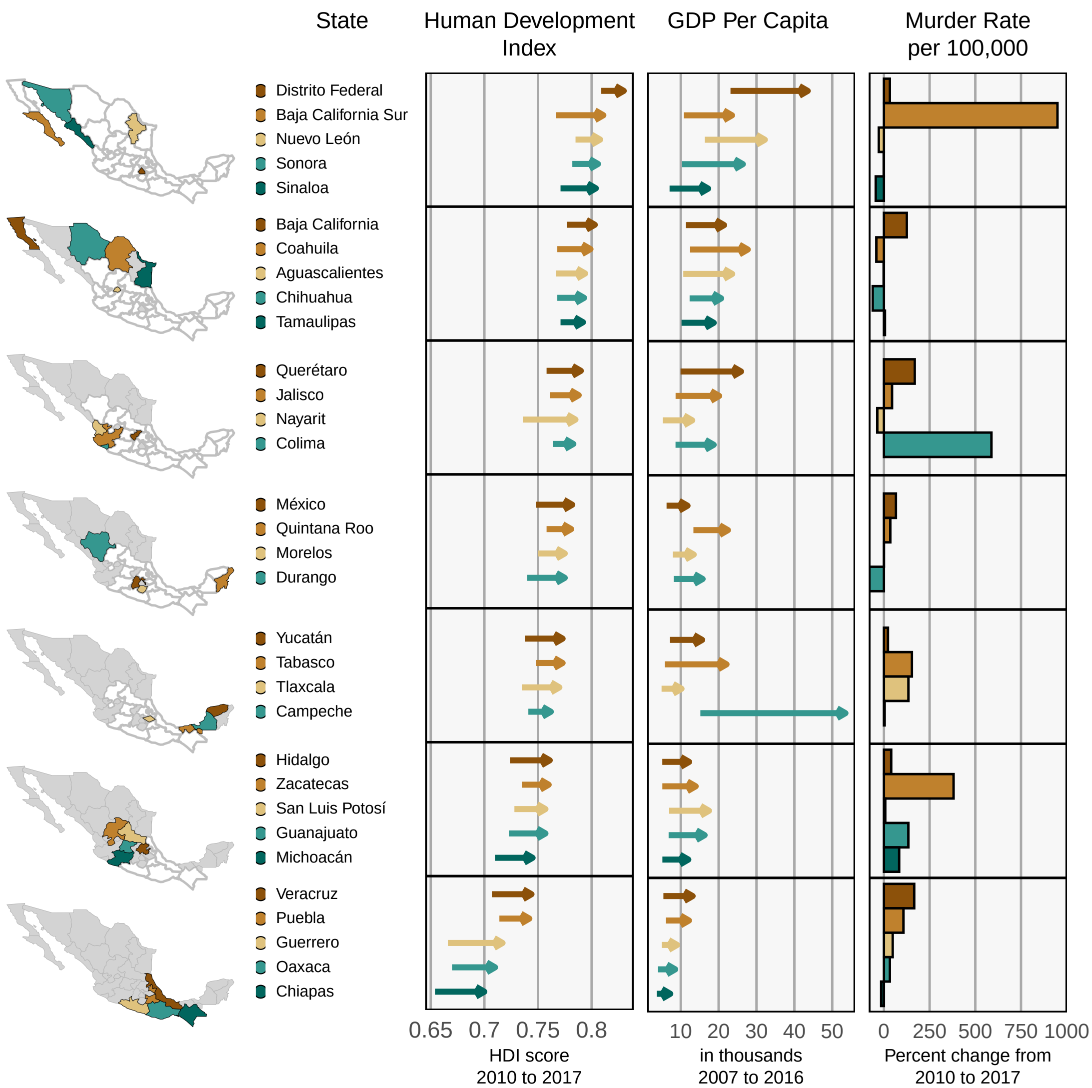


Fig. 7.4: Linked micromap plot for Mexico. An arrow plot represents the change in Human Development Index scores from 2010 (tail of the arrow) to 2017 (tip of the arrow) for each region. The second statistical panel shows an arrow plot representing change in GDP per capita from the years 2007 to 2016. The third statistical panel shows the percent change in murder rate for each region from 2010 to 2017, represented as a bar plot.

7.2.1 Further Visualizing Mexican Murder Rates

As previously mentioned, the difference in murder rate from 2010 to 2017 did not strongly relate to either the HDI score or GDP per capita. Using the same shapefile provided in the 'LMshapemaker' R package, we can more closely examine the single panel of murder rates from Figure 7.4.

In Figure 7.5, the first statistical panel in this graph shows the murder rate from 2010 for each state. The whole LM plot is sorted by this variable. The second statistical panel contains the murder rate from 2017. The final panel is the same data as in the third statistical panel of Figure 7.4, but sorted differently. In this configuration, there aren't any striking patterns between the murder rates in our starting and ending years. There appears to be some inverse relation between the murder rate 2010 variable and the percent change observed in those states. By sorting with respect to a different variable such as the change in murder rate from 2010 to 2017, the phenomenon becomes clearer.

Figure 7.6 shows the same data used in Figure 7.5 but sorted by the overall change in murder rate (this variable is not directly shown in the LM plot but is the difference between the murder rates in 2017 and 2010). While the majority of states are showing an increase in murder rate, sorting by this variable shows that the previously most dangerous areas of Mexico (the northern-central states) are experiencing a drop in murder rate while some states that were safer are experiencing sharp rises in murder rate.

The states of Coahuila, Nayarit, Sinaloa, Durango, and Chihuahua all experienced noticeable decreases in murder rate (the largest decreases being in the state of Chihuahua which dropped from 114.6 to 44.4 murders per 100,000 people, a drop of about 61% and the state of Durango which dropped from 62.7 to 12.3 murders per 100,000, a decrease of about 80%). These decreases are, unfortunately, overshadowed by some increases in some of the other states. The three most notable cases here are the states of Colima, Baja California Sur, and Zacetecas. These states respectively experienced increases of 588%, 949%, and 381% in their murder rates.

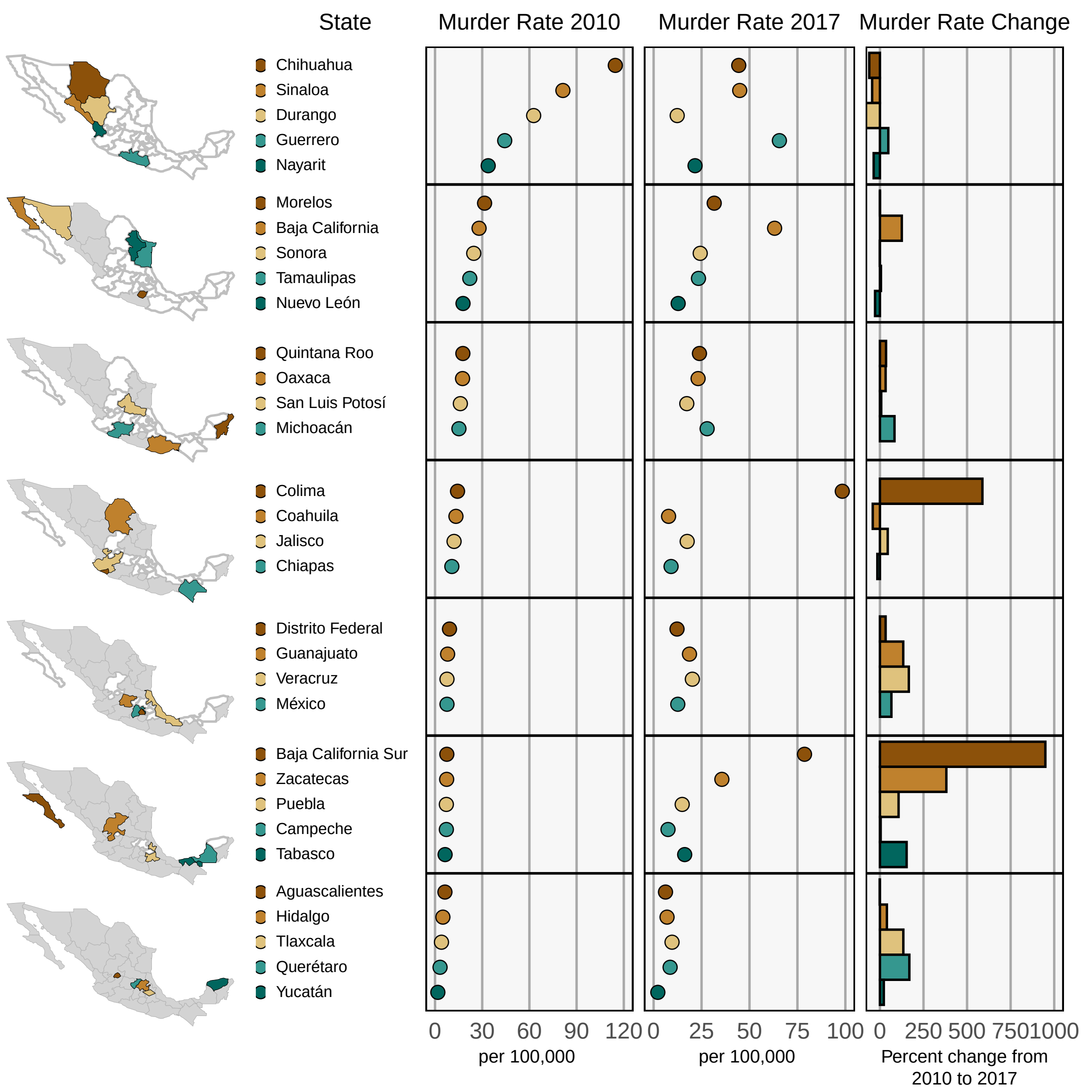


Fig. 7.5: Linked micromap plot comparing murder rates in 2010 and 2017 and the overall percent change in murder rate for each state in Mexico. Sorted by murder rate in 2010.

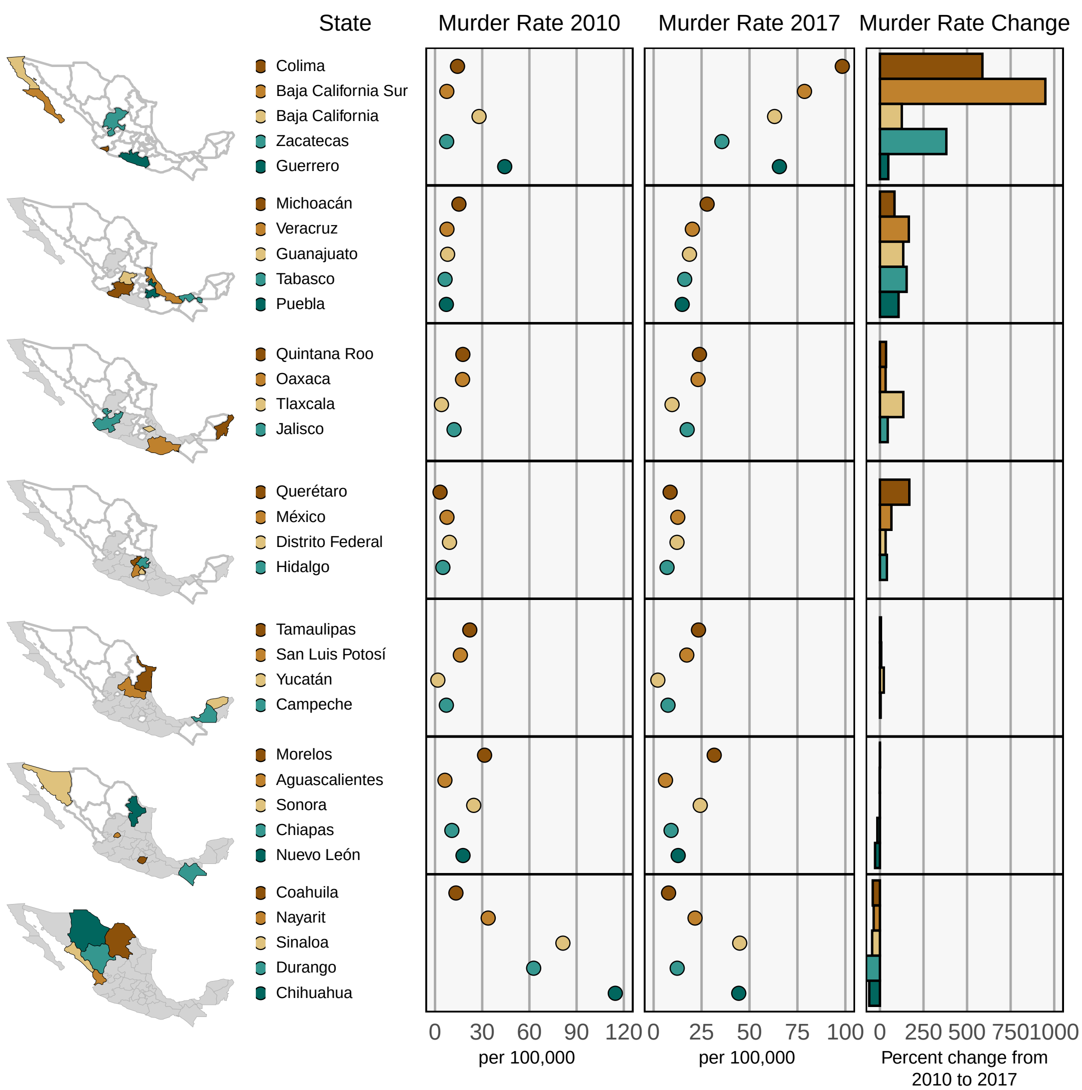


Fig. 7.6: Linked micromap plot comparing murder rates in 2010 and 2017 and the overall percent change in murder rate for each state in Mexico. Note that this plot is sorted by overall change in murder rate (not shown) and not percent change (shown).

CHAPTER 8

Conclusions and Future Work

This thesis has introduced the 'LMshapemaker' R package as a new tool for applying modifications and changes to regions and boundaries contained in shapefiles. With this R package, users have access to ready-to-use shapefiles for most of the countries in the world. These shapefiles were objectively created, however a user can choose to tailor their shapefile to be a more suitable representation from their own perspective. In either case, users can access shapefiles that are more readable and interpretable in map-based visualizations, particularly linked micromap plots.

As mentioned in Chapter 4, there are a few modifications that cannot be performed within the provided Shiny app. These modifications include manually filtering islands (Section 4.1.2) and separately modifying polygons of the same administrative district (Section 4.4). For the purpose of creating a repository of ready-to-use shapefiles, these modifications were rarely used and not initially included in the Shiny app, but will be added in the future.

Additionally, the scope of this research and R package only included modifications for the first administrative level of countries in the world. Thus, unfortunately, several countries were excluded from consideration due to having too few regions to create a meaningful LM plot. By applying the modifications listed in Chapter 4, shapefiles for the second level of administration could be prepared for the excluded countries, or even to provide more detailed options for the countries that exist in the package at this time. As mentioned in Section 5.2, the United Kingdom was excluded from the 'LMshapemaker' R package. Going beyond the first level of administration could result in meaningful shapefiles for the countries of England, Wales, Scotland, and Northern Ireland.

Countries that consist of many small land masses such as the Bahamas, the Philippines, and Indonesia pose problems for creating linked micromap plots as most of these small islands cannot be colored in distinguishable ways. There are often too many islands or land masses that resizing and/or shifting all of them would be burdensome and meaningless. Potential solutions to representing these countries in LM plots could involve creating custom shapefiles based on Voronoi tessellations or other geometric polygonal constructions that make use of the capital area or the geographic center in each administrative district in each area's construction. These custom shapefiles could be used in place of the maps one typically sees in the map panel of a LM plot.

In addition to modifying geopolitical boundaries, these techniques could also be used for non political divisions within a given region. For example, geological boundaries defining ecoregions, climate zones, or watershed boundaries could be processed and modified for a more meaningful display in context of LM plots.

References

Ahn, J. Y. (2013). Visualizing Statistical Information using Korean Linked Micromap Plots. In Cho, S.-H., editor, *Proceedings of IASC–Satellite Conference for the 59th ISI WSC & The 8th Conference of IASC–ARS*, pages 219–221. Asian Regional Section of the IASC. http://hbutton.com/IASC/iascproceedings/proceeding_pdf/Proceedings.pdf.

Bivand, R. S., Pebesma, E., and Gomez-Rubio, V. (2013). *Applied Spatial Data Analysis with R, Second Edition*. Springer, NY.

Bonnal, L., Favard, P., Laurent, T., and Ruiz-Gazen, A. (2011). Pourquoi le coût de l'éducation est–il plus élevé en zone rurale? Le cas de la région Midi–Pyrénées. *Revue d'Économie Régionale & Urbaine*, 2011/5:887–910. (In French).

Brewer, C. A. and Pickle, L. W. (2002). Comparison of Methods for Classifying Epidemiological Data on Choropleth Maps in Series. *Annals of the Association of American Geographers*, 92(4):662–681.

Carr, D. B., Chen, J., Bell, B. S., Pickle, L. W., and Zhang, Y. (2002). Interactive Linked Micromap Plots and Dynamically Conditioned Choropleth Maps. In *Proceedings of the Second National Conference on Digital Government Research*, pages 61–67. Digital Government Research Center (DGRC). http://dl.acm.org/citation.cfm?id=1123098.

Carr, D. B., Olsen, A. R., Courbois, J. P., Pierson, S. M., and Carr, D. A. (1998). Linked Micromap Plots: Named and Described. *Statistical Computing and Statistical Graphics Newsletter*, 9(1):24–32.

Carr, D. B. and Pierson, S. M. (1996). Emphasizing Statistical Summaries and Showing Spatial Context with Micromaps. *Statistical Computing and Statistical Graphics Newsletter*, 7(3):16–23.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2018). *shiny: Web Application Framework for R*. R package version 1.2.0.

Dempsey, C. (2016). What is a Shapefile? *GIS Lounge*. https://www.gislounge.com/what-is-a-shapefile/.

Dorling, D. (1995). *A New Social Atlas of Great Britain*. John Wiley and Sons, New York, NY.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *The Canadian Cartographer*, 10(2):112–122.

Ellis, P. (2017). Luke-Warm about Micromaps. http://freerangestats.info/blog/2017/04/30/micromaps.

ESRI (1997). ESRI Shapefile Technical Description. *An ESRI White Paper*. https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

Global Administrative Areas (2012). GADM Database of Global Administrative Areas. https://gadm.org/.

Han, K. S., Park, S. J., Mun, G. S., Choi, S. H., Symanzik, J., Gebreab, S., and Ahn, J. Y. (2014). Linked Micromaps for the Visualization of Geographically Referenced Data. *ICIC Express Letters*, 8(2):443–448.

Harrower, M. and Bloch, M. (2006). MapShaper.org: A Map Generalization Web Service. *IEEE Computer Graphics and Applications*, 26(4):22–27.

McManus, M. G., Pond, G. J., Reynolds, L., and Griffith, M. B. (2016). Multivariate Condition Assessment of Watersheds with Linked Micromaps. *Journal of America Water Resources Association*, 52(3):494–507.

Monmonier, M. (1993). *Mapping It Out: Expository Cartography for the Humanities and Social Sciences*. University of Chicago Press, Chicago, IL.

Monmonier, M. (1996). *How to Lie with Maps (Second Edition)*. University of Chicago Press, Chicago, IL.

Olsen, A. R., Carr, D. B., Courbois, J. P., and Pierson, S. M. (1996). Presentation of Data in Linked Attribute and Geographic Space. In *1996 Abstracts, Joint Statistical Meetings, Chicago, Illinois*, page 271, Alexandria, VA. American Statistical Association.

Payton, Q. C., McManus, M. G., Weber, M. H., Olsen, A. R., and Kincaid, T. M. (2015). micromap: A Package for Linked Micromaps. *Journal of Statistical Software*, 63(2):1–16.

Pickle, L. W., Pearson Jr., J. B., and Carr, D. B. (2015). micromapST: Exploring and Communicating Geospatial Patterns in U.S. State Data. *Journal of Statistical Software*, 63(3). http://www.jstatsoft.org/v63/i03/.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rosenbaum, A. S., Axelrad, D. A., Woodruff, T. J., Wei, Y.-H., Ligocki, M. P., and Cohen, J. P. (1999). National Estimates of Outdoor Air Toxics Concentrations. *Journal of the Air and Waste Management Association*, 49:1138–1152.

Singapore Management University (2017). The Indian Story Report. https://wiki.smu.edu.sg/1617t3isss608g1/The_Indian_Story_Report.

Symanzik, J., Bao, S., Dai, X., Shui, M., and She, B. (2016). Recent Advancements in Geovisualization, with a Case Study on Chinese Religions. In Jin, Z., Liu, M., and Luo, X., editors, *New Developments in Statistical Modeling, Inference and Application: Selected Papers from the 2014 ICSA/KISS Joint Applied Statistics Symposium in Portland, OR*, pages 151–166, Switzerland. Springer.

Symanzik, J. and Carr, D. B. (2008). Interactive Linked Micromap Plots for the Display of Geographically Referenced Statistical Data. In Chen, C., Härdle, W., and Unwin, A., editors, *Handbook of Data Visualization*, pages 267–294 & 2 Color Plates. Springer, Berlin, Heidelberg.

Symanzik, J., Dai, X., Weber, M. H., Payton, Q., and McManus, M. G. (2014). Linked Micromap Plots for South America — General Design Considerations and Specific Adjustments. *Revista Colombiana de Estadística*, 37(2):451–469.

Teucher, A. and Russell, K. (2018). *rmapshaper: Client for 'mapshaper' for 'Geospatial' Operations*. R package version 0.4.0.

United Nations (2019). Member States. https://www.un.org/en/member-states/.

Wainer, H. and Francolini, C. M. (1980). An Empirical Inquiry Concerning Human Understanding of Two–Variable Color Maps. *The American Statistician*, 34(2):81–93.

Wang, X., Chen, J. X., Carr, D. B., Bell, B. S., and Pickle, L. W. (2002). Geographic Statistics Visualization: Web–based Linked Micromap Plots. *Computing in Science & Engineering*, 4(3):90–94.

Wickham, H., Hester, J., and Chang, W. (2019). *devtools: Tools to Make Developing R Packages Easier*. R package version 2.2.0.

APPENDICES

APPENDIX A

List of Country Modifications

Included in this appendix are the modifications that were done for each country included in the 'LMshapemaker' R Package, arranged by continent. For each country (with some exceptions due to limitations discussed in Section 6.2, denoted by an asterisk), output from the 'LMshapemaker' Shiny app shows the modification table and the shapefile before and after modifications are applied. Modified regions are highlighted in red. Commentary describing the modifications is also included.

A.1 Africa

Table A.1: Table listing African countries modified and included in the 'LMshapemaker' R package.

Country	Page Number
Angola	62
Benin	63
Botswana	64
Burkina Faso	65
Burundi	66
Cameroon	67
Central African Republic	68
Republic of Congo	69
Democratic Republic of the Congo	70
Equitorial Guinea	71
Eritea	72
Ethiopia	73
Gabon	74
Gambia	75
Ghana	76
Guinea	77
Guinea - Bissau	78
Ivory Coast	79
Lesotho	80
Liberia	81
Libya	82
Madagascar	83
Malawi	84

Continued on next page

Table A.1 – continued from previous page

The following countries have been excluded from the 'LMshapemaker' R package for the reasons indicated.

- Algeria, Cape Verde, Chad, Egypt, Kenya, Mauritius, and the Seychelles, and Uganda were excluded for consisting complicated geography.
- Comoros, Djibouti, Eswatini, Rwanda, Saõ Tomé and Príncipe, Sierra Leone, and Togo were excluded for containing five or less first level administrative districts.

Angola

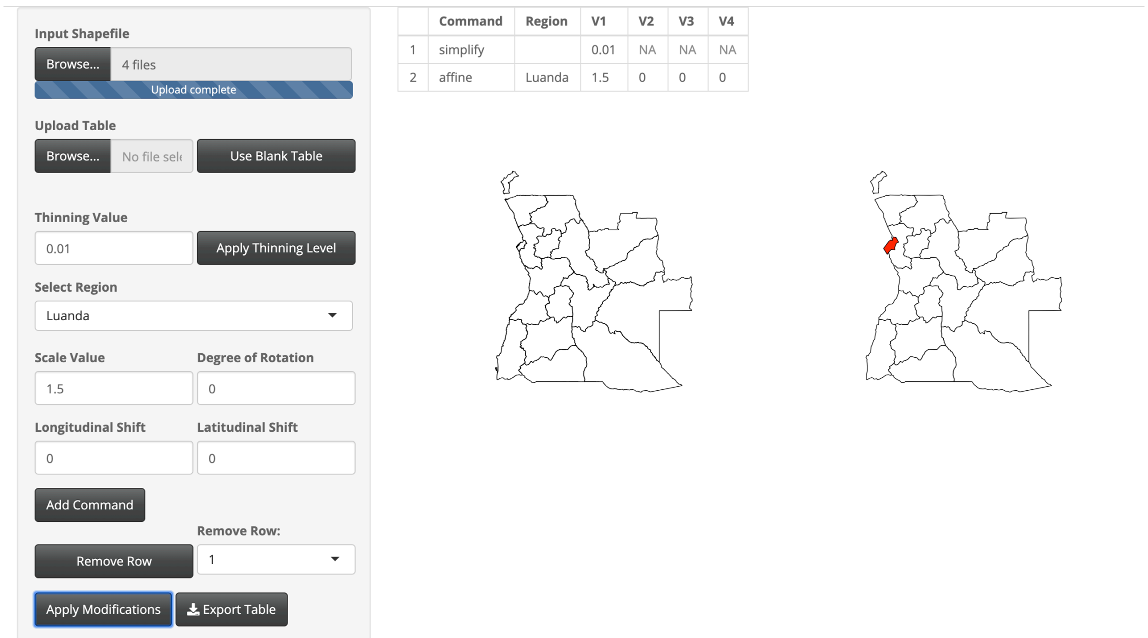


Fig. A.1: Modifications done to Angola.

Thinning was done to keep 1% of the original boundary points. The Luanda region was scaled larger with a scale factor of 1.5.

Benin

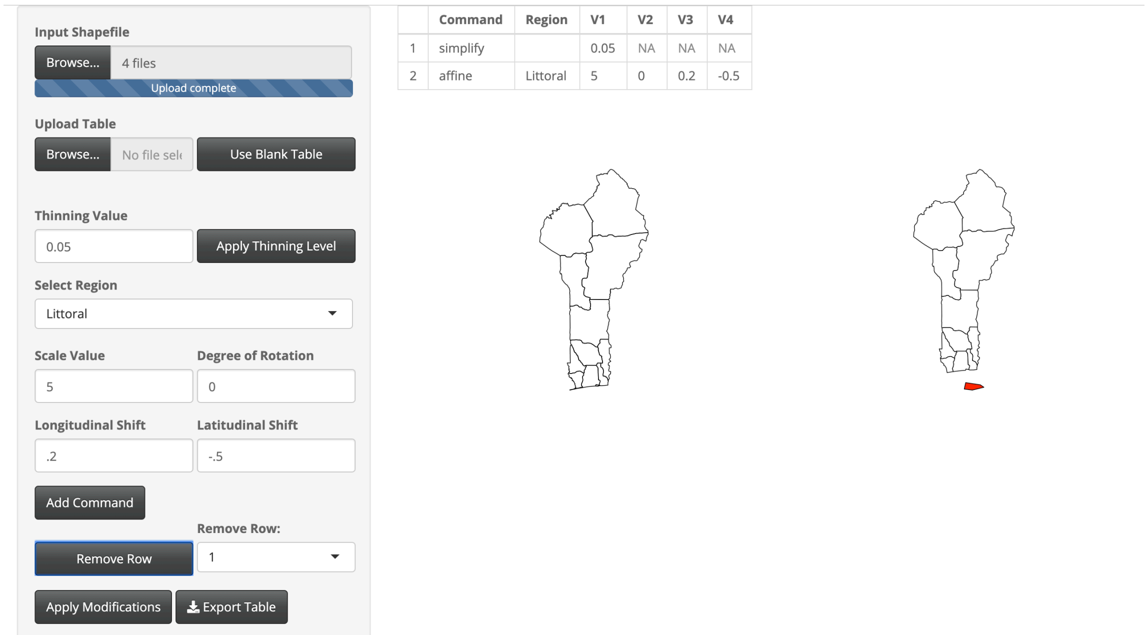
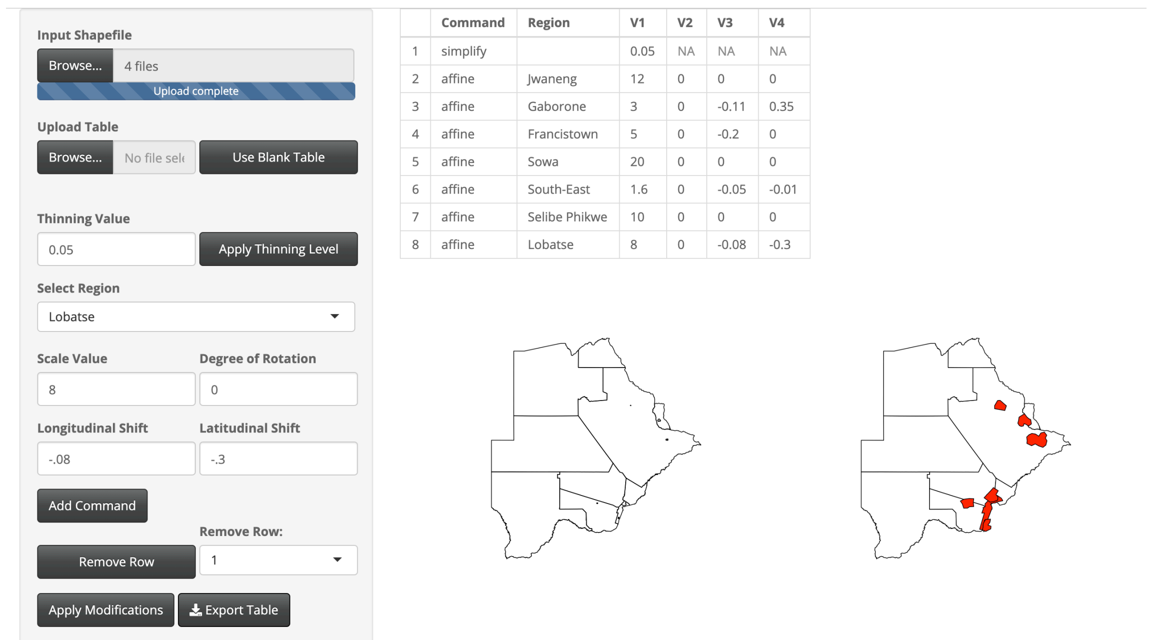


Fig. A.2: Modifications done to Benin.

Thinning was done to keep 5% of the original boundary points. The Littoral region was scaled larger with a scale factor of 5, shifted east 0.2 degrees longitude, and south 0.5 degrees latitude.

Botswana



	Command	Region	V1	V2	V3	V4
1	simplify		0.05	NA	NA	NA
2	affine	Jwaneng	12	0	0	0
3	affine	Gaborone	3	0	-0.11	0.35
4	affine	Francistown	5	0	-0.2	0
5	affine	Sowa	20	0	0	0
6	affine	South-East	1.6	0	-0.05	-0.01
7	affine	Selibe Phikwe	10	0	0	0
8	affine	Lobatse	8	0	-0.08	-0.3

Fig. A.3: Modifications done to Botswana.

Thinning was done to keep 5% of the original boundary points. The Jwaneng region was scaled larger with a scale factor of 12. The Gaborone region was scaled larger with a scale factor of 3, shifted west 0.11 degrees longitude, and shifted north 0.35 degrees latitude. The Francistown region was scaled larger with a scale factor of 5 and shifted west 0.2 degrees longitude. The Sowa region was scaled larger with a scale factor of 20. The South-East regions was scaled larger with a scale factor of 1.6, shifted west 0.5 degrees longitude, and south 0.01 degrees latitude. The Selibe Phikwe region was scaled larger with a scale factor of 10. The Lobatse region was scaled larger with a scale factor of 8, shifted west 0.08 degrees longitude, and south 0.3 degrees latitude.

Burkina Faso

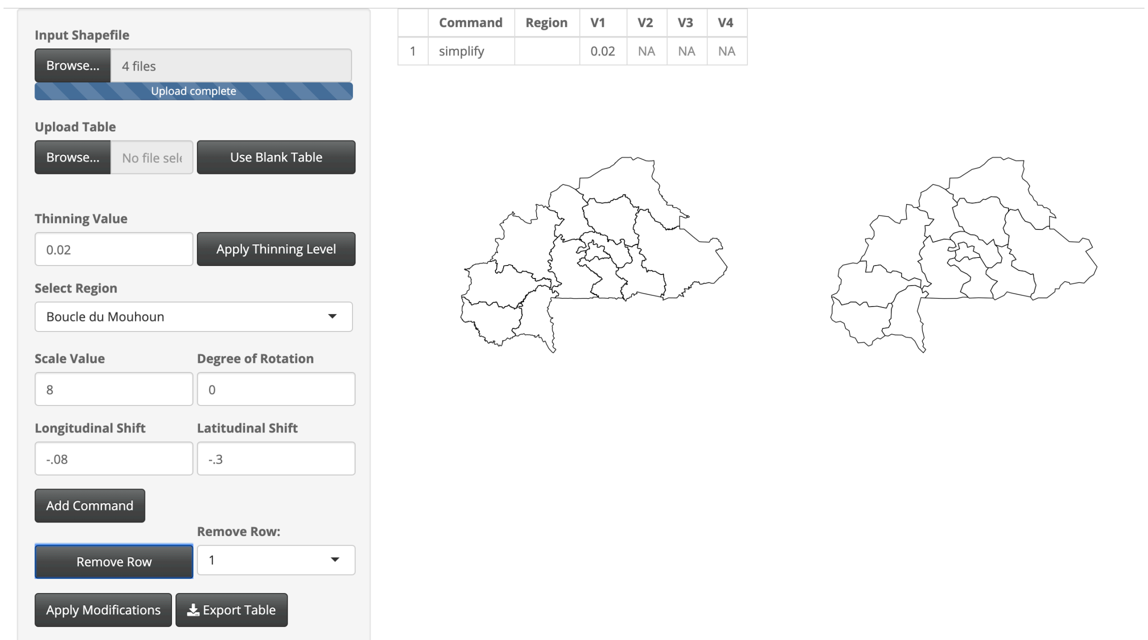


Fig. A.4: Modifications done to Burkina Faso.

Thinning was done to keep 2% of the original boundary points.

Burundi

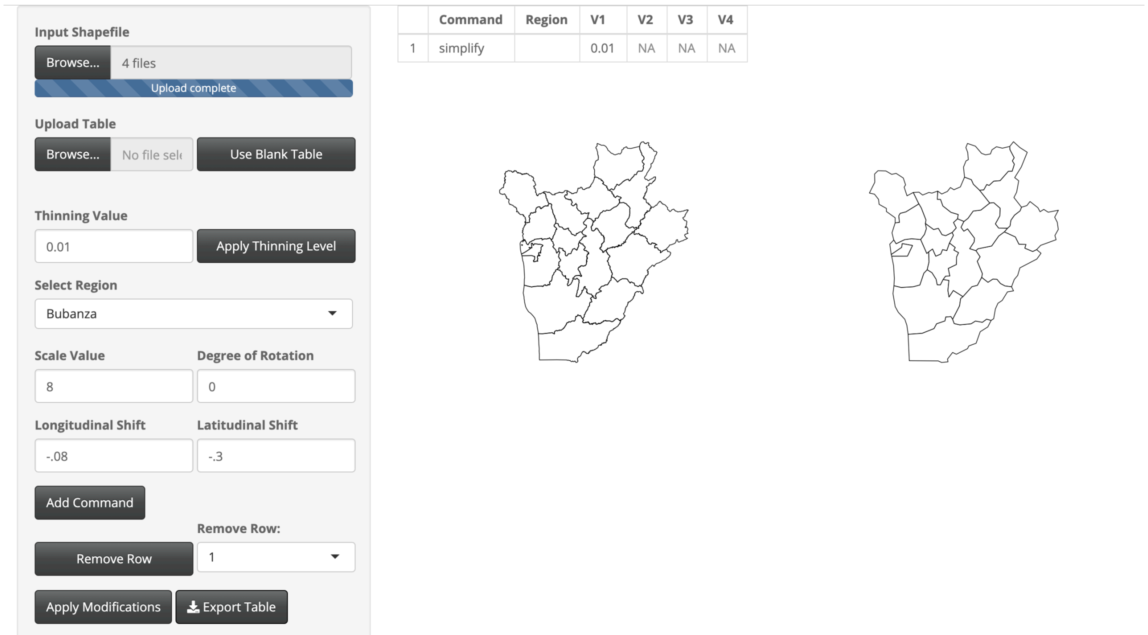


Fig. A.5: Modifications done to Burundi.

Thinning was done to keep 1% of the original boundary points.

Cameroon

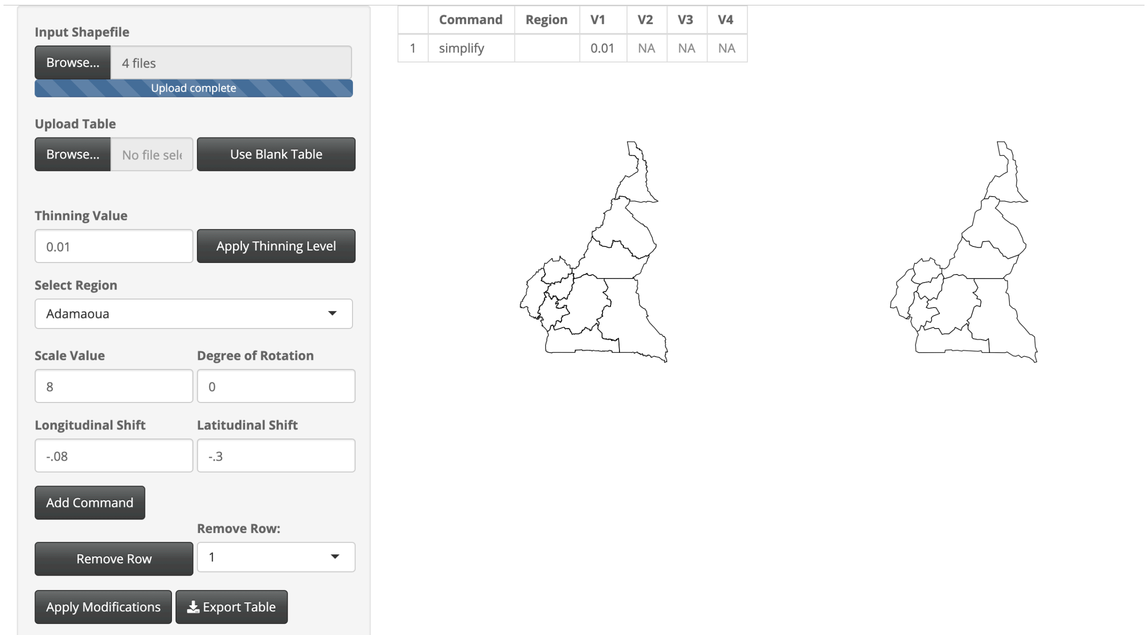


Fig. A.6: Modifications done to Cameroon.

Thinning was done to keep 1% of the original boundary points.

Central African Republic

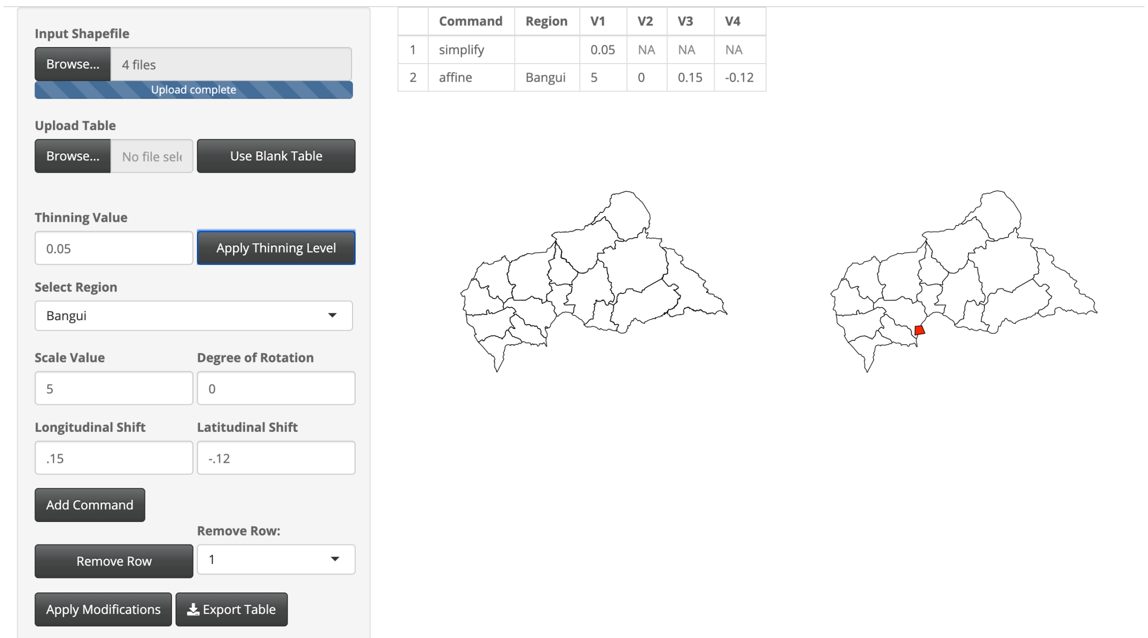


Fig. A.7: Modifications done to the Central African Republic.

Thinning was done to keep 5% of the original boundary points. The Bangui region was scaled larger with a scale factor of 5, shifted east 0.15 degrees longitude, and south 0.12 degrees latitude.

Republic of the Congo

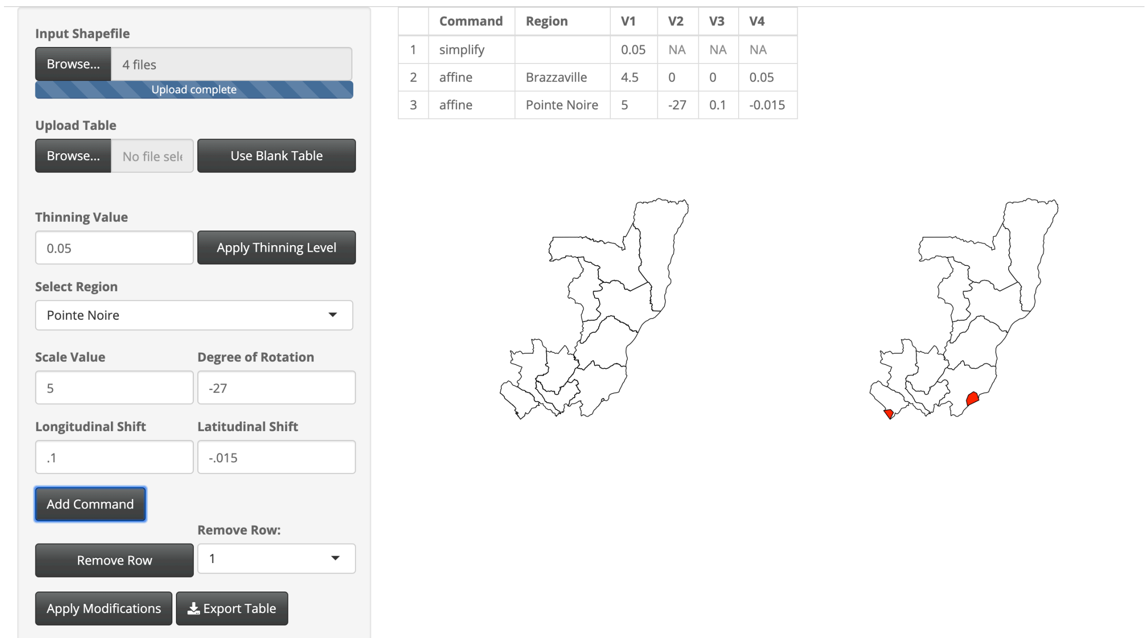


Fig. A.8: Modifications done to the Republic of the Congo.

Thinning was done to keep 5% of the original boundary points. The Brazzaville region was scaled larger with a scale factor of 4.5 and shifted north 0.05 degrees latitude. The Pointe Noire region was scaled larger with a scale factor of 5, rotated 27 degrees counter-clockwise, shifted east 0.1 degrees longitude, and south 0.015 degrees latitude.

Democratic Republic of the Congo

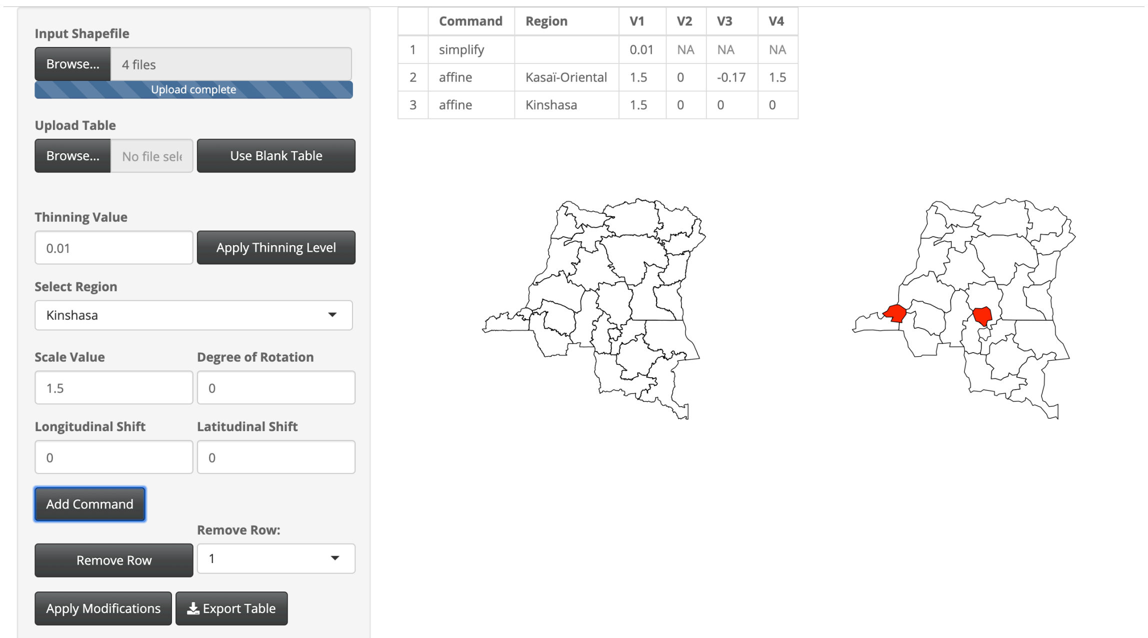


Fig. A.9: Modifications done to the Democratic Republic of the Congo.

Thinning was done to keep 1% of the original boundary points. The Kasai-Oriental region was scaled larger with a scale factor of 1.5, shifted west 0.17 degrees longitude, and north 1.5 degrees latitude. The Kinshasa region was scaled larger with a scale factor of 1.5.

Equitorial Guinea

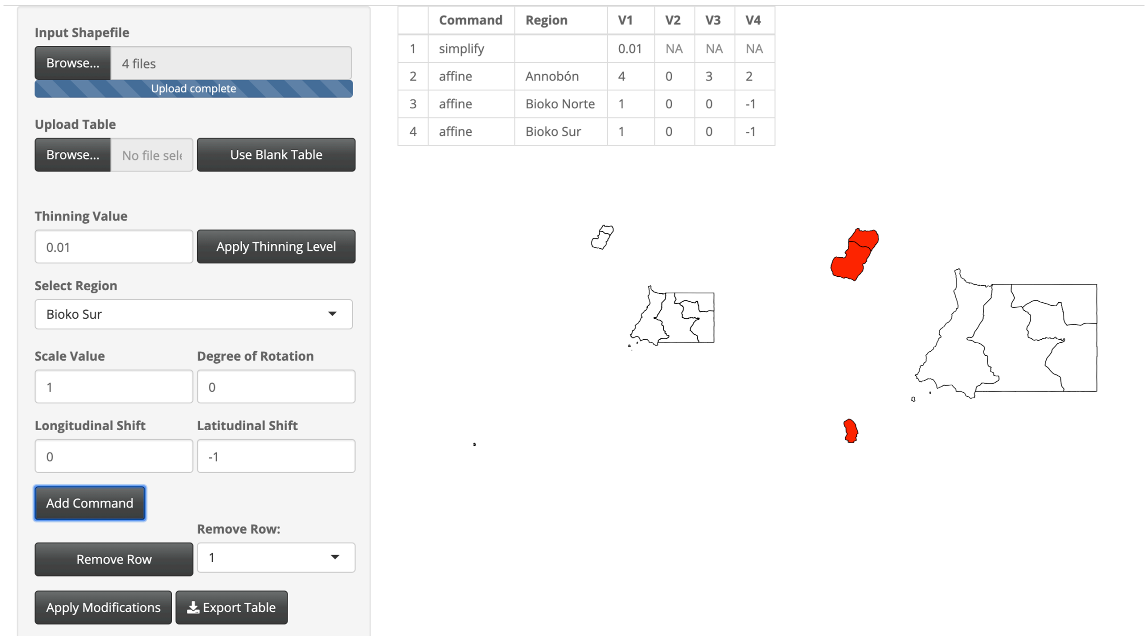
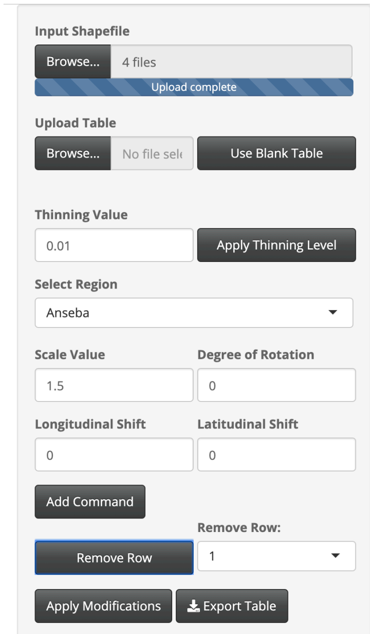


Fig. A.10: Modifications done to Equitorial Guinea.

Thinning was done to keep 1% of the original boundary points. The Annobón region was scaled larger with a scale factor of 4, shifted east 3 degrees longitude, and shifted north 2 degrees latitude. The Bioko Norte region was shifted south 1 degrees latitude. The Bioko Sur region was shifted south 1 degrees latitude.

Eritea



	Command	Region	V1	V2	V3	V4
1	simplify		0.01	NA	NA	NA
2	affine	Anseba	1.5	0	0	0

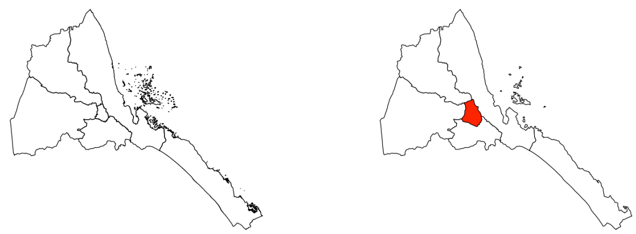
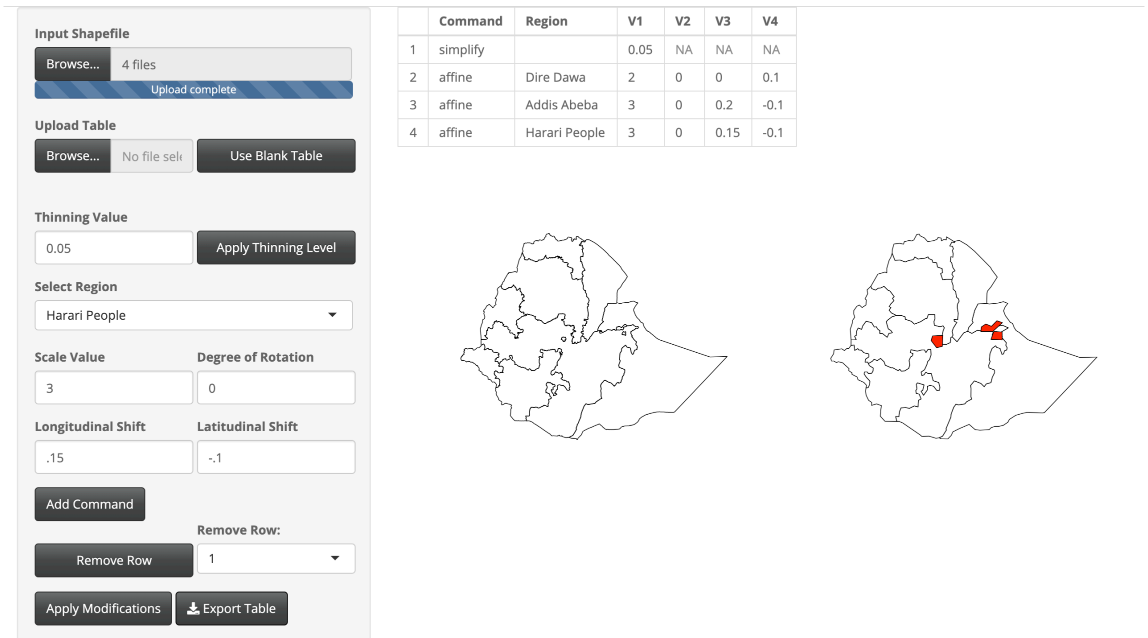


Fig. A.11: Modifications done to Eritea.

Thinning was done to keep 1% of the original boundary points. The Anseba region was scaled larger with a scale factor of 1.5.

Ethiopia



	Command	Region	V1	V2	V3	V4
1	simplify		0.05	NA	NA	NA
2	affine	Dire Dawa	2	0	0	0.1
3	affine	Addis Abeba	3	0	0.2	-0.1
4	affine	Harari People	3	0	0.15	-0.1

Fig. A.12: Modifications done to Ethiopia.

Thinning was done to keep 5% of the original boundary points. The Dire Dawa region was scaled larger with a scale factor of 2 and shifted north 3 degrees latitude. The Addis Abeba region was scaled larger with a scale factor of 3, shifted east 0.2 degrees longitude, and south 0.1 degrees latitude. The Harari People region was scaled larger with a scale factor of 3, shifted east 0.15 degrees longitude, and south 0.1 degrees latitude.

Gabon

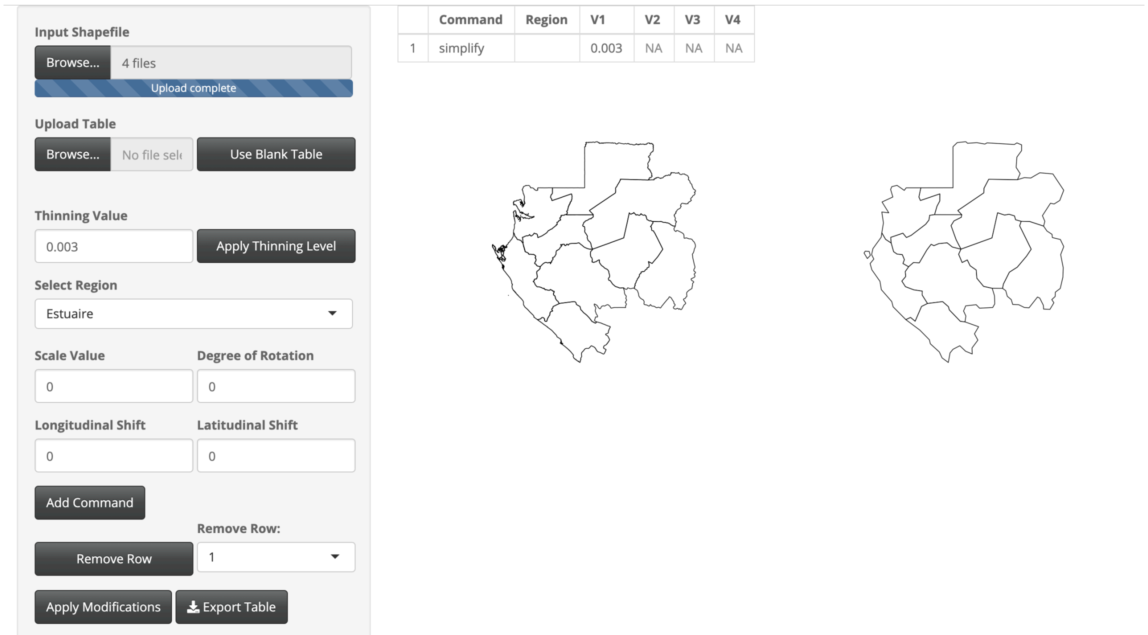


Fig. A.13: Modifications done to Gabon.

Thinning was done to keep 0.3% of the original boundary points.

Gambia

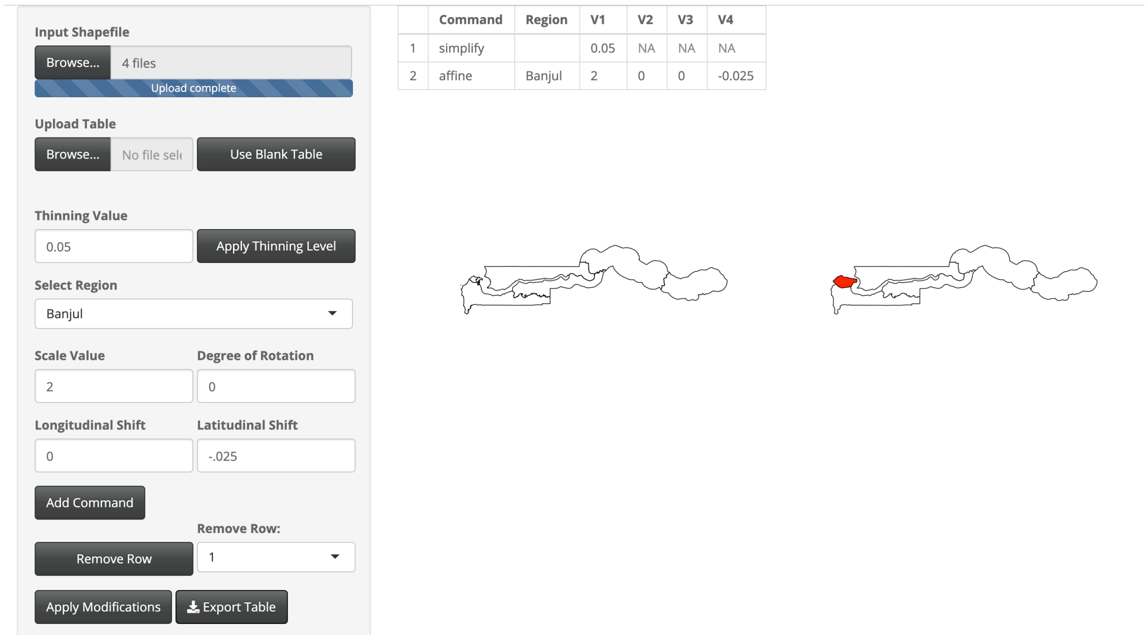


Fig. A.14: Modifications done to Gambia.

Thinning was done to keep 5% of the original boundary points. The Banjul region was scaled larger with a scale factor of 2 and shifted south 0.025 degrees latitude.

Ghana

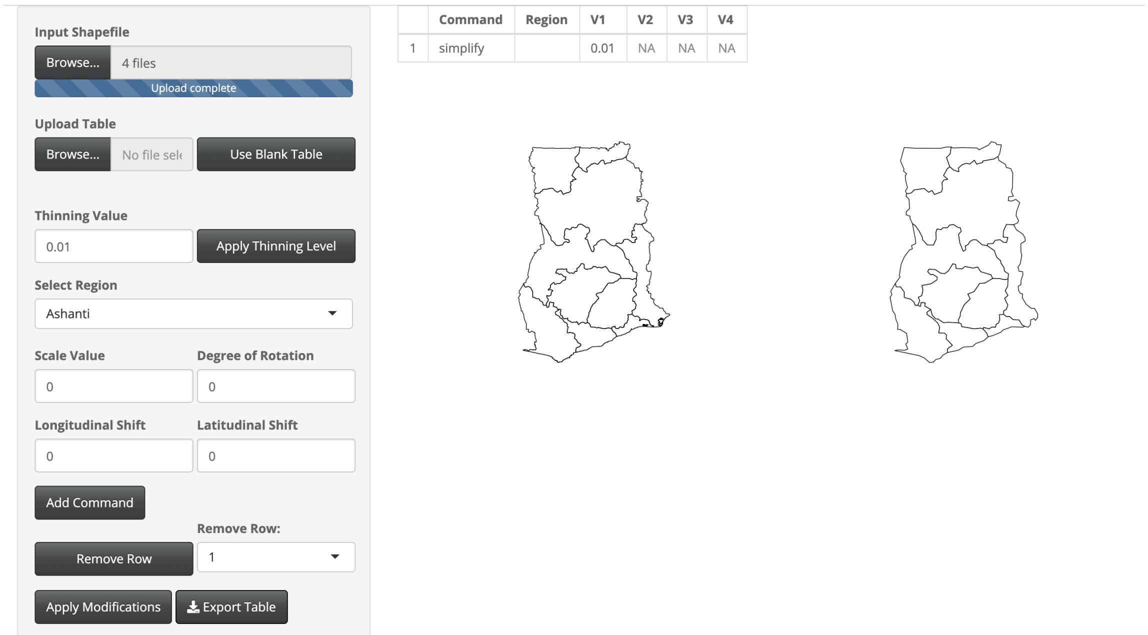


Fig. A.15: Modifications done to Ghana.

Thinning was done to keep 1% of the original boundary points.

Guinea

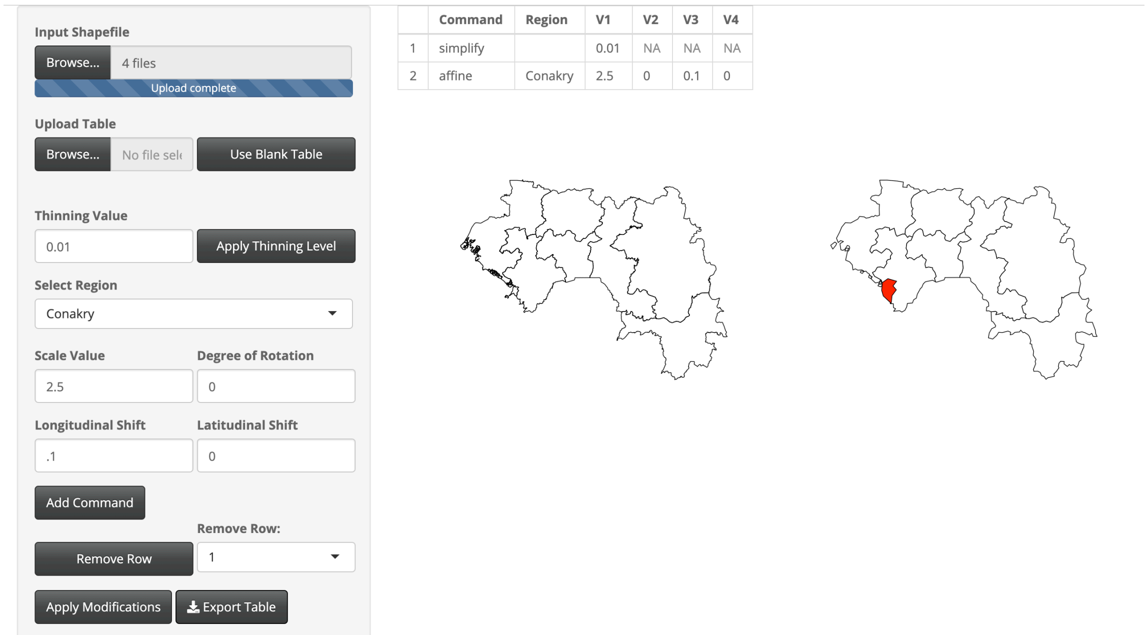


Fig. A.16: Modifications done to Guinea.

Thinning was done to keep 1% of the original boundary points. The Conakry region was scaled larger with a scale factor of 2.5 and shifted east 0.1 degrees longitude.

Guinea - Bissau

Fig. A.17: Modifications done to Guinea-Bissau.

Thinning was done to keep 0.9% of the original boundary points. The Bissau region was scaled larger with a scale factor of 2, rotated 5 degrees clockwise, shifted east 0.045 degrees longitude and shifted north 0.05 degrees latitude.

Ivory Coast

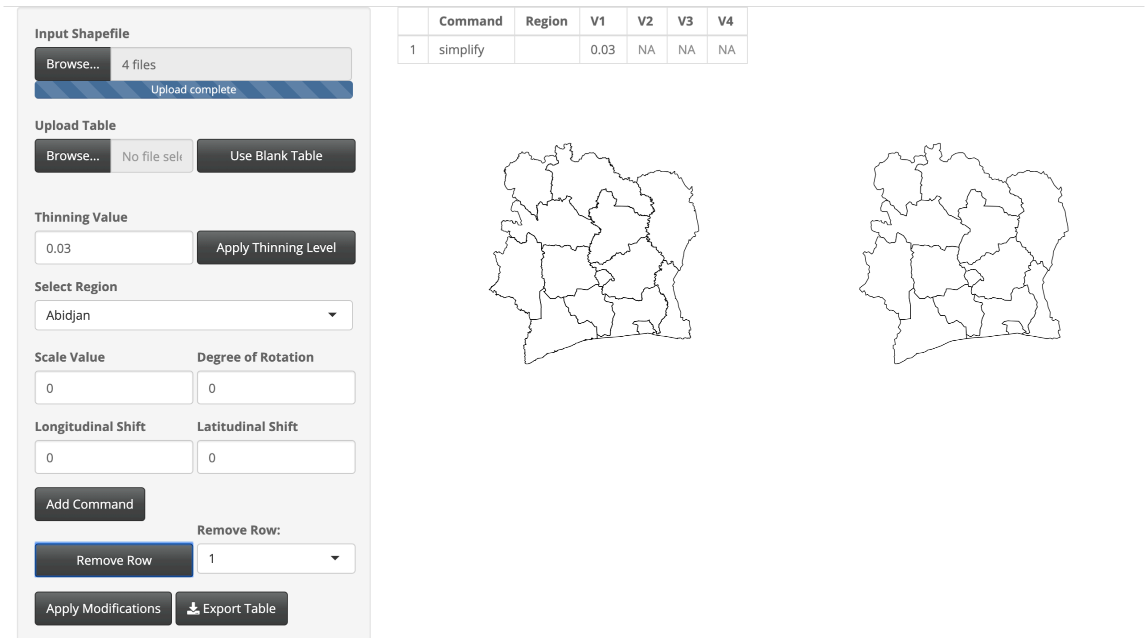


Fig. A.18: Modifications done to the Ivory Coast.

Thinning was done to keep 3% of the original boundary points.

Lesotho

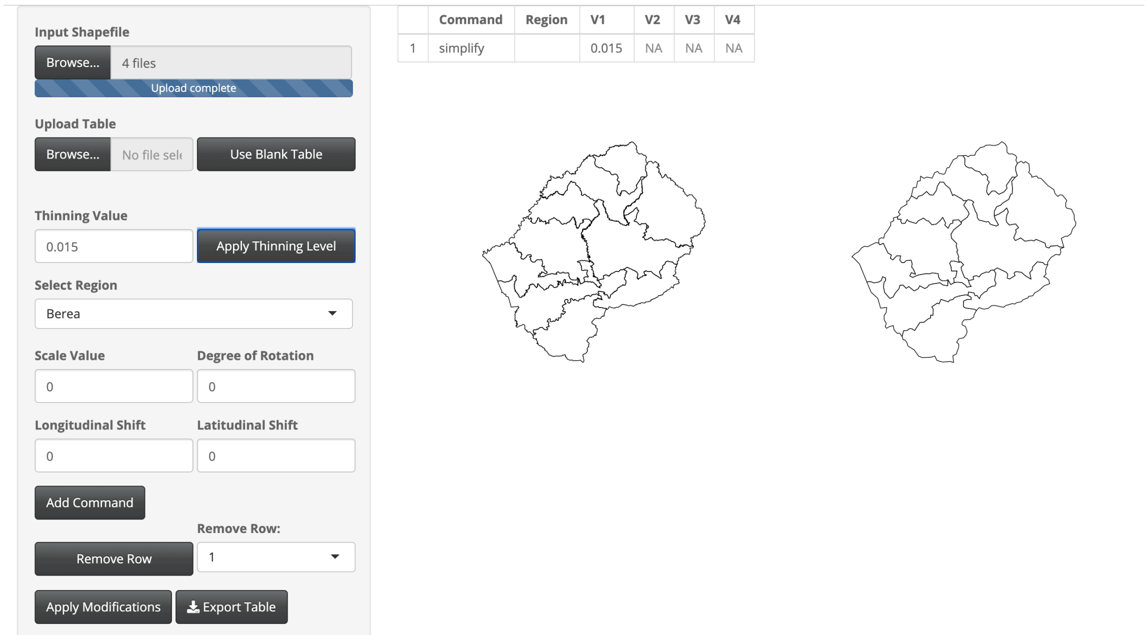


Fig. A.19: Modifications done to Lesotho.

Thinning was done to keep 1.5% of the original boundary points.

Liberia

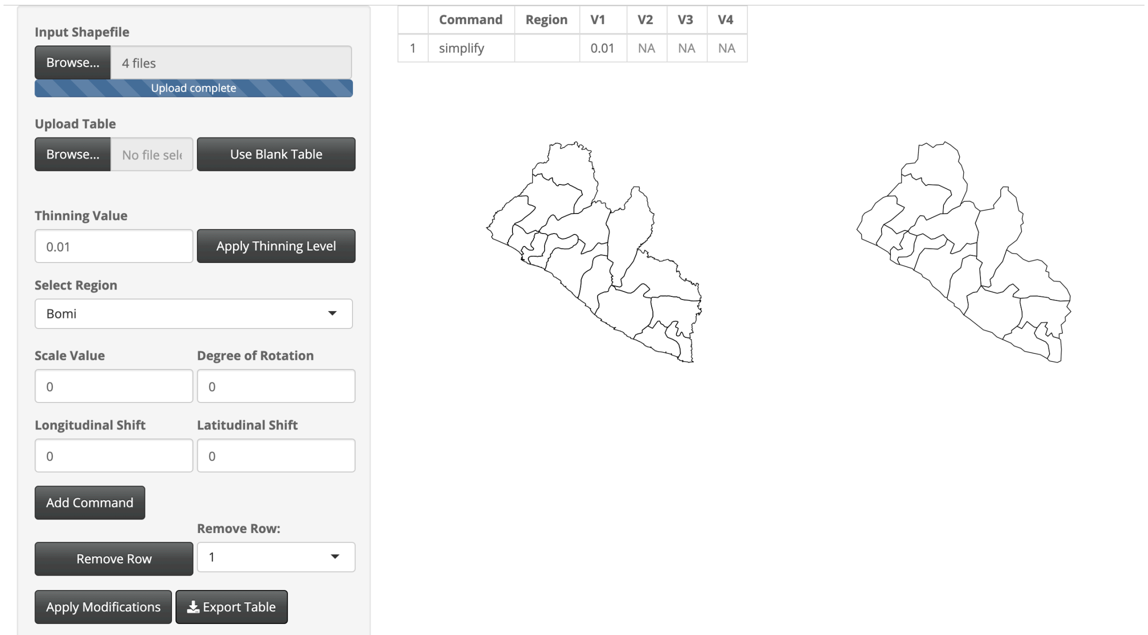


Fig. A.20: Modifications done to Liberia.

Thinning was done to keep 1% of the original boundary points.

Libya

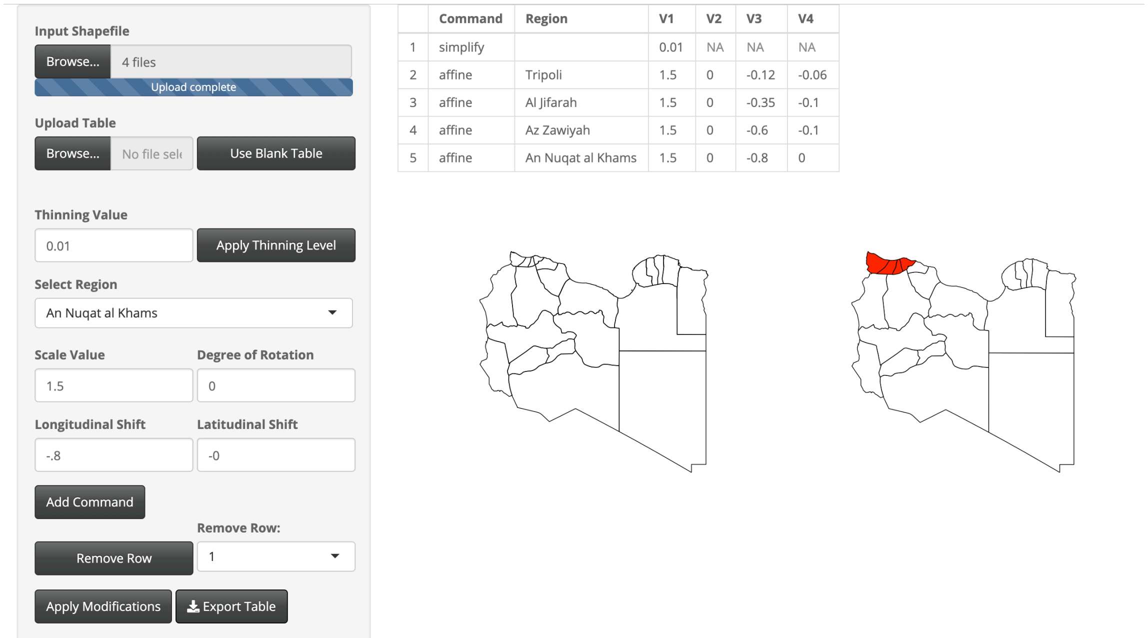


Fig. A.21: Modifications done to Libya.

Thinning was done to keep 1% of the original boundary points. The Tripoli region was scaled larger with a scale factor of 1.5, shifted west 0.12 degrees longitude, and shifted south 0.06 degrees latitude. The Al Jifarah region was scaled larger with a scale factor of 1.5, shifted west 0.35 degrees longitude and south 0.1 degrees latitude. The Az Zawiya region was scaled larger with a scale factor of 1.5, shifted west 0.6 degrees longitude, and south 0.1 degrees latitude. The An Nuqat al Khams region was scaled larger with a scale factor of 1.5 and shifted west 0.8 degrees longitude.

Madagascar

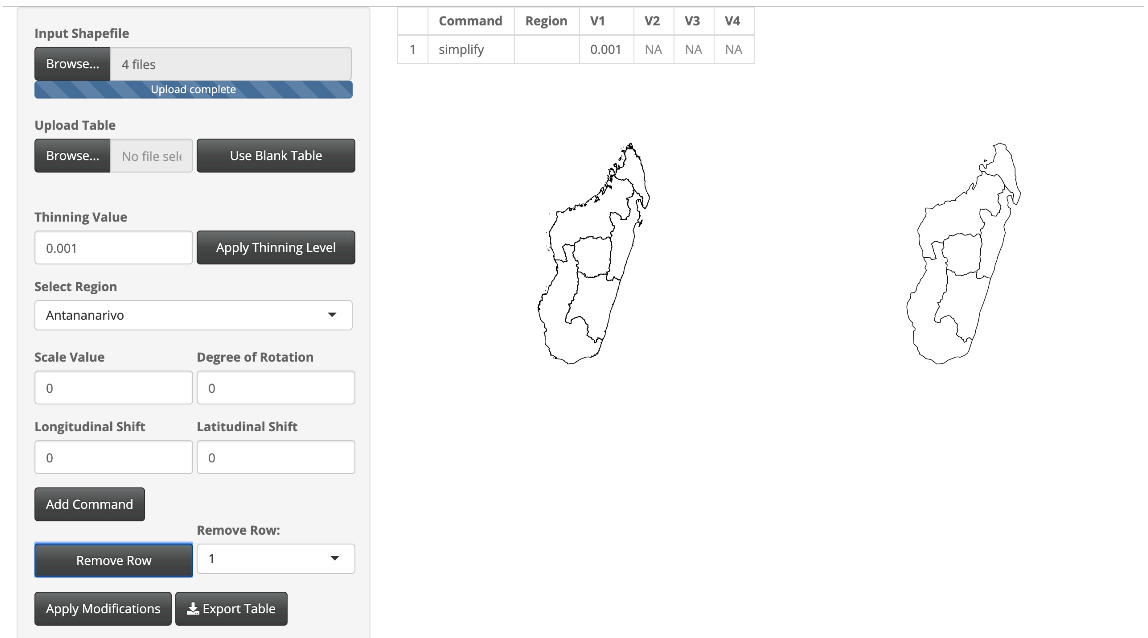


Fig. A.22: Modifications done to Madagascar.

Thinning was done to keep 1% of the original boundary points.

Malawi

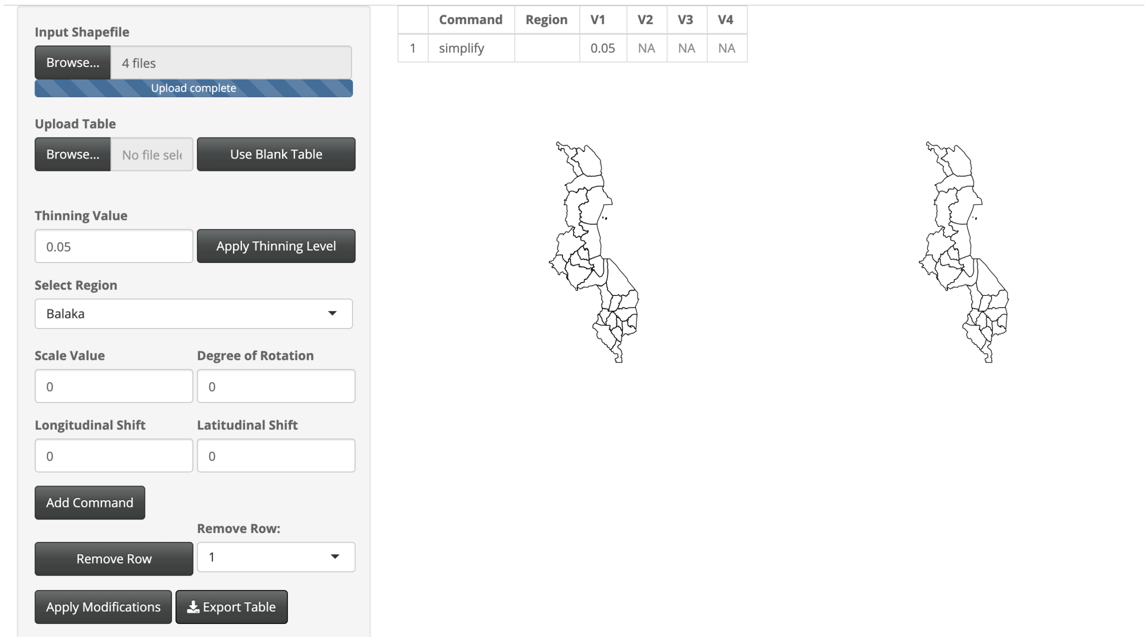


Fig. A.23: Modifications done to Malawi.

Thinning was done to keep 5% of the original boundary points.

Mali

.

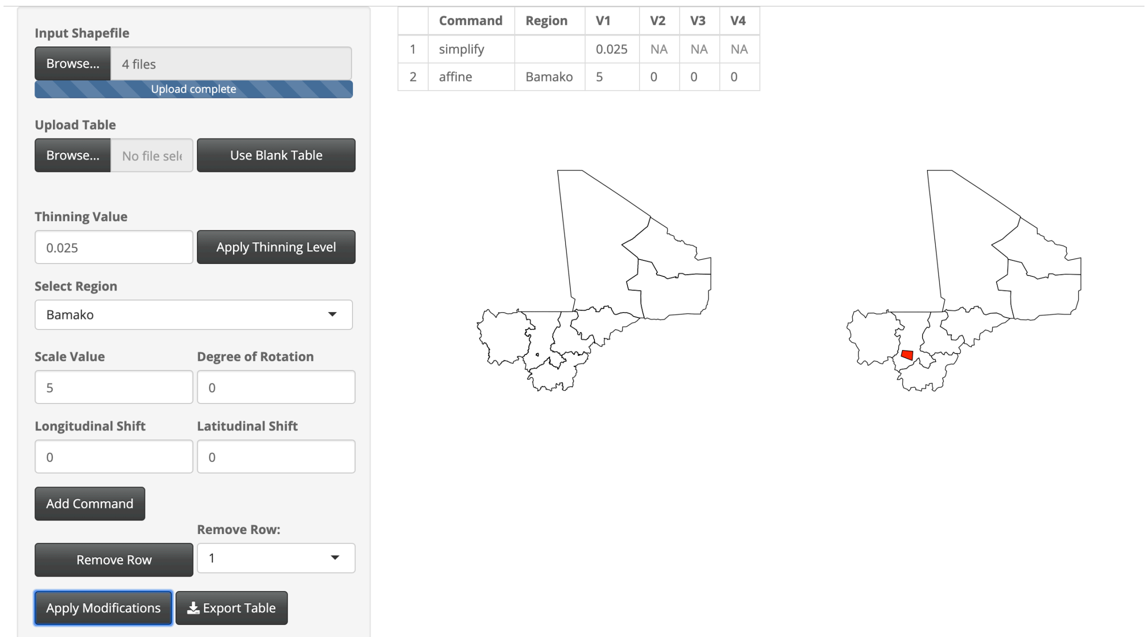


Fig. A.24: Modifications done to Mali.

Thinning was done to keep 2.5% of the original boundary points. The Barnako region was scaled larger with a scale factor of 5.

Mauritania

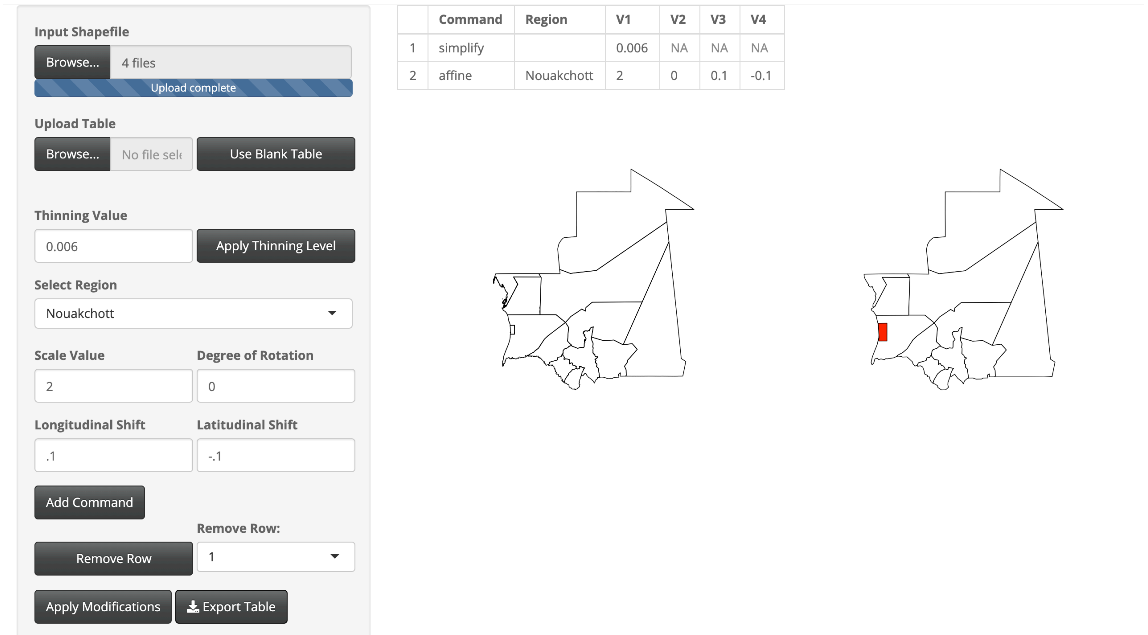


Fig. A.25: Modifications done to Mauritania.

Thinning was done to keep 0.6% of the original boundary points. The Nouakchott region was scaled larger with a scale factor of 2, shifted east 0.1 degrees longitude, and shifted south 0.1 degrees latitude.

Morocco

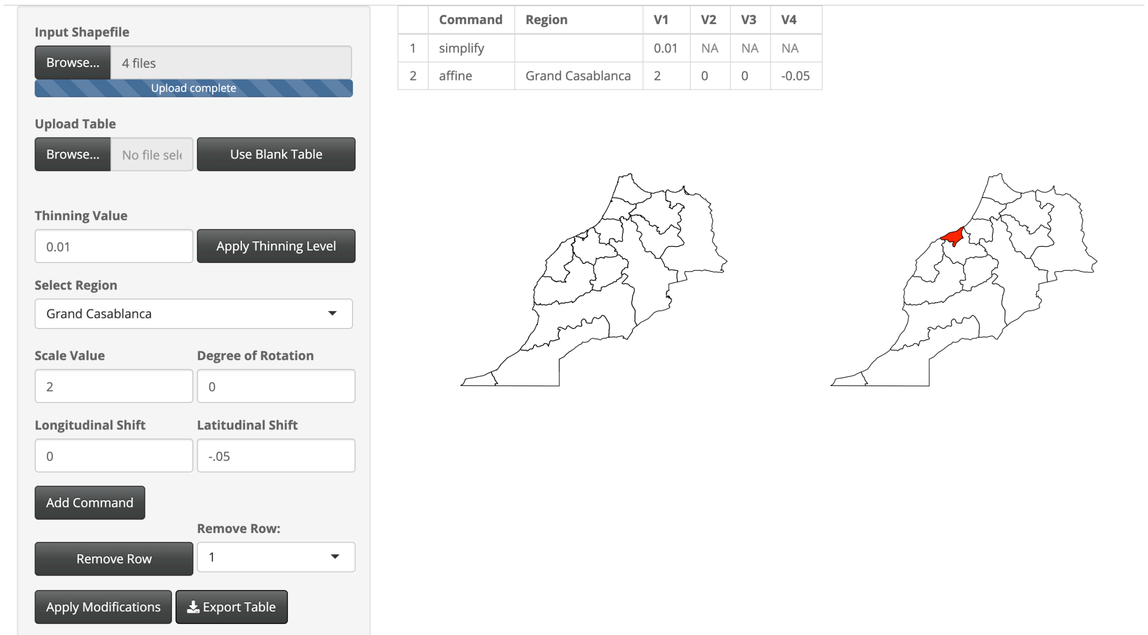


Fig. A.26: Modifications done to Morocco.

Thinning was done to keep 1% of the original boundary points. The Grand Casablanca region was scaled larger with a scale factor of 2 and shifted south 0.05 degrees latitude.

Mozambique

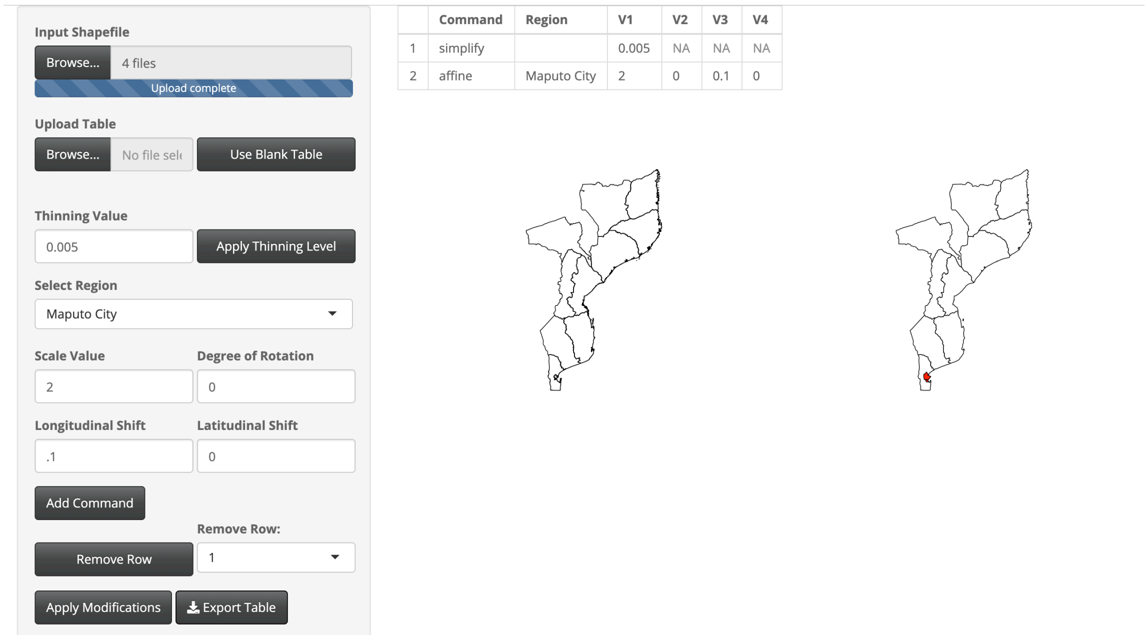


Fig. A.27: Modifications done to Mozambique.

Thinning was done to keep 0.5% of the original boundary points. The Maputo City region was scaled larger with a scale factor of 2 and shifted east 0.1 degrees longitude.

Namibia

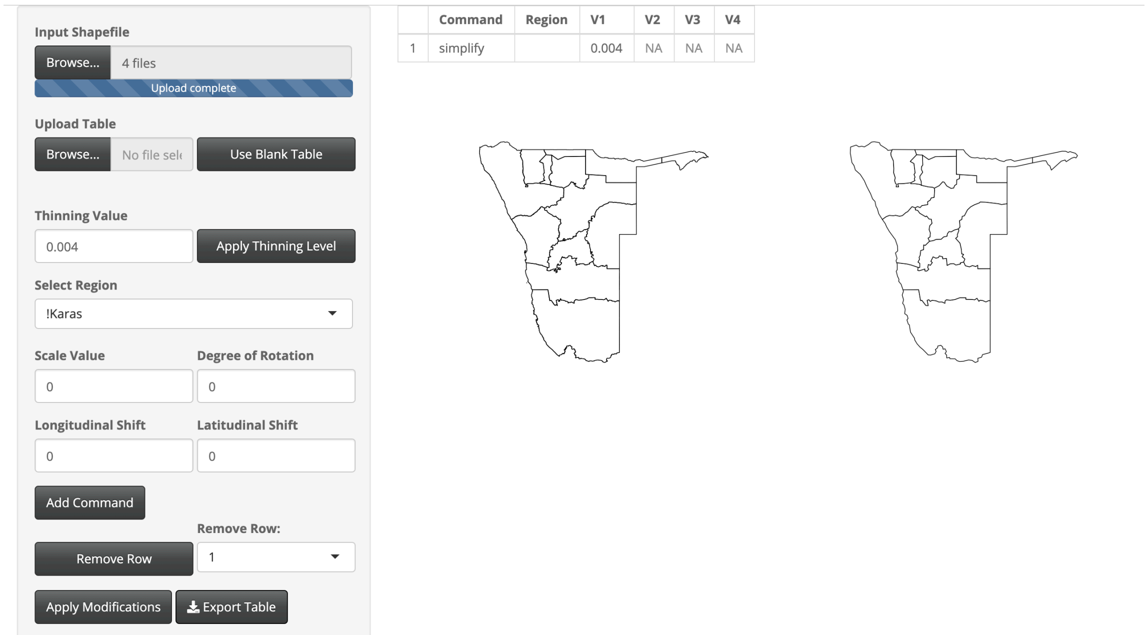


Fig. A.28: Modifications done to Namibia.

Thinning was done to keep 0.4% of the original boundary points.

Niger

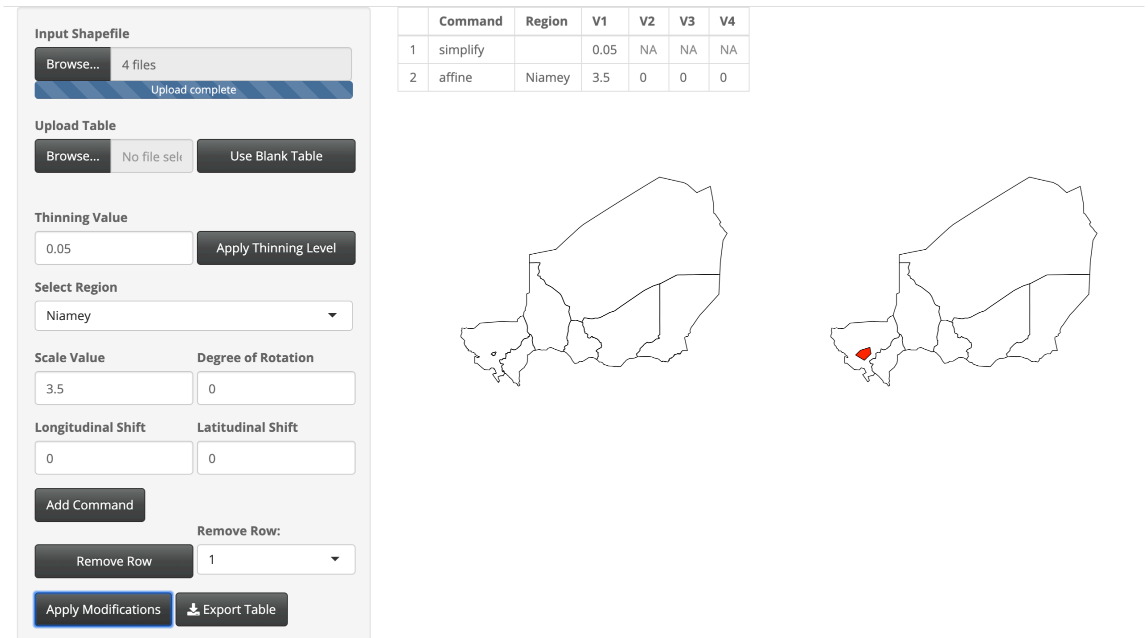


Fig. A.29: Modifications done to Niger.

Thinning was done to keep 5% of the original boundary points. The Niamey region was scaled larger with a scale factor of 3.5.

Nigeria

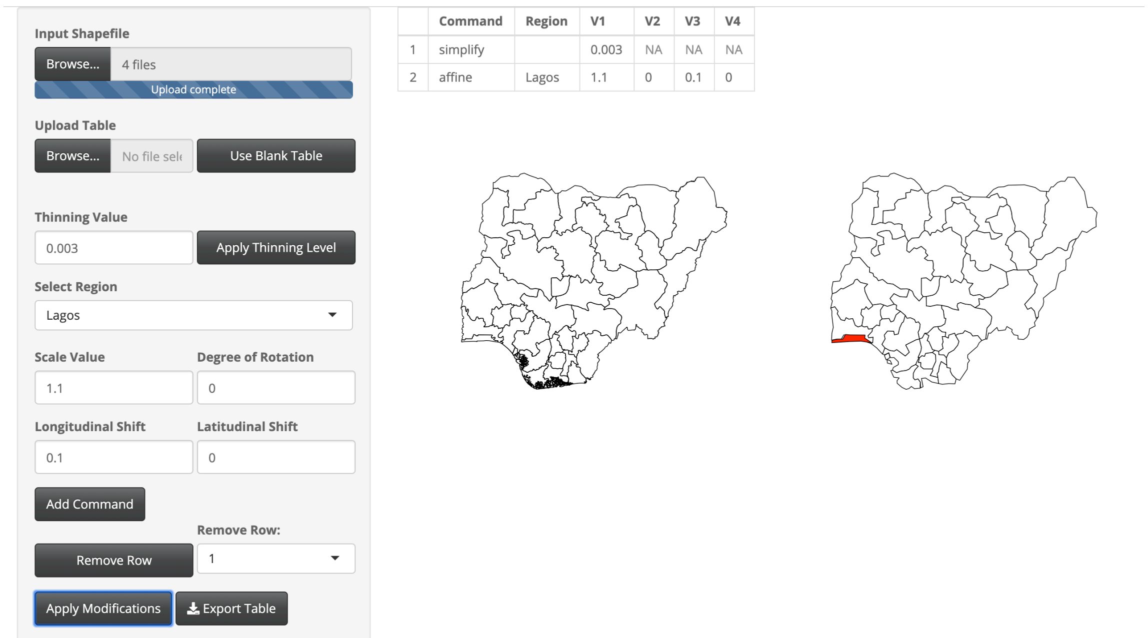


Fig. A.30: Modifications done to Nigeria.

Thinning was done to keep 0.3% of the original boundary points. The Laos region was scaled larger with a scale factor of 1.1 and shifted east 0.1 degrees longitude.

Senegal

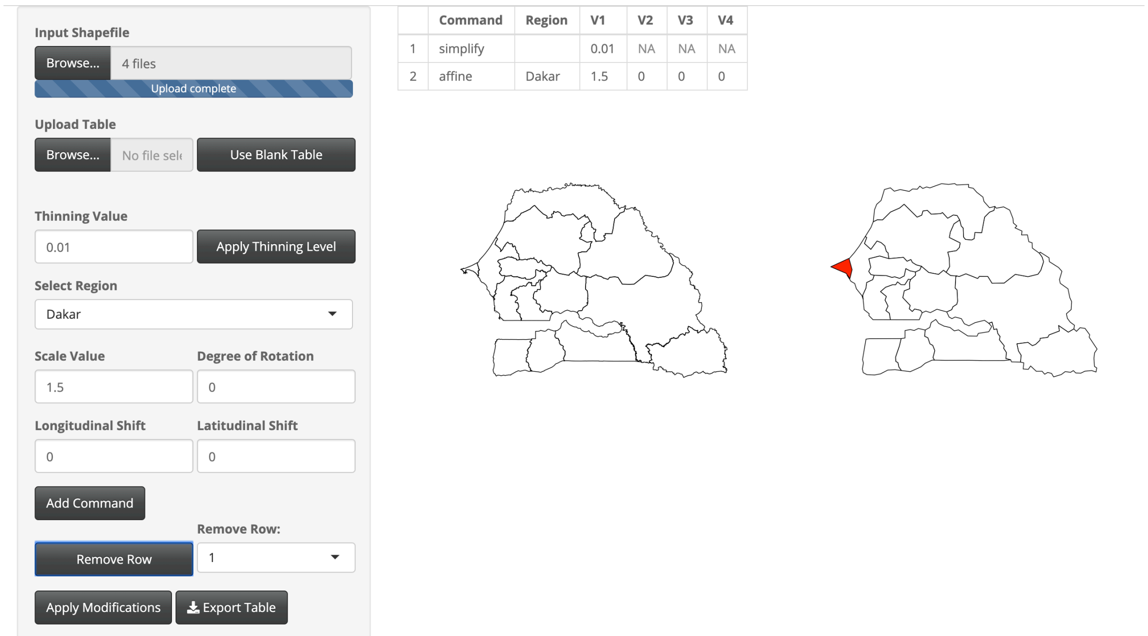


Fig. A.31: Modifications done to Senegal.

Thinning was done to keep 1% of the original boundary points. The Dakar region was scaled larger with a scale factor of 1.5.

Somalia

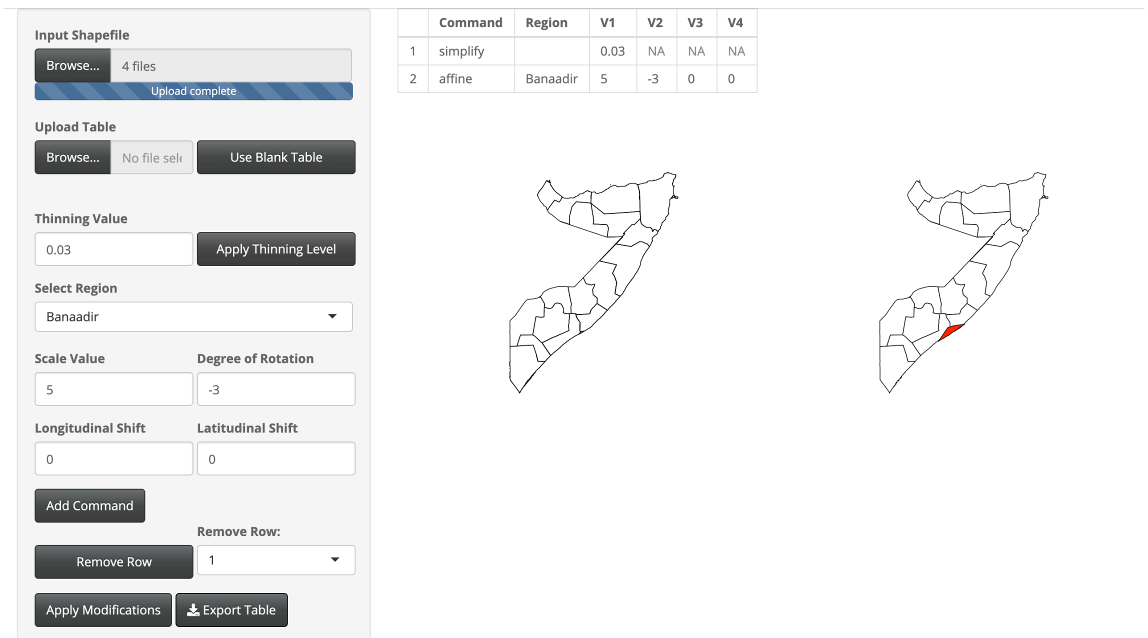


Fig. A.32: Modifications done to Somalia.

Thinning was done to keep 3% of the original boundary points. The Banaadir region was scaled larger with a scale factor of 5 and rotated 3 degrees counter-clockwise.

South Africa

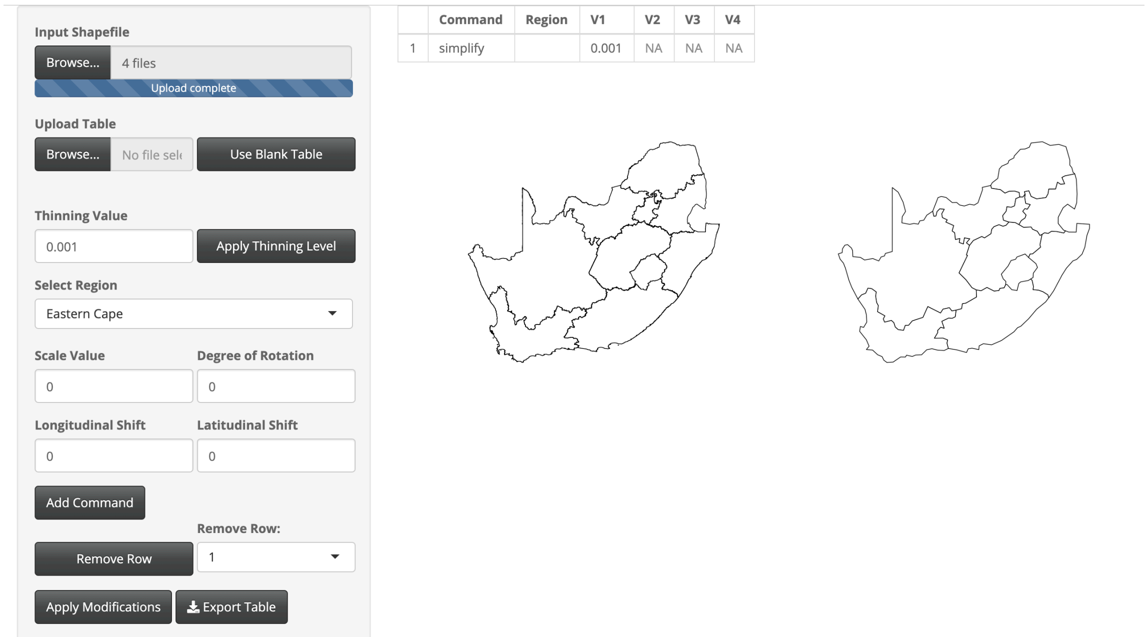


Fig. A.33: Modifications done to South Africa.

Thinning was done to keep 0.1% of the original boundary points.

South Sudan

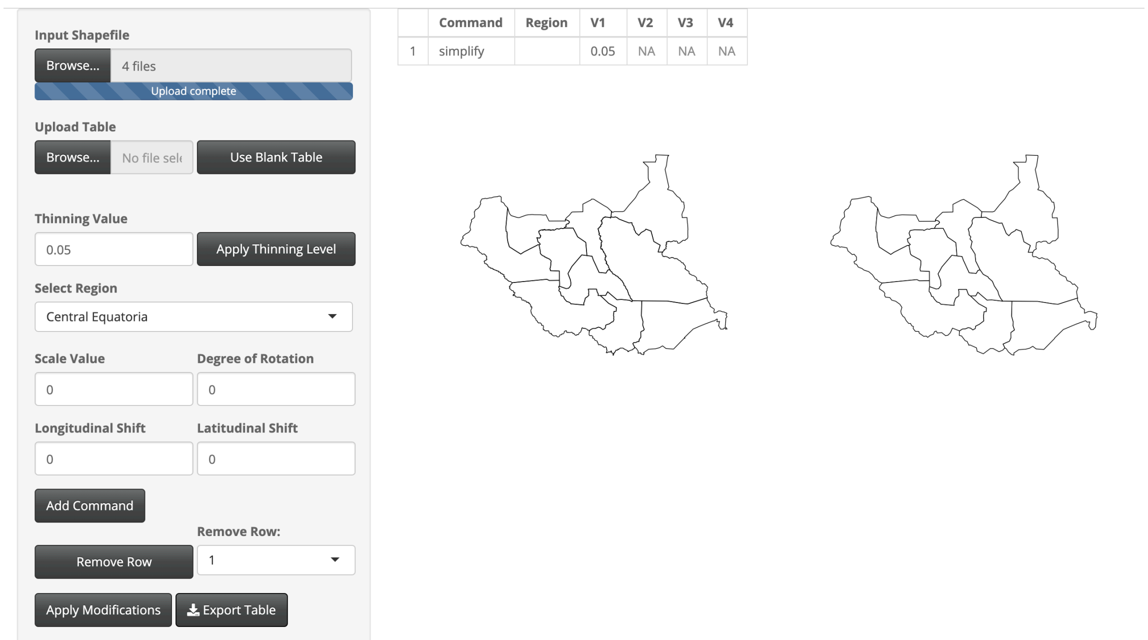


Fig. A.34: Modifications done to South Sudan.

Thinning was done to keep 5% of the original boundary points.

Sudan

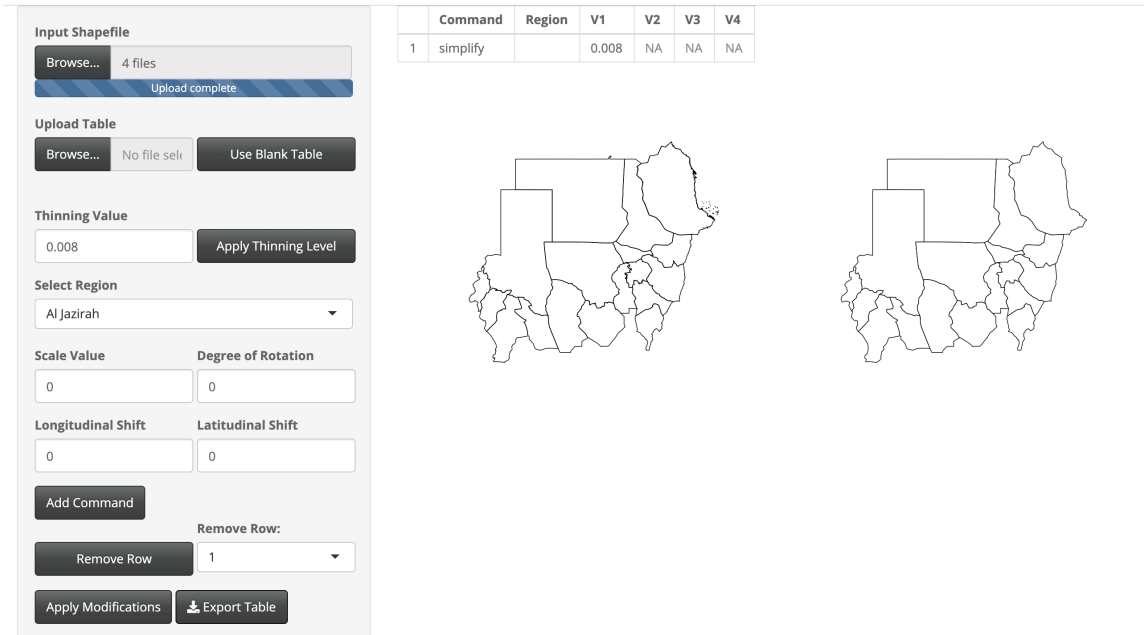


Fig. A.35: Modifications done to Sudan.

Thinning was done to keep 0.8% of the original boundary points.

Tanzania

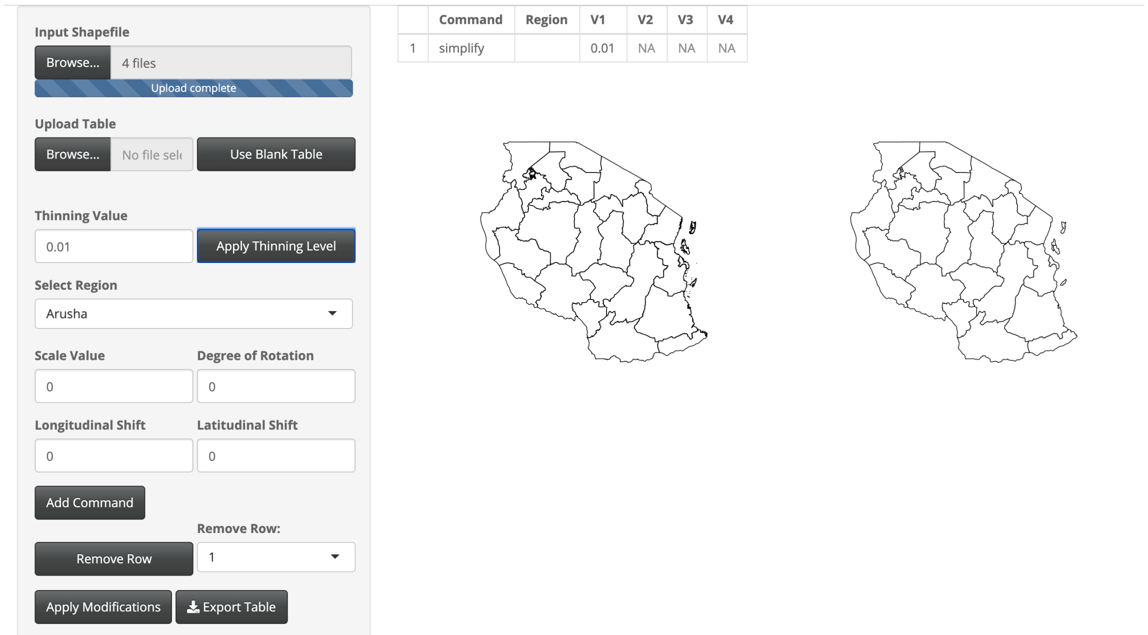


Fig. A.36: Modifications done to Tanzania.

Thinning was done to keep 1% of the original boundary points.

Tunisia

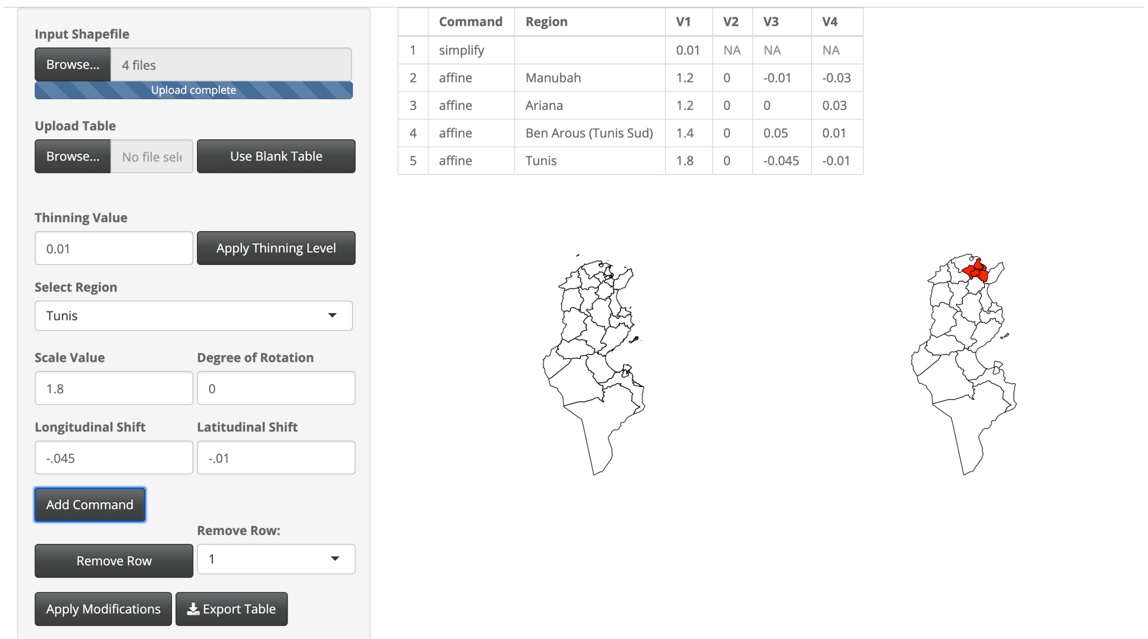


Fig. A.37: Modifications done to Tunisia.

Thinning was done to keep 1% of the original boundary points. The Manubah region was scaled larger with a scale factor of 1.2, shifted west 0.01 degrees longitude, and shifted south 0.03 degrees latitude. The Ariana region was scaled larger with a scale factor of 1.2 and shifted north 0.03 degrees latitude. The Ben Arous (Tunis Sud) region was scaled larger with a scale factor of 1.4, shifted east 0.05 degrees longitude, and north 0.01 degrees latitude. The Tunis region was scaled larger with a scale factor of 1.8, shifted west 0.045 degrees longitude, and south 0.01 degrees latitude.

Zambia

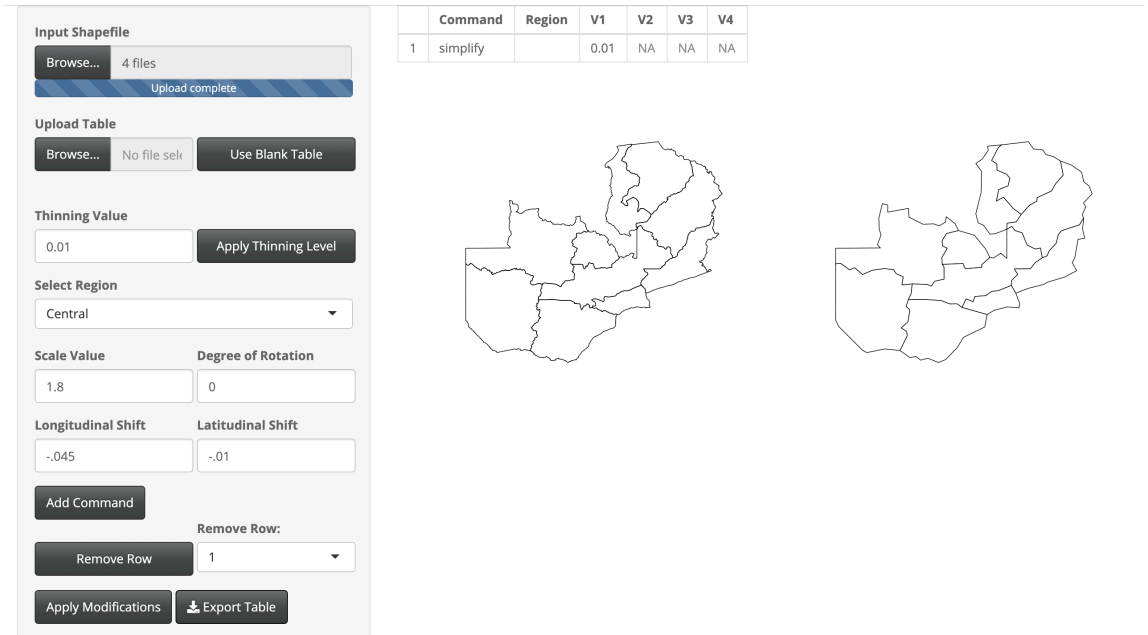


Fig. A.38: Modifications done to Zambia.

Thinning was done to keep 1% of the original boundary points.

Zimbabwe

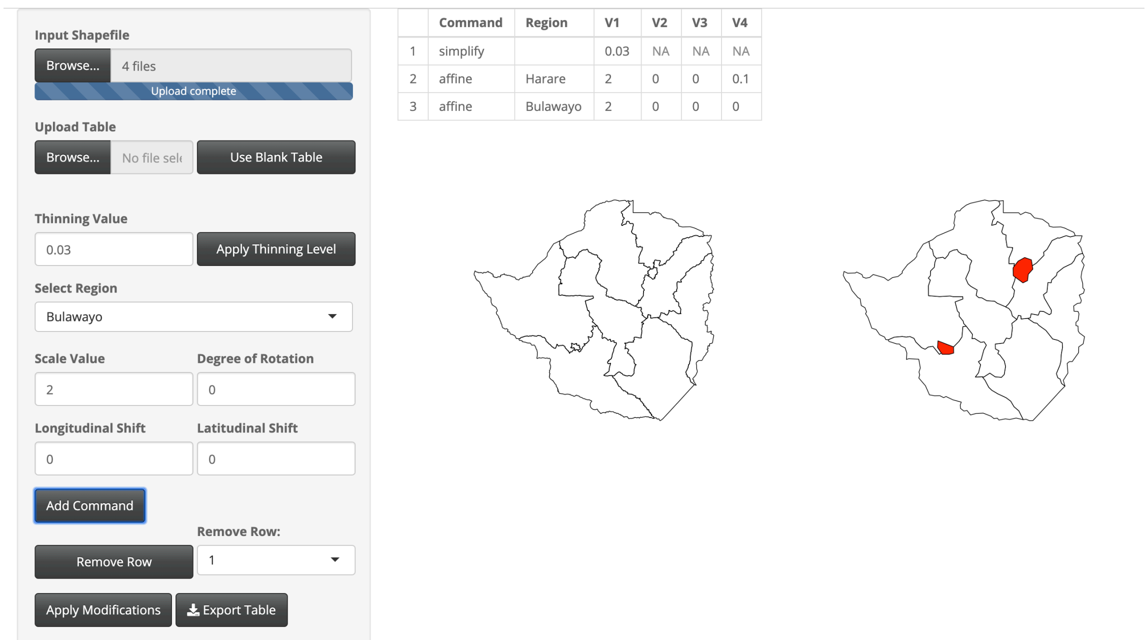


Fig. A.39: Modifications done to Zimbabwe.

Thinning was done to keep 3% of the original boundary points. The Harare region was scaled larger with a scale factor or 2 and shifted north 0.1 degrees latitude. The Bulawayo region was scaled larger with a scale factor of 2.

A.2 Asia

Table A.2: Table listing Asian countries modified and included in the 'LMshapemaker' R package. Countries indicated with a * were created at the command level outside of the 'LMshapemaker' Shiny app.

Country	**Page Number**
Afghanistan	103
Armenia	104
Azerbaijan	105
Bangladesh	106
Bhutan	107
Cambodia	108
China*	109
Georgia	111
Iran	112
Iraq	113
Israel	114
Japan*	115
Jordan	117
Kazakhstan	118
Kuwait	119
Laos	120
Lebanon	121
Malaysia	122
Mongolia	123
Myanmar	124
Oman	125
Pakistan	126

Continued on next page

Table A.2 – continued from previous page

Country	Page Number
Qatar	127
Saudi Arabia	128
South Korea	129
Sri Lanka	130
Syria	131
Timor-Leste	132
Turkey	133
Turkmenistan	134
Uzbekistan	135
Vietnam	136
Yemen	137

The following countries have been excluded from the 'LMshapemaker' R package for the reasons indicated.

- Indonesia, Philippines, and Thailand were excluded for consisting of complicated archipelagic geography.
- Bahrain, Brunei, Cyprus, Kyrgyzstan, Maldives, Nepal, the Palestinian Territories, Singapore, Tajikistan, and Taiwan were excluded for containing five or less first level administrative districts.
- The United Arab Emirates and North Korea for political and geographical disputes over boundaries.
- India was excluded for consisting of complicated first level administrative divisions in the GADM shapefile.

Afghanistan

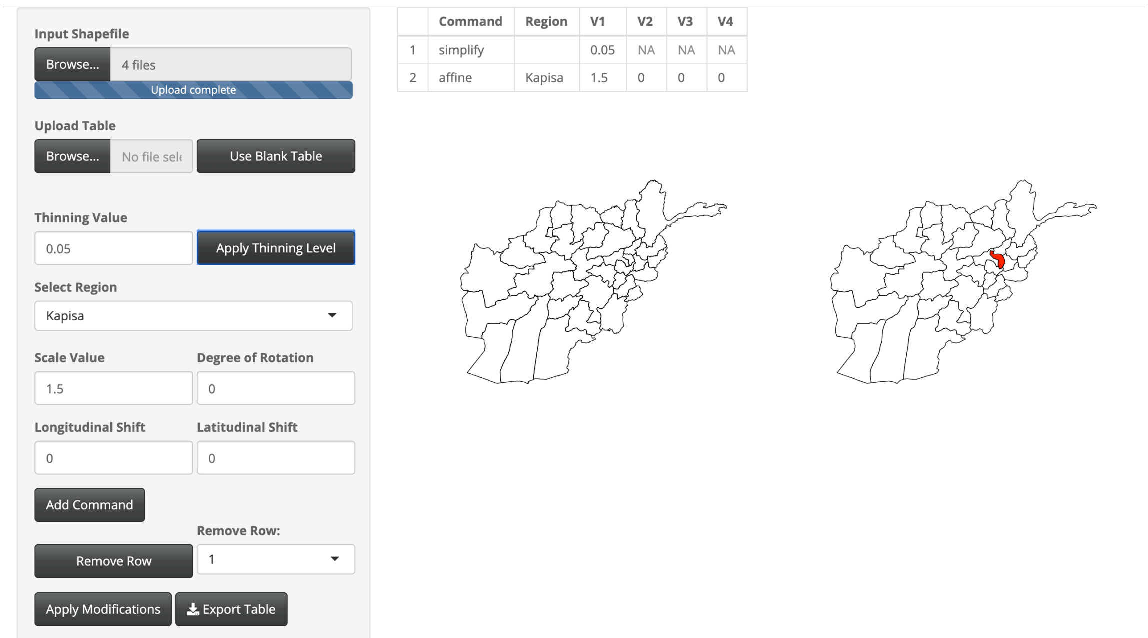


Fig. A.40: Modifications done to Afghanistan.

Thinning was done to keep 5% of the original boundary points. The Kapisa region was scaled larger with a scale factor of 1.5.

Armenia

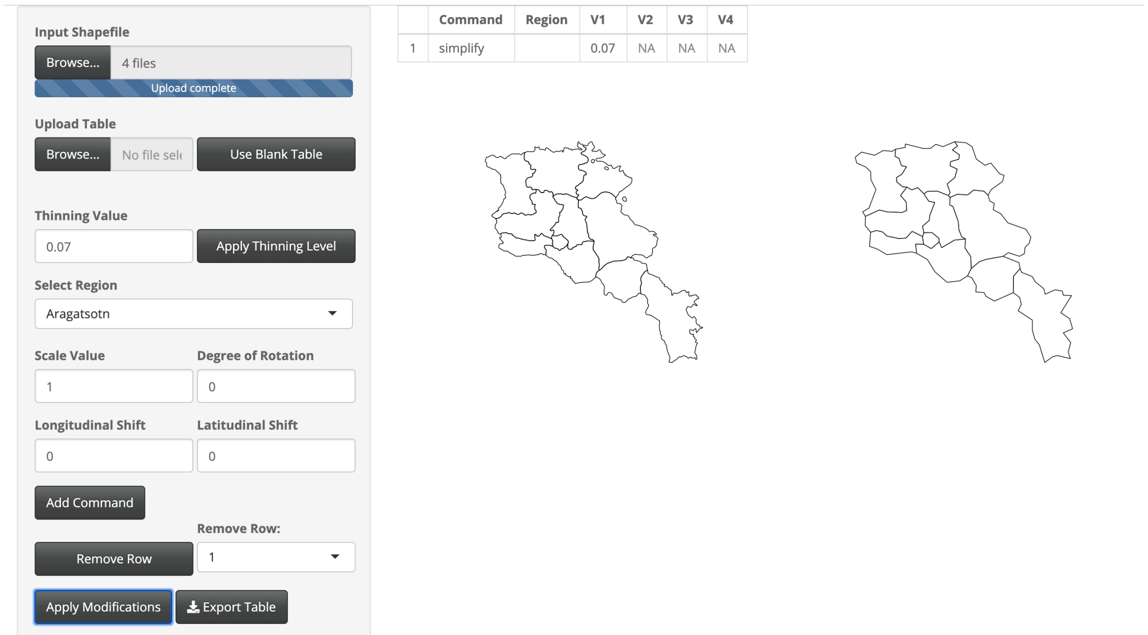


Fig. A.41: Modifications done to Armenia.

Thinning was done to keep 7% of the original boundary points.

Azerbaijan

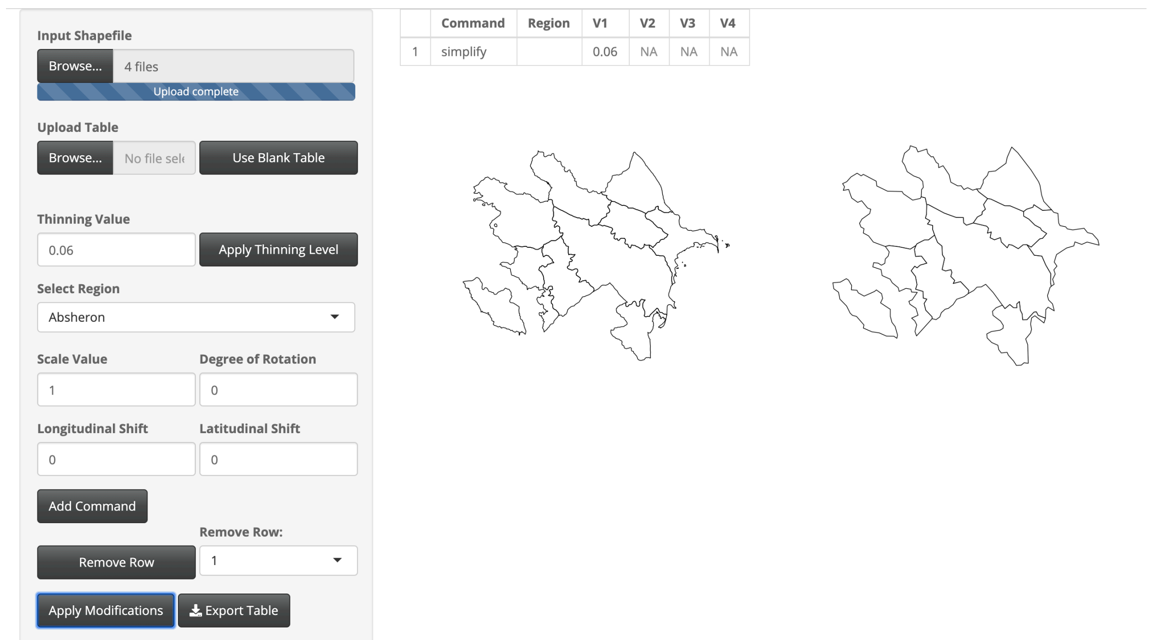


Fig. A.42: Modifications done to Azerbaijan.

Thinning was done to keep 6% of the original boundary points.

Bangladesh

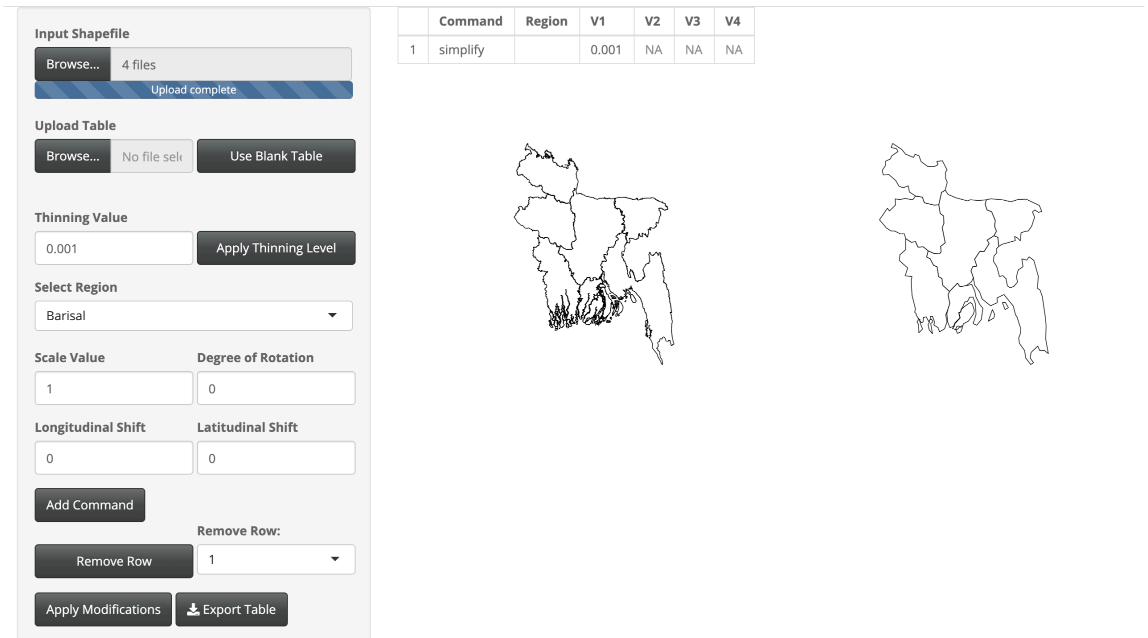


Fig. A.43: Modifications done to Bangladesh.

Thinning was done to keep 0.1% of the original boundary points.

Bhutan

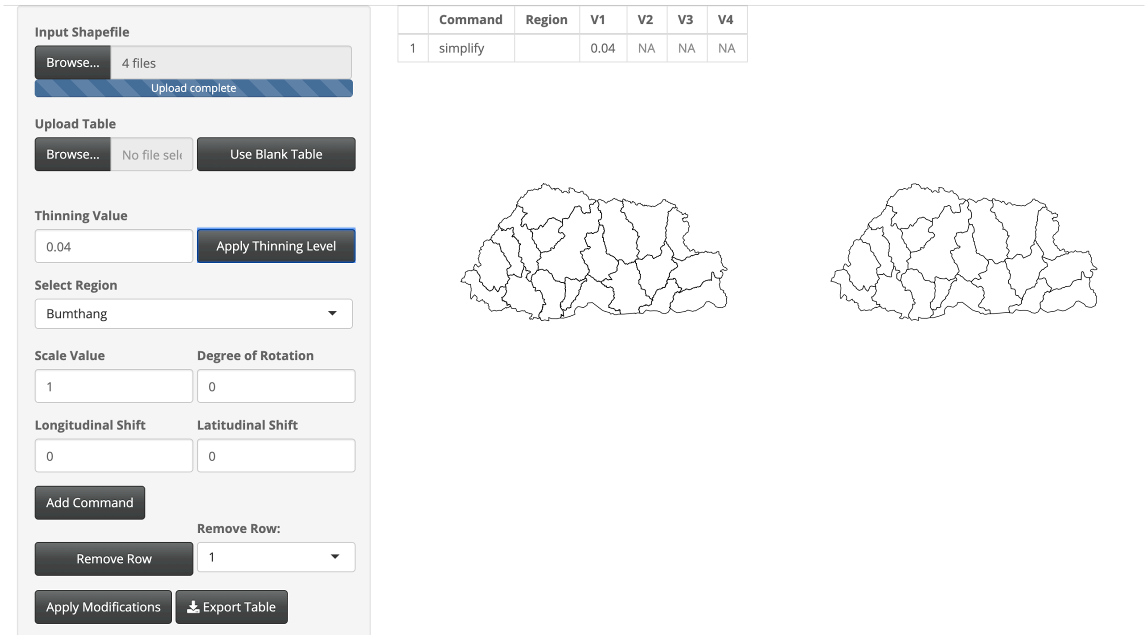


Fig. A.44: Modifications done to Bhutan.

Thinning was done to keep 4% of the original boundary points.

Cambodia

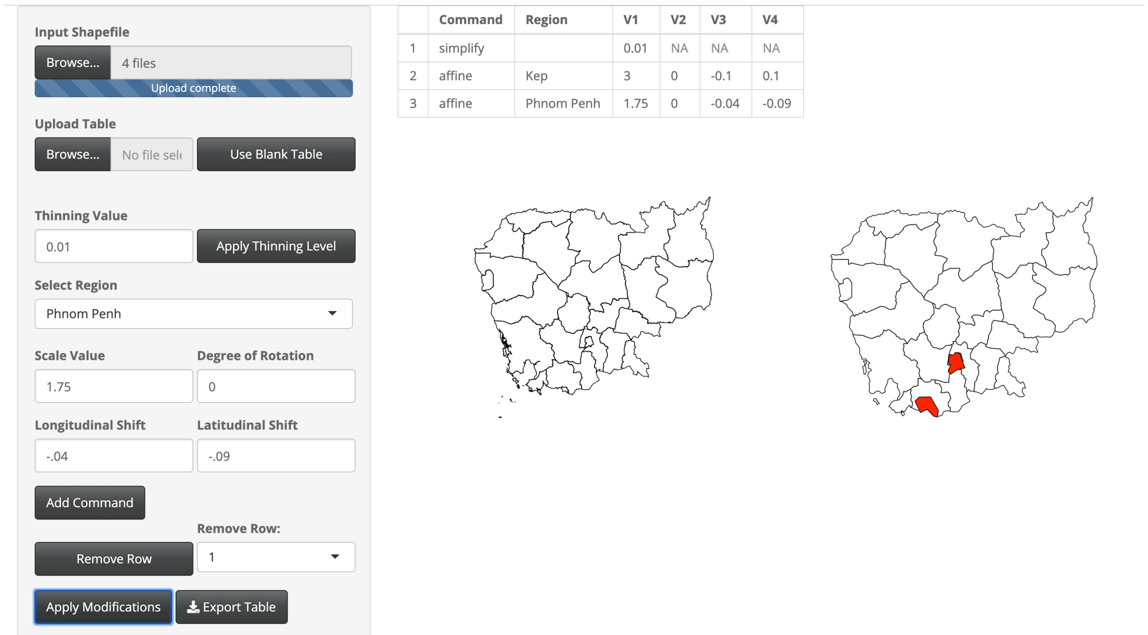
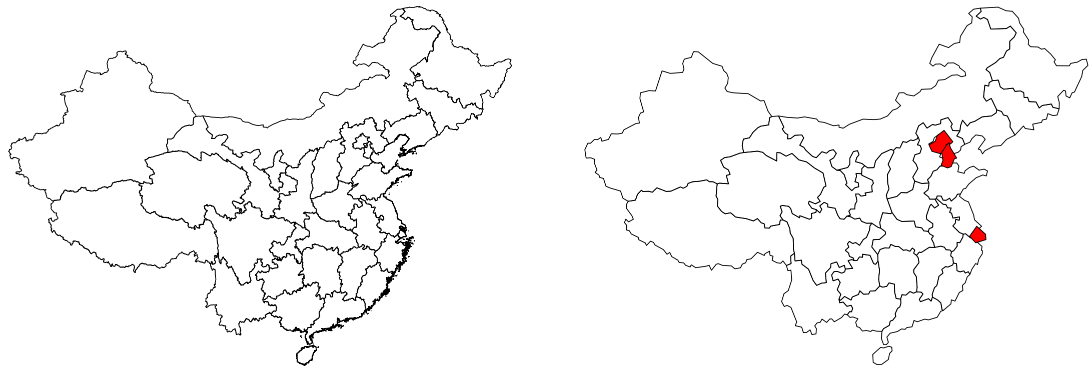


Fig. A.45: Modifications done to Cambodia.

Thinning was done to keep 1% of the original boundary points. The Kep region was scaled larger with a scale factor of 3, shifted west 0.1 degrees longitude, and shifted north 0.1 degrees latitude. The Phnom Penh region was scaled larger with a scale factor of 1.75, shifted west 0.4 degrees longitude, and shifted south 0.9 degrees latitude.

China

```
library(rgdal)
library(rgeos)
library(geojsonio)
library(rmapshaper)
library(sp)
library(dplyr)

shape <- readOGR(dsn = "gadm36_CHN_shp", layer = "gadm36_CHN_1", verbose = TRUE)
CHN_json <- geojson_json(shape, geometry = "polygon", group = "group")

# Unmodified Shapefile
CHN_sp <- geojson_sp(CHN_json)

# Modifed Shapefile
CHN_simplified <- CHN_sp %>%
  ms_simplify(keep = .001) %>%
  geojson_json() %>%
  apply_mapshaper_commands("-affine scale=1.5 shift=.5,-0.5 where='NAME_1 == \"
      Tianjin\"'
                           -affine scale=1.5 shift=.5,0 where='NAME_1 == \"Beijing\"
                               '
                           -affine scale=2 shift=0,0 where='NAME_1 == \"Shanghai\"'"
                               ,
                          force_FC = TRUE) %>%
  geojson_sp()
```



(a) (b)

Fig. A.46: China shown (a) with no modification and (b) with simplification to boundaries and modifications to some regions (colored in red).

Thinning was done to keep 0.1% of the original boundary points. The Tinajin region was scaled larger with a scale factor of 1.5, shifted east 0.5 degrees longitude, and south 0.5 degrees latitude. The Beijing region was scaled larger with a scale factor of 1.5 and shifted east 0.5 degrees longitude. The Shanghai region was scaled larger with a scale factor of 2.

Georgia

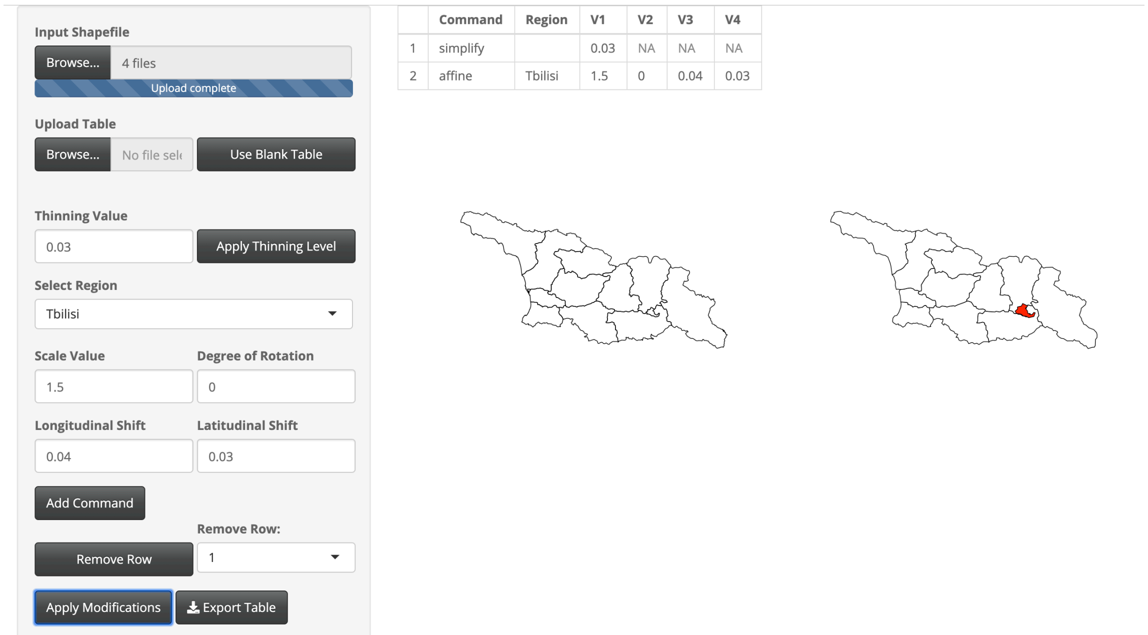


Fig. A.47: Modifications done to Georgia.

Thinning was done to keep 3% of the original boundary points. The Tbilisi region was scaled larger with a scale factor of 1.5, shifted east 0.04 degrees longitude, and north 0.03 degrees latitude.

Iran

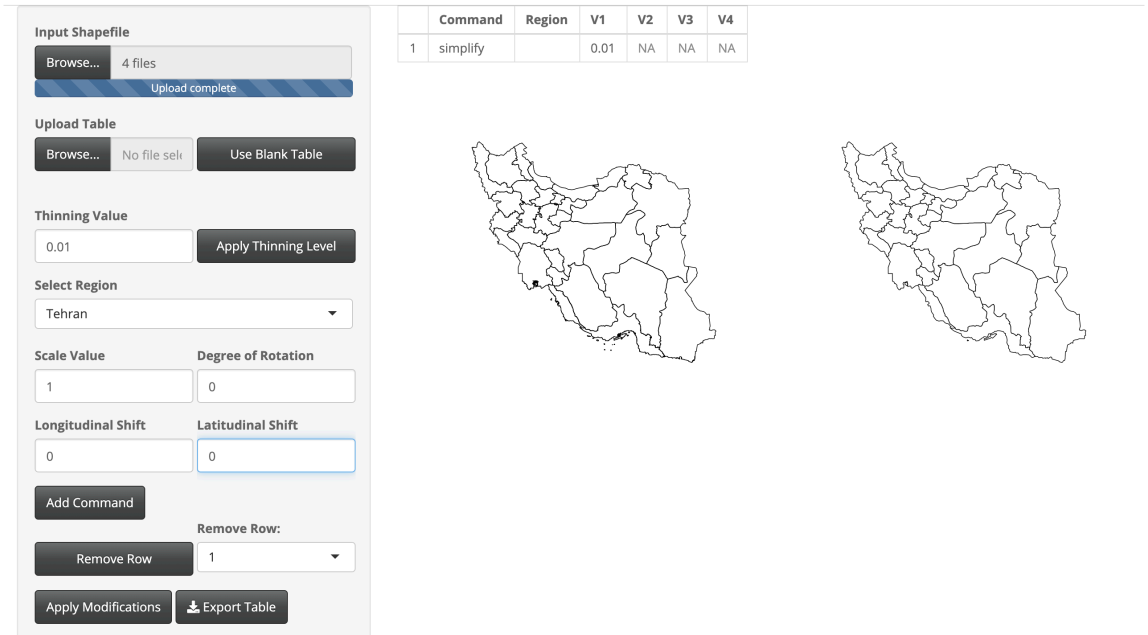


Fig. A.48: Modifications done to Iran.

Thinning was done to keep 1% of the original boundary points.

Iraq

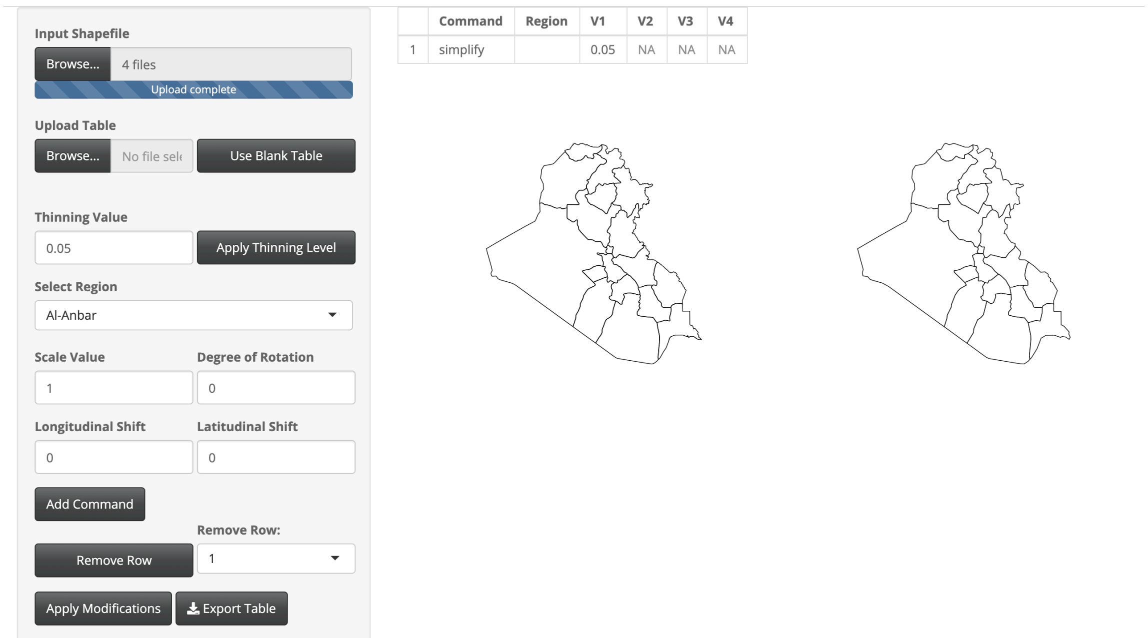


Fig. A.49: Modifications done to Iraq.

Thinning was done to keep 5% of the original boundary points.

Israel

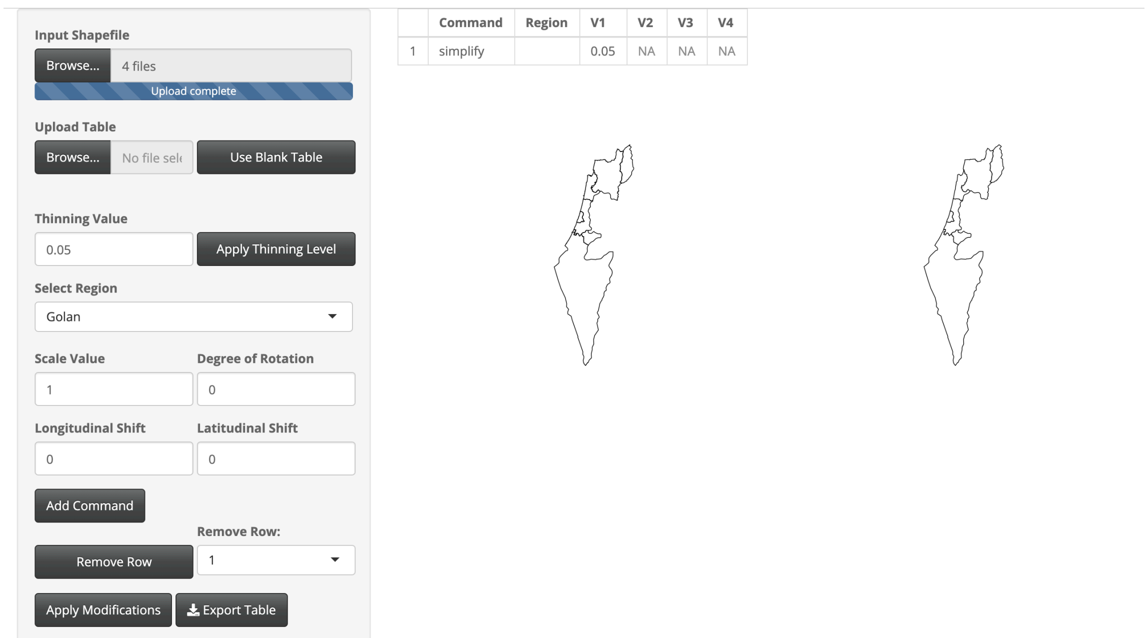
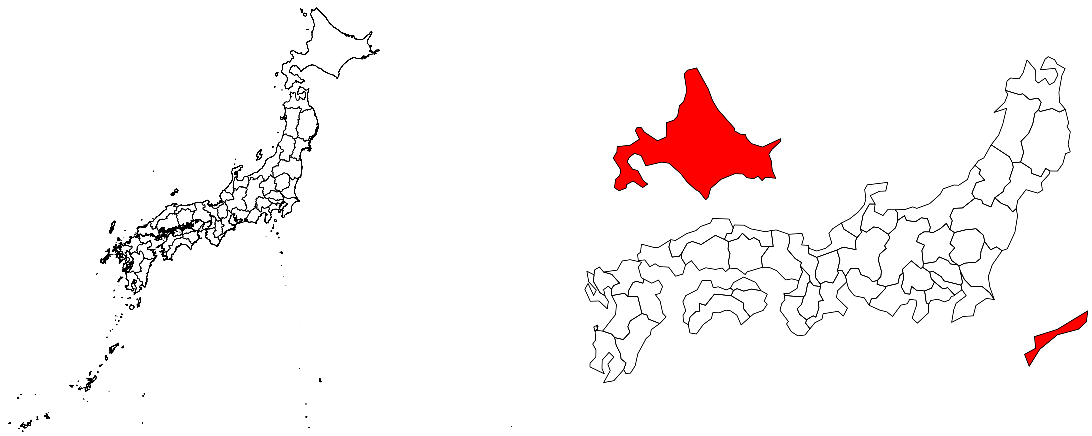


Fig. A.50: Modifications done to Israel.

Thinning was done to keep 5% of the original boundary points.

Japan

```
library(rgdal)
library(rgeos)
library(geojsonio)
library(rmapshaper)
library(sp)
library(dplyr)

shape <- readOGR(dsn = "gadm36_JPN_shp", layer = "gadm36_JPN_1", verbose = TRUE)
JPN_json <- geojson_json(shape, geometry = "polygon", group = "group")

# Unmodified Shapefile
JPN_sp <- geojson_sp(JPN_json)

# Modifed Shapefile
JPN_simplified <- JPN_sp %>%
  ms_simplify(keep = .001) %>%
  geojson_json() %>%
  apply_mapshaper_commands("-affine scale=1 rotate=15 where='NAME_0 == \"Japan\"'
                            -affine scale=.8 shift=-12,-3.5 where='NAME_1 == \"
                                Hokkaido\"'
                            -affine scale=2.2 shift=14,4.25 where='NAME_1 == \"
                                Okinawa\"'",
                           force_FC = TRUE) %>%
  geojson_sp()

JPN_simplified <- ms_filter_islands(JPN_simplified, min_area = 1000000000)
```



(a) (b)

Fig. A.51: Japan shown (a) with no modification and (b) with simplification to boundaries and modifications to some regions (colored in red).

Thinning was done to keep 0.1% of the original boundary points. The whole of mainland Japan (‘NAME_0 == "Japan") was rotated 15 degrees clockwise. The Hokkaido region was scaled smaller with a scale factor of 0.8, shifted west 12 degrees longitude, and south 3.5 degrees latitude. The Okinawa region was scaled larger with a scale factor of 2.2, shifted east 14 degrees longitude, and north 4.25 degrees latitude.

Jordan

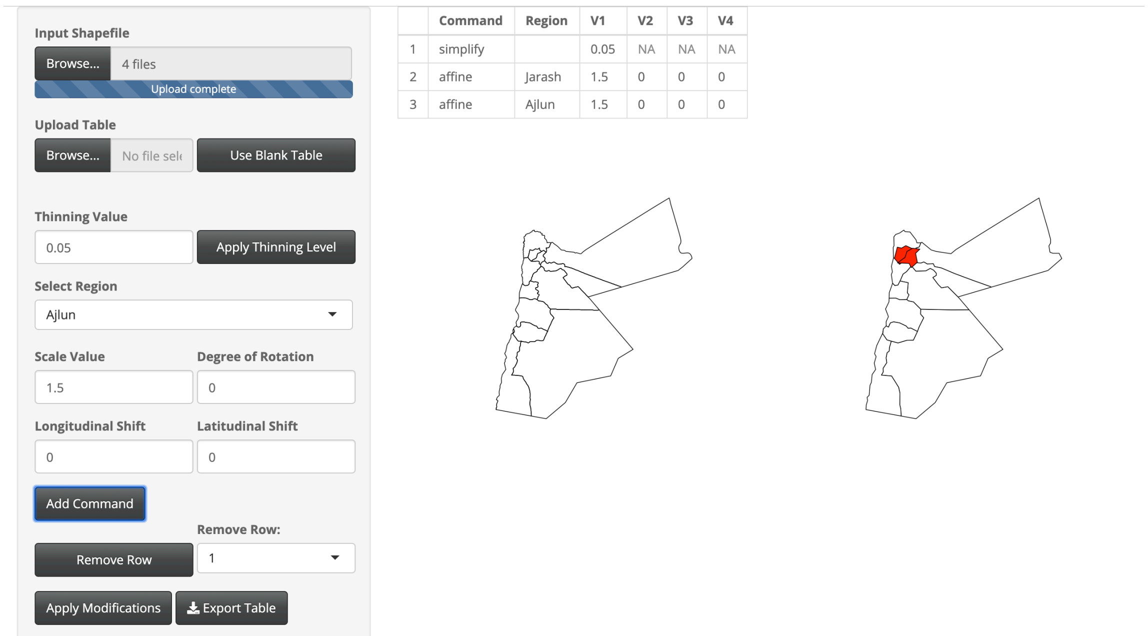


Fig. A.52: Modifications done to Jordan.

Thinning was done to keep 5% of the original boundary points. The Jarash region was scaled larger with a scale factor of 1.5. The Ajlun region was scaled larger with a scale factor of 1.5.

Kazakhstan

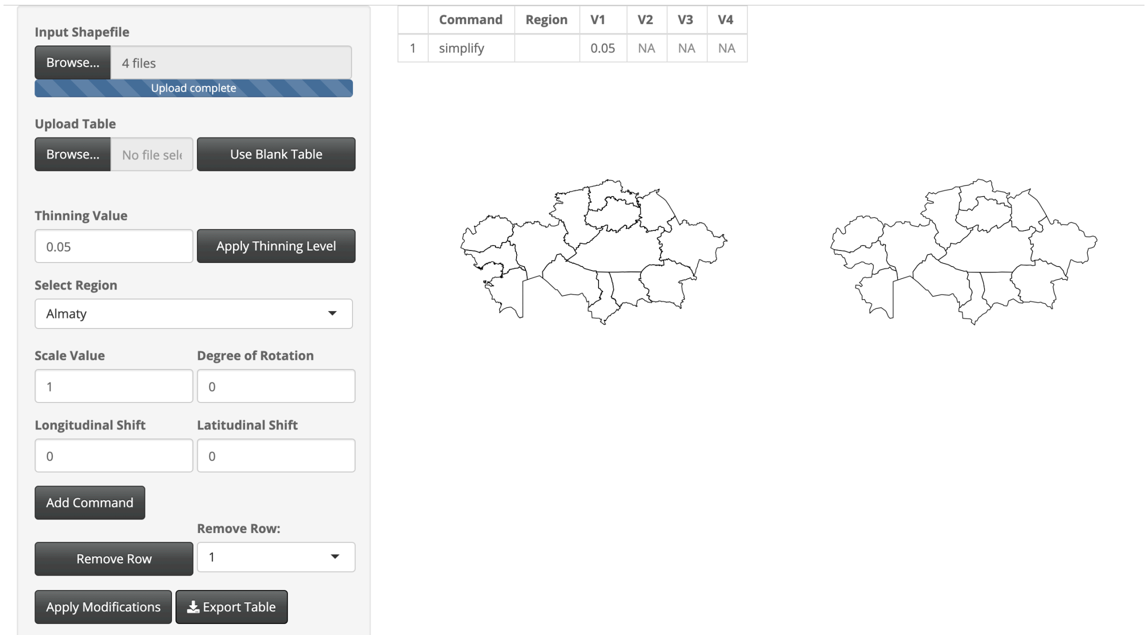


Fig. A.53: Modifications done to Kazakhstan.

Thinning was done to keep 5% of the original boundary points.

Kuwait

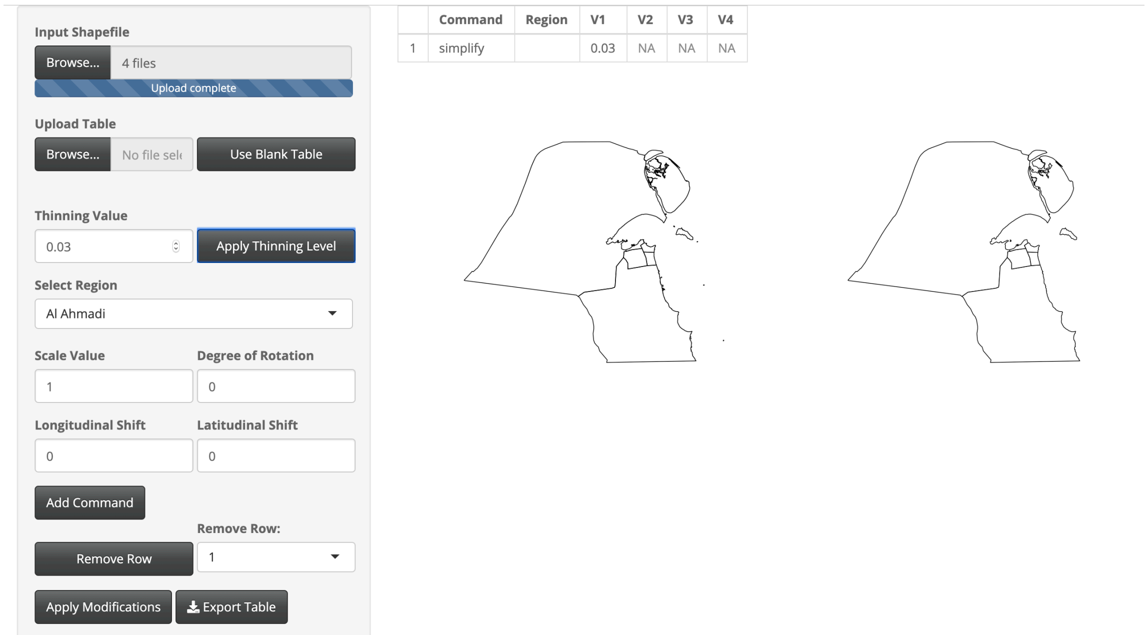


Fig. A.54: Modifications done to Kuwait.

Thinning was done to keep 3% of the original boundary points.

Laos

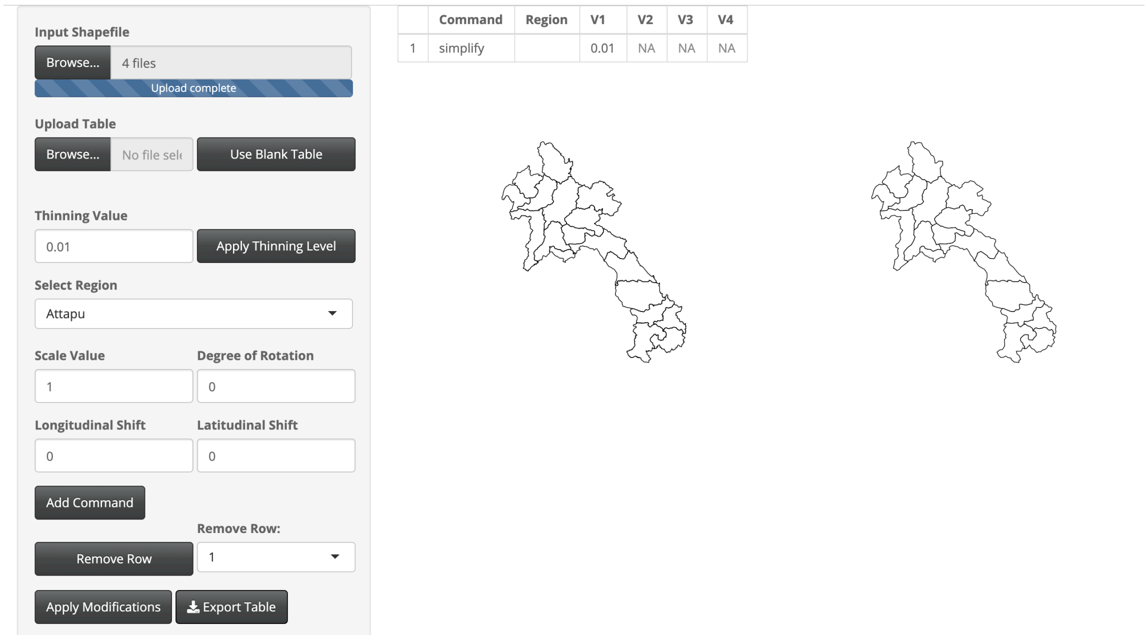


Fig. A.55: Modifications done to Laos.

Thinning was done to keep 1% of the original boundary points.

Lebanon

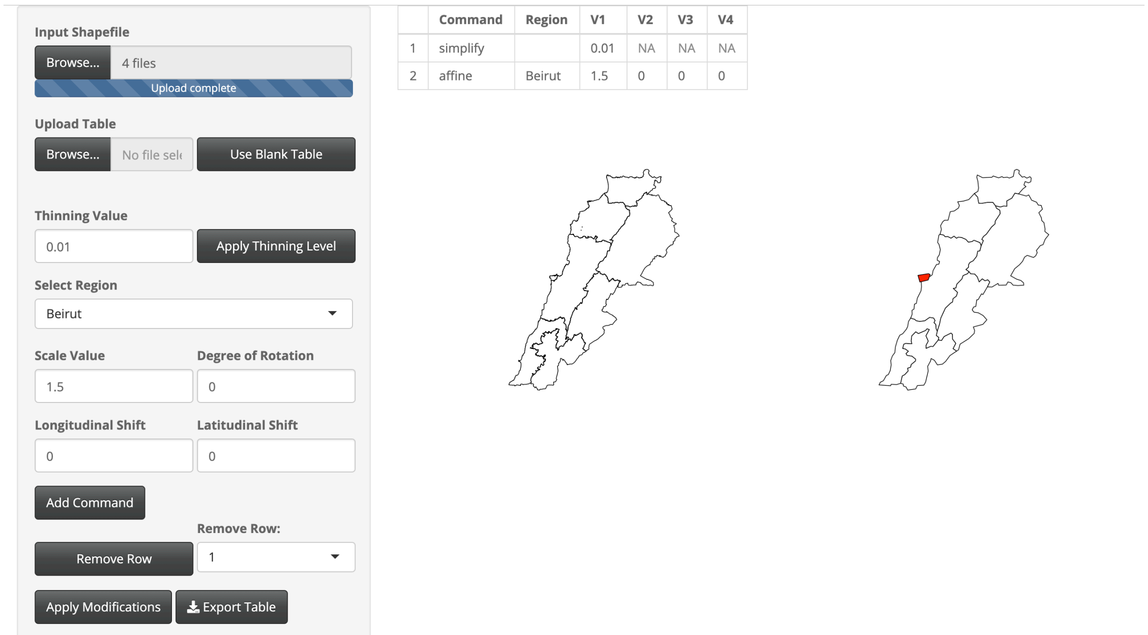
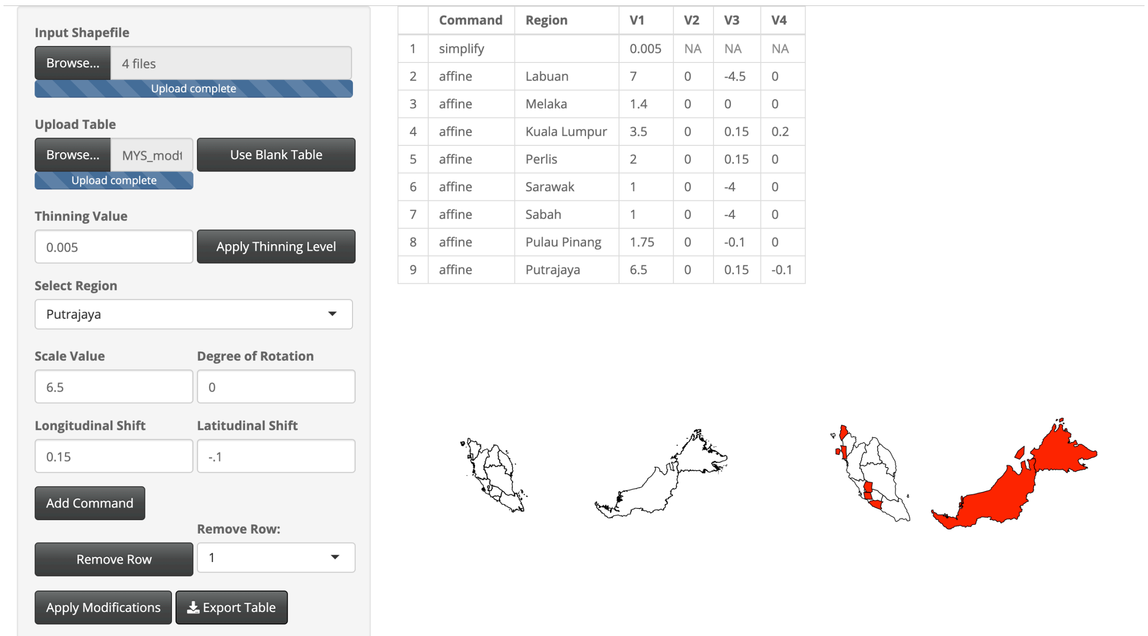


Fig. A.56: Modifications done to Lebanon.

Thinning was done to keep 1% of the original boundary points. The Beirut region was scaled larger with a scale factor of 1.5.

Malaysia



	Command	Region	V1	V2	V3	V4
1	simplify		0.005	NA	NA	NA
2	affine	Labuan	7	0	-4.5	0
3	affine	Melaka	1.4	0	0	0
4	affine	Kuala Lumpur	3.5	0	0.15	0.2
5	affine	Perlis	2	0	0.15	0
6	affine	Sarawak	1	0	-4	0
7	affine	Sabah	1	0	-4	0
8	affine	Pulau Pinang	1.75	0	-0.1	0
9	affine	Putrajaya	6.5	0	0.15	-0.1

Fig. A.57: Modifications done to Malaysia.

Thinning was done to keep 0.5% of the original boundary points. The Labuan region was scaled larger with a scale factor of 7 and shifted west 5.4 degrees longitude. The Melaka region was scaled larger with a scale factor of 1.4. The Kuala Lumpur region was scaled larger with a scale factor of 3.5, shifted east 0.15 degrees longitude, and north 0.2 degrees latitude. The Perlis region was scaled larger with a scale factor of 2 and shifted east 0.15 degrees longitude. The Sarawak region was shifted west 4 degrees longitude. The Sabah region was shifted west 4 degrees longitude. The Pulau Pinang region was scaled larger with a scale factor of 1.75 and shifted west 0.1 degrees longitude. The Putrajaya region was scaled larger with a scale factor of 6.5, shifted east 0.15 degrees longitude, and south 0.1 degrees latitude.

Mongolia

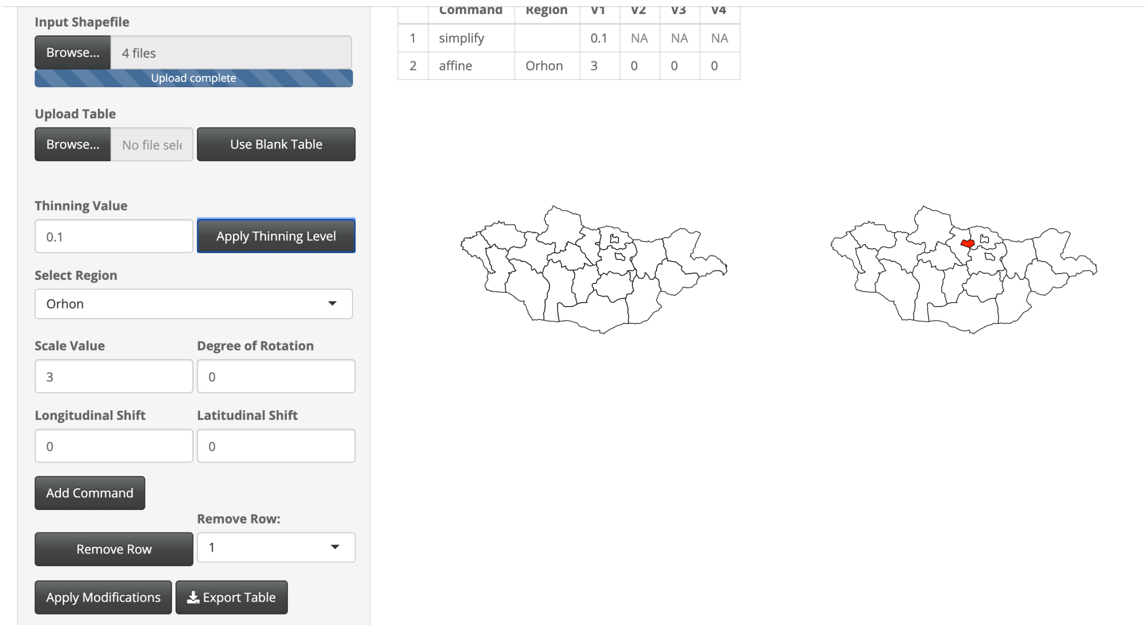


Fig. A.58: Modifications done to Mongolia.

Thinning was done to keep 10% of the original boundary points. The Orhon region was scaled larger with a scale factor of 3.

Myanmar

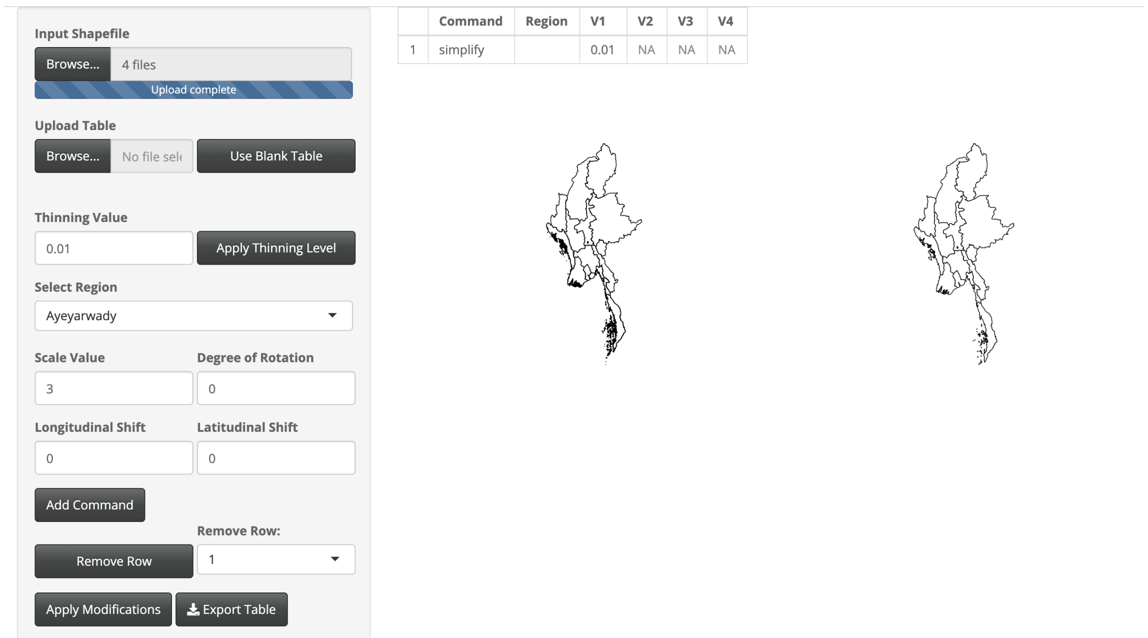


Fig. A.59: Modifications done to Myanmar.

Thinning was done to keep 1% of the original boundary points.

Oman

Fig. A.60: Modifications done to Oman.

Thinning was done to keep 1% of the original boundary points.

Pakistan

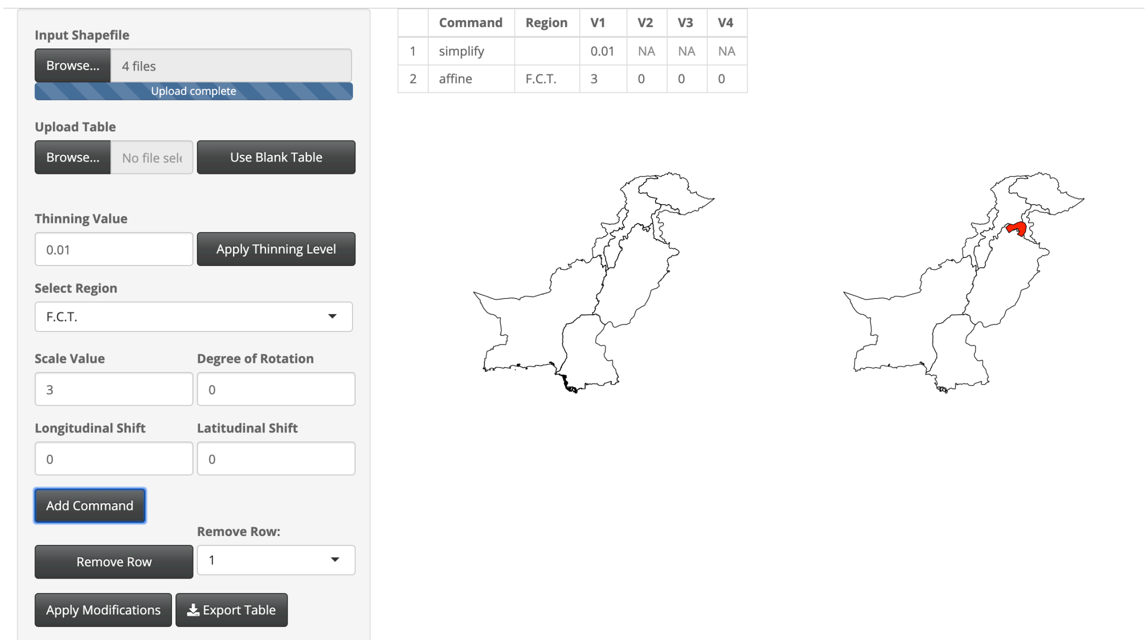


Fig. A.61: Modifications done to Pakistan.

Thinning was done to keep 1% of the original boundary points. The F.C.T. region was scaled larger with a scale factor of 3.

Qatar

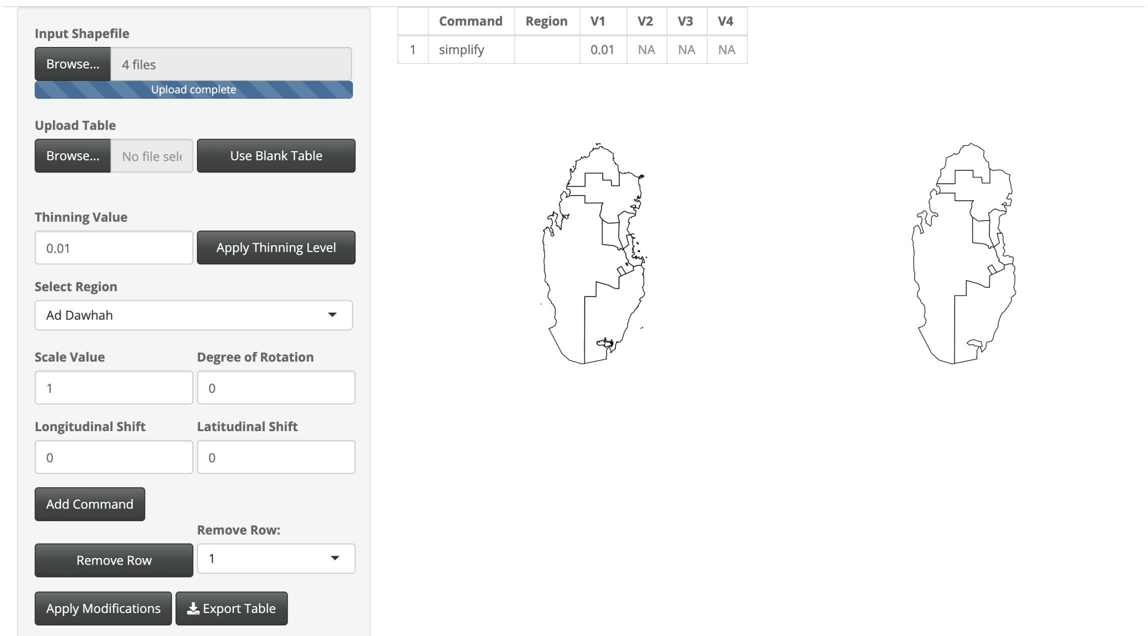


Fig. A.62: Modifications done to Qatar.

Thinning was done to keep 1% of the original boundary points.

Saudi Arabia

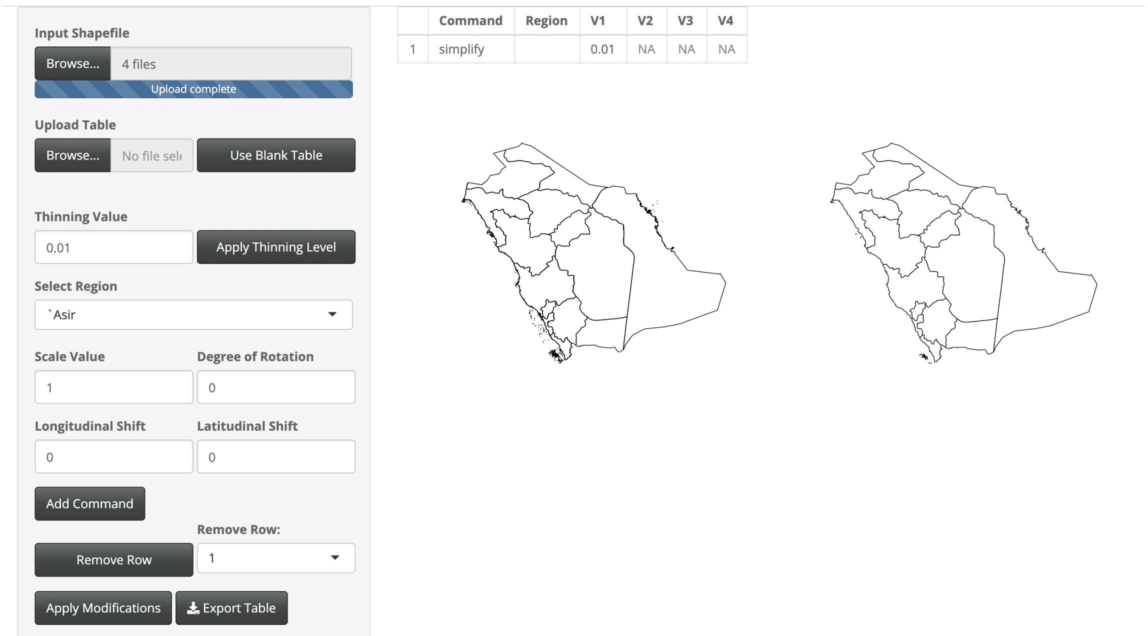


Fig. A.63: Modifications done to Saudia Arabia.

Thinning was done to keep 1% of the original boundary points.

South Korea

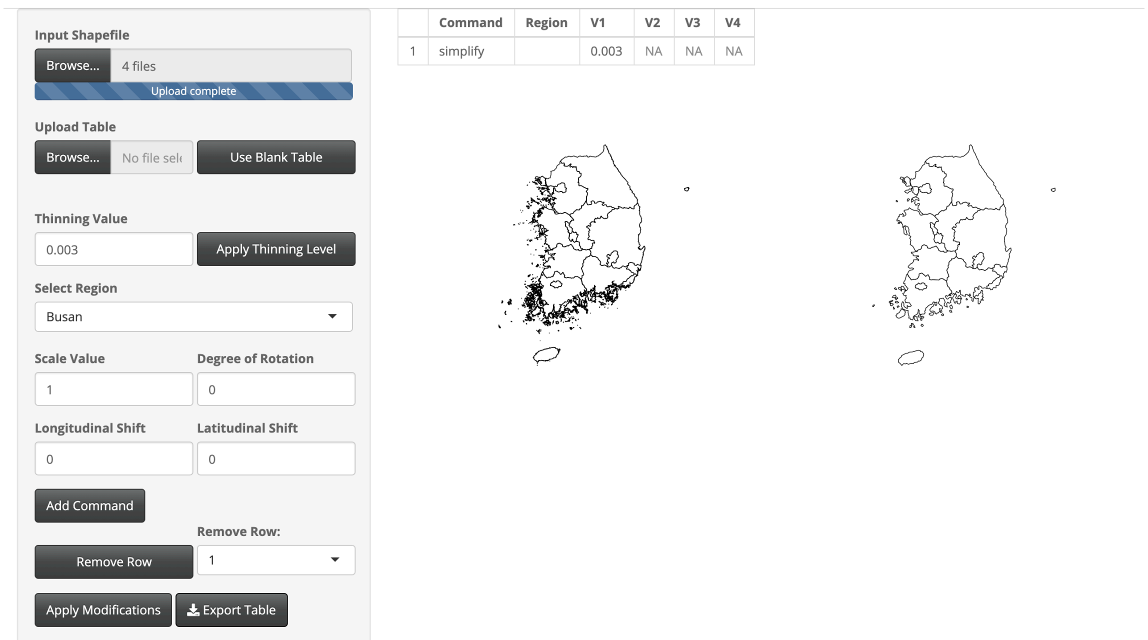


Fig. A.64: Modifications done to South Korea.

Thinning was done to keep 3% of the original boundary points.

Sri Lanka

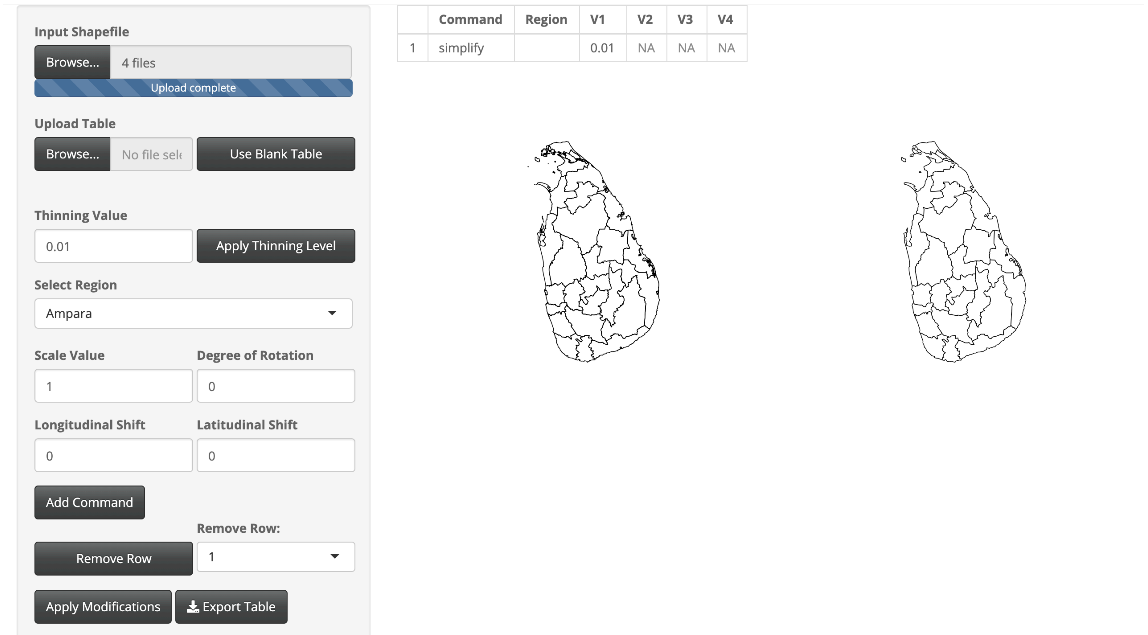


Fig. A.65: Modifications done to Sri Lanka.

Thinning was done to keep 1% of the original boundary points.

Syria

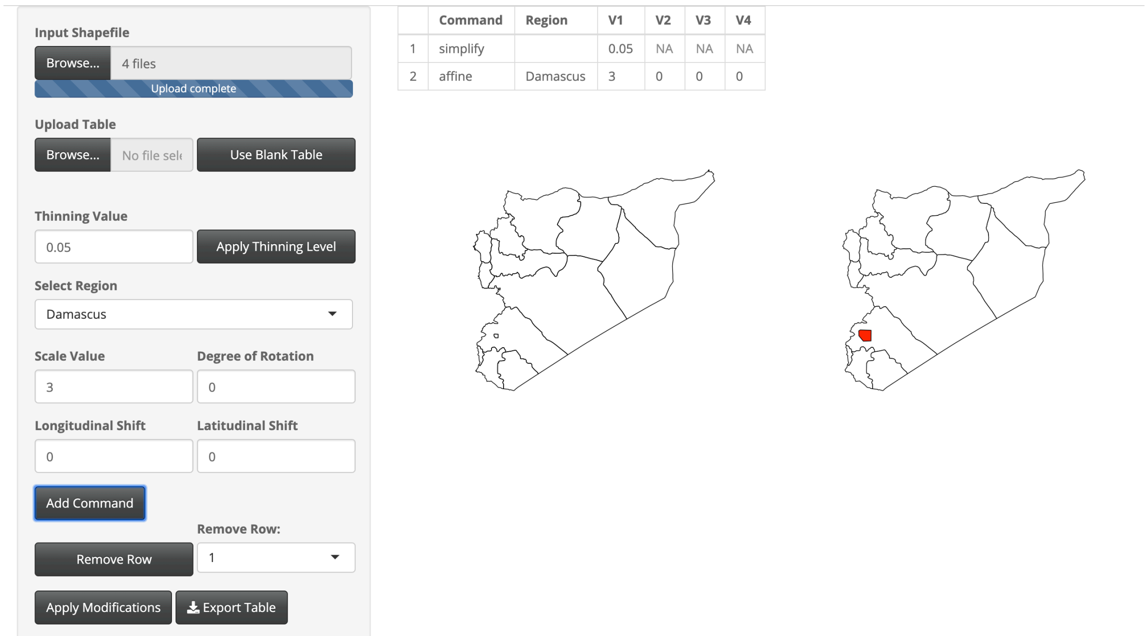


Fig. A.66: Modifications done to Syria.

Thinning was done to keep 5% of the original boundary points. The Damascus region was scaled larger with a scale factor of 3.

Timor-Leste

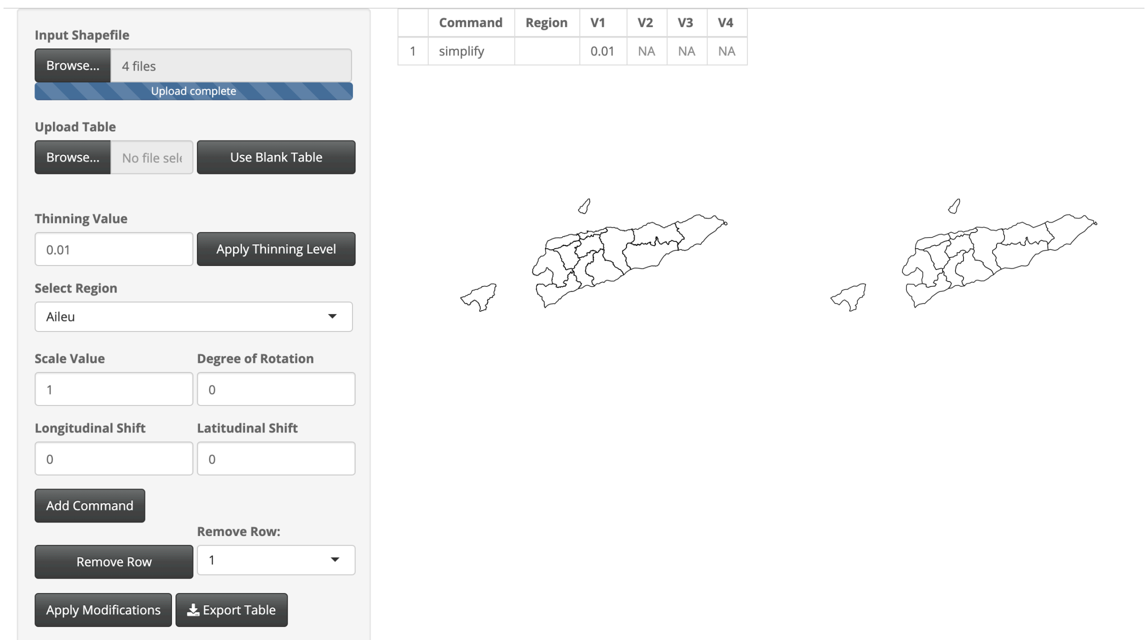


Fig. A.67: Modifications done to Timor-Leste.

Thinning was done to keep 1% of the original boundary points.

Turkey

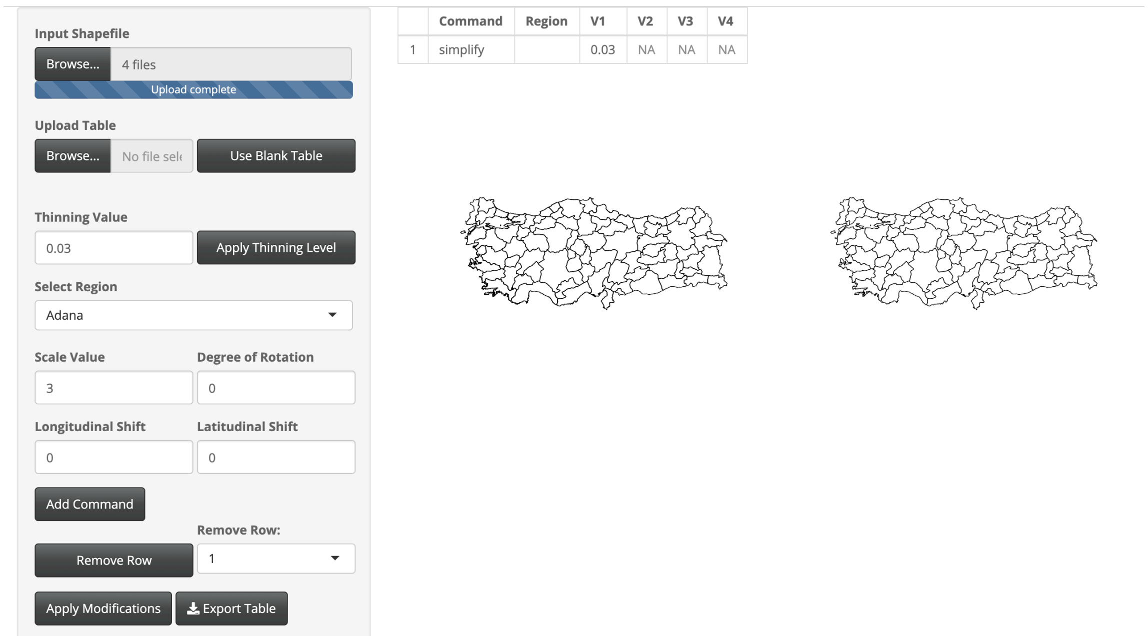


Fig. A.68: Modifications done to Turkey.

Thinning was done to keep 3% of the original boundary points.

Turkmenistan

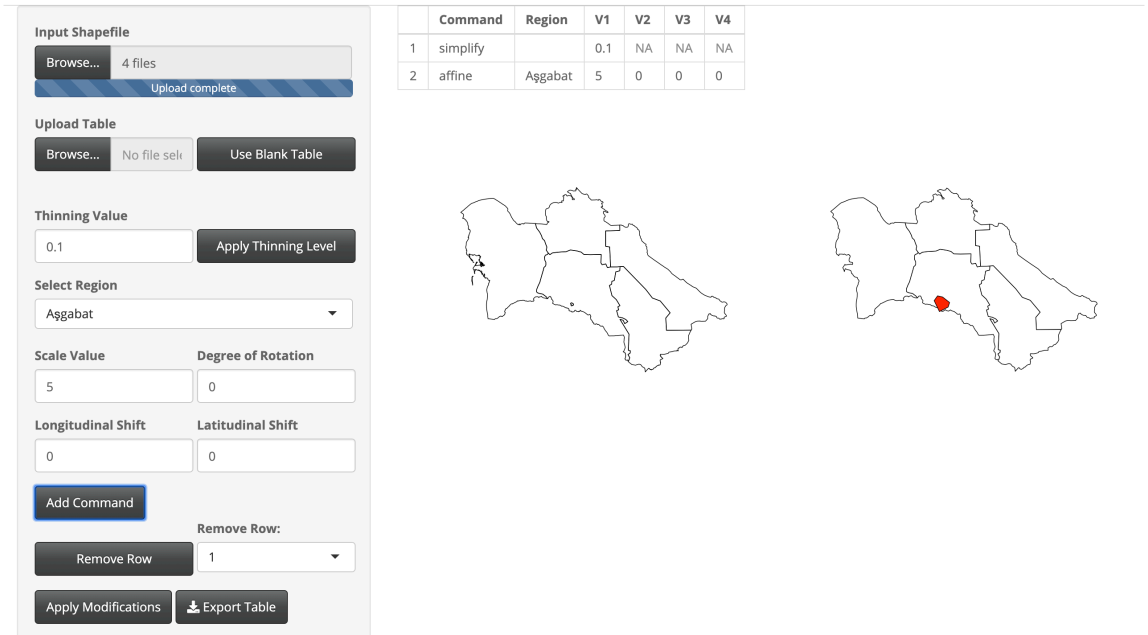


Fig. A.69: Modifications done to Turkmenistan.

Thinning was done to keep 10% of the original boundary points. The Asgabat region was scaled larger with a scale factor of 5.

Uzbekistan

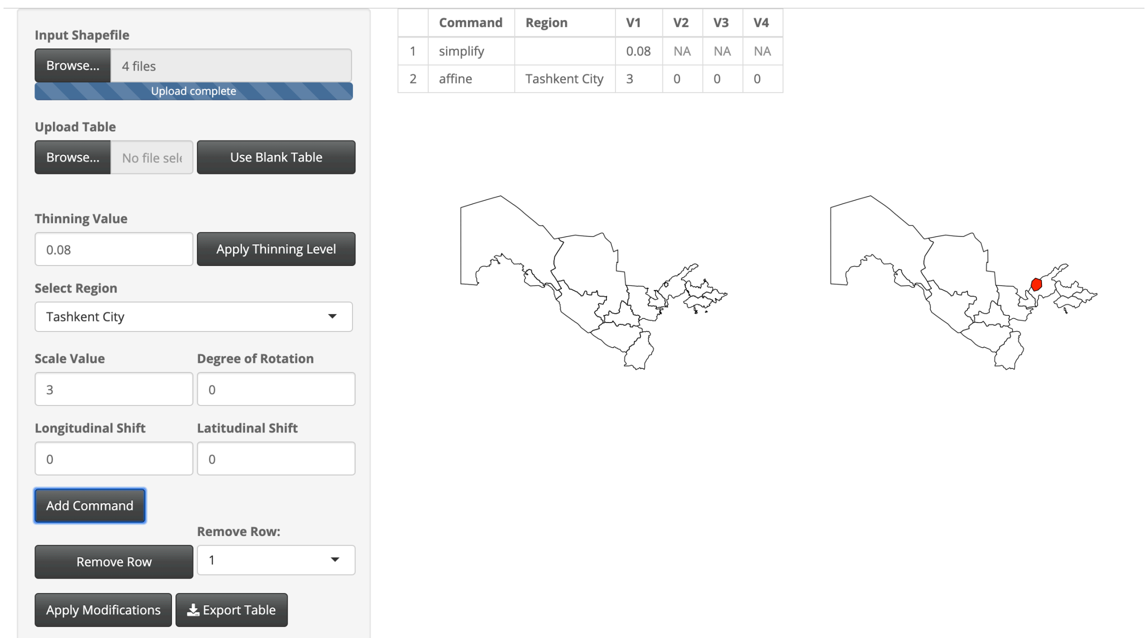


Fig. A.70: Modifications done to Uzbekistan.

Thinning was done to keep 8% of the original boundary points. The Tashkent City region was was scaled larger with a scale factor of 3.

Vietnam

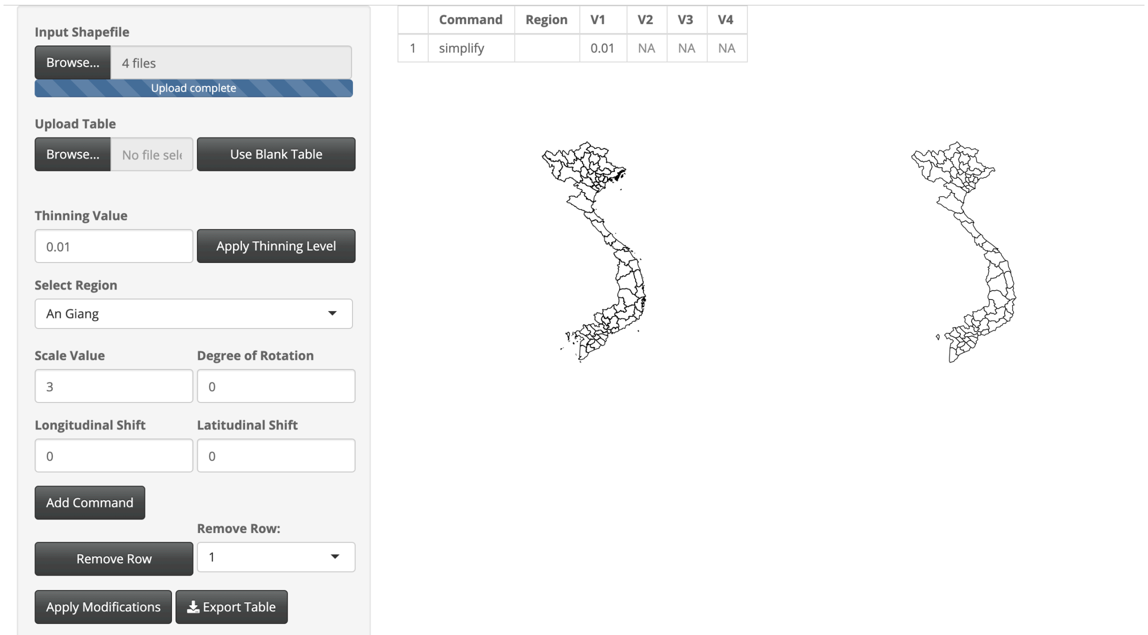


Fig. A.71: Modifications done to Vietnam.

Thinning was done to keep 1% of the original boundary points.

Yemen

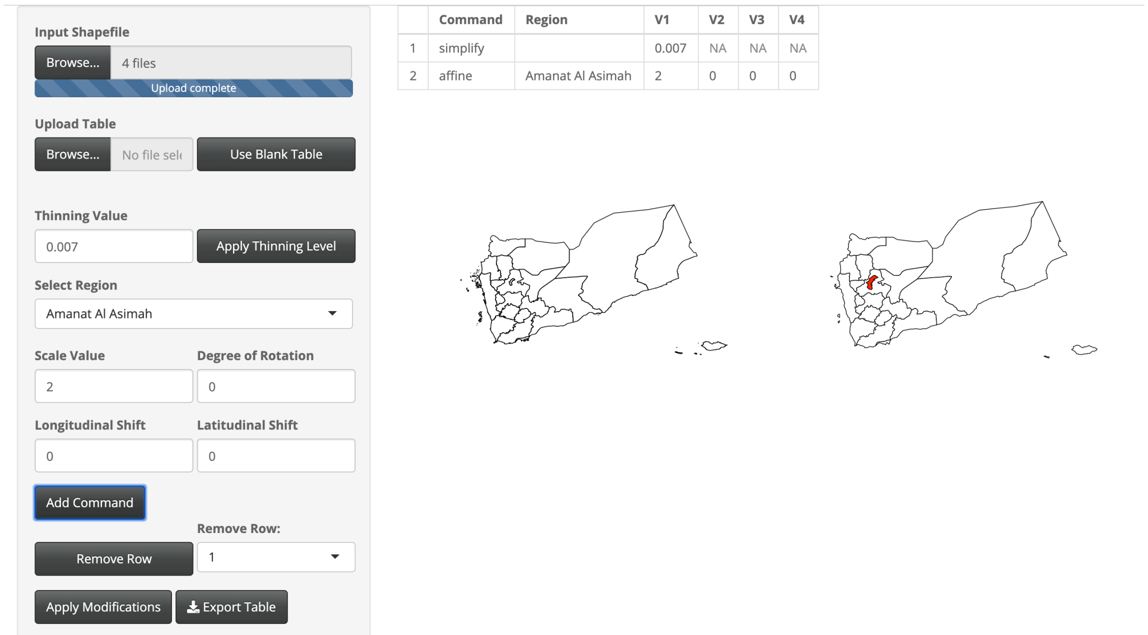


Fig. A.72: Modifications done to Yemen.

Thinning was done to keep 0.7% of the original boundary points. The Amanat Al Asimah region was scaled larger with a scale factor of 2.

A.3 Europe

Table A.3: Table listing European countries modified and included in the 'LMshapemaker' R package.

Continued on next page

Table A.3 – continued from previous page

The following countries have been excluded from the 'LMshapemaker' R package for the reasons indicated.

- Liechtenstein, Portugal, Finland, Macedonia, and Romania were excluded for consisting of complicated geography.
- Belgium, Bosnia and Herzegovina, Denmark, Great Britain, Luxembourg, Latvia, Monaco, Malta, and the Vatican City were excluded for containing five or less first level administrative districts.
- Russia was excluded for consisting of complicated first level administrative divisions in the GADM shapefile.

Albania

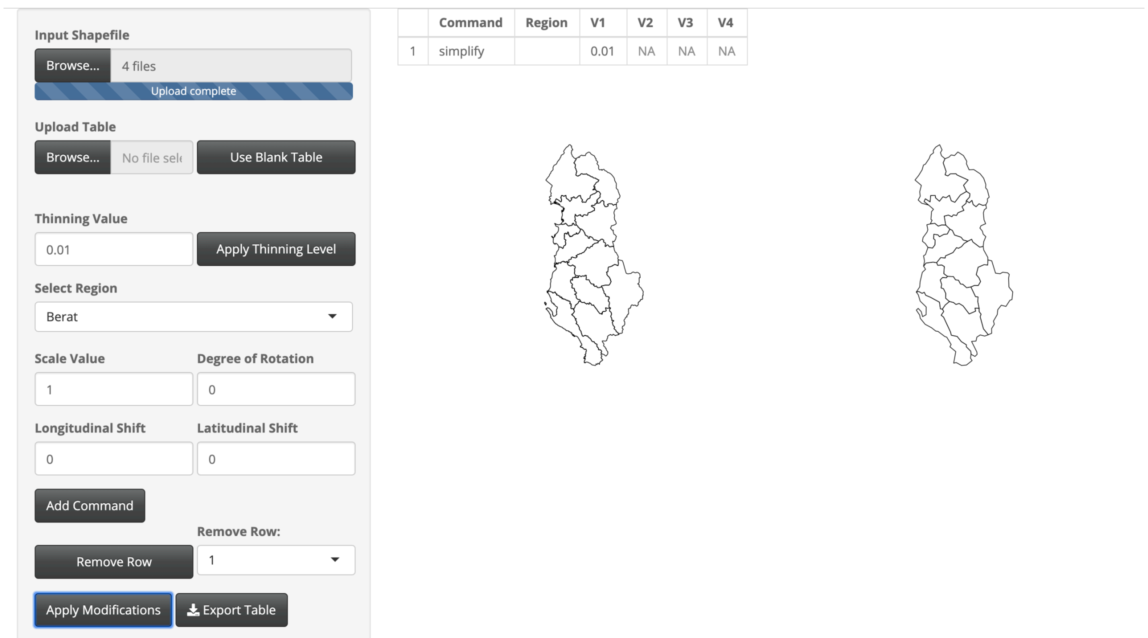


Fig. A.73: Modifications done to Albania.

Thinning was done to keep 1% of the original boundary points.

Andorra

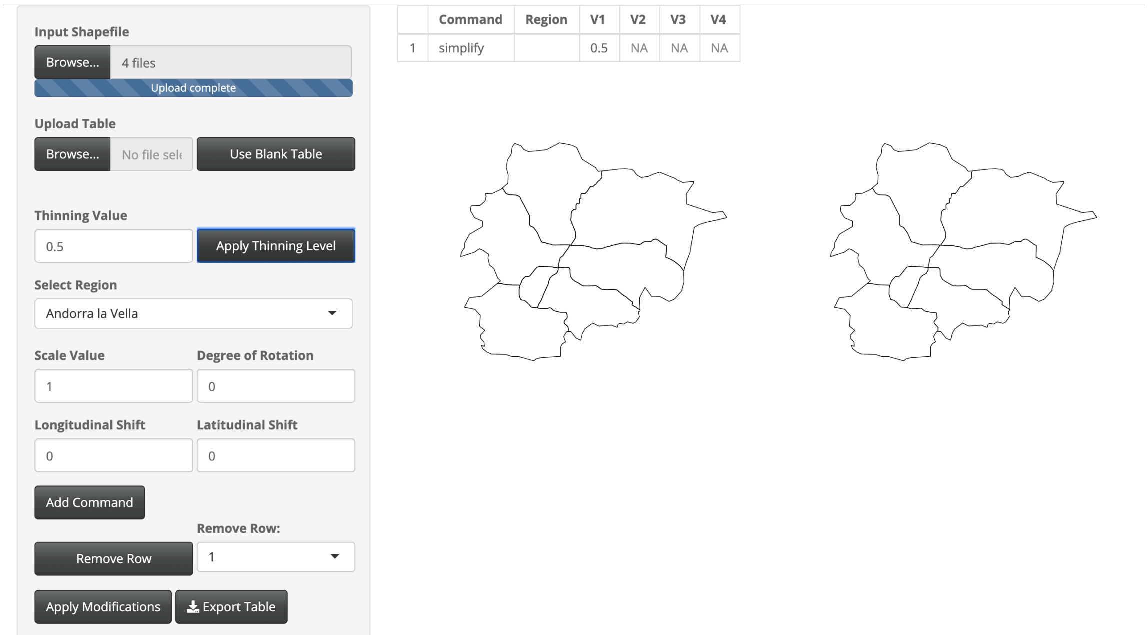


Fig. A.74: Modifications done to Andorra.

Thinning was done to keep 50% of the original boundary points.

Austria

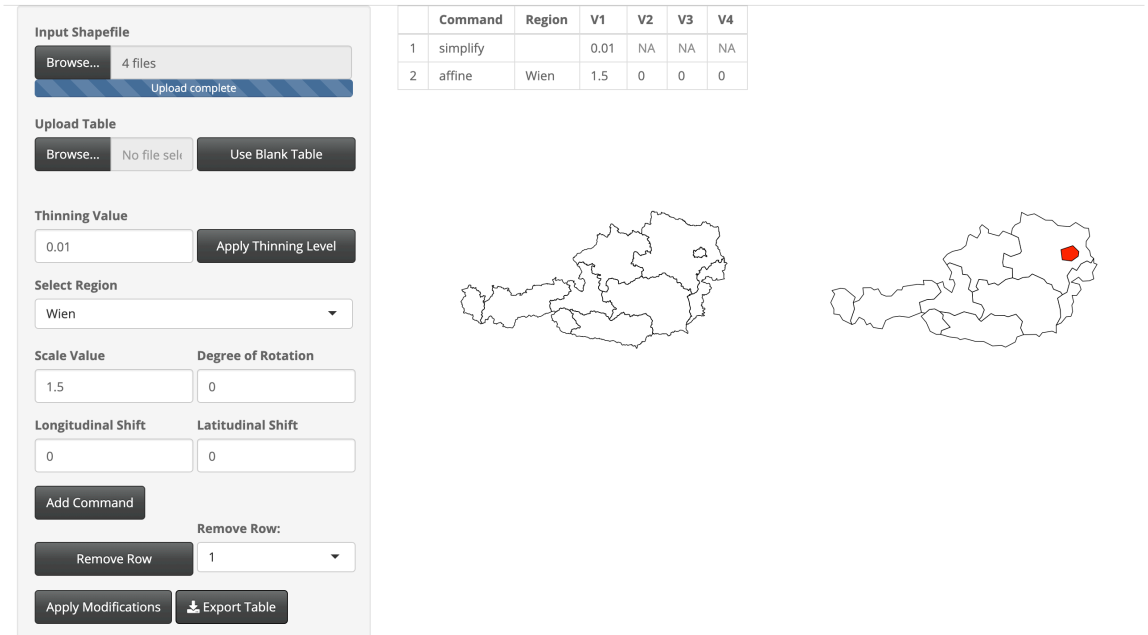


Fig. A.75: Modifications done to Austria.

Thinning was done to keep 1% of the original boundary points. The Wien region was scaled larger with a scale factor of 1.5.

Belarus

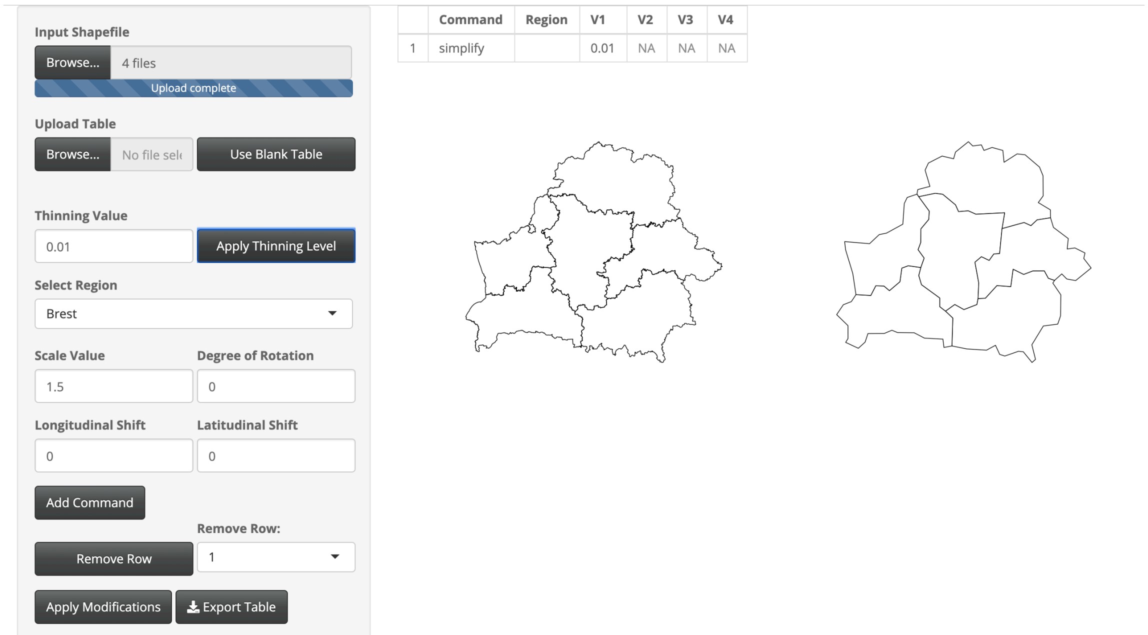


Fig. A.76: Modifications done to Belarus.

Thinning was done to keep 1% of the original boundary points.

Bulgaria

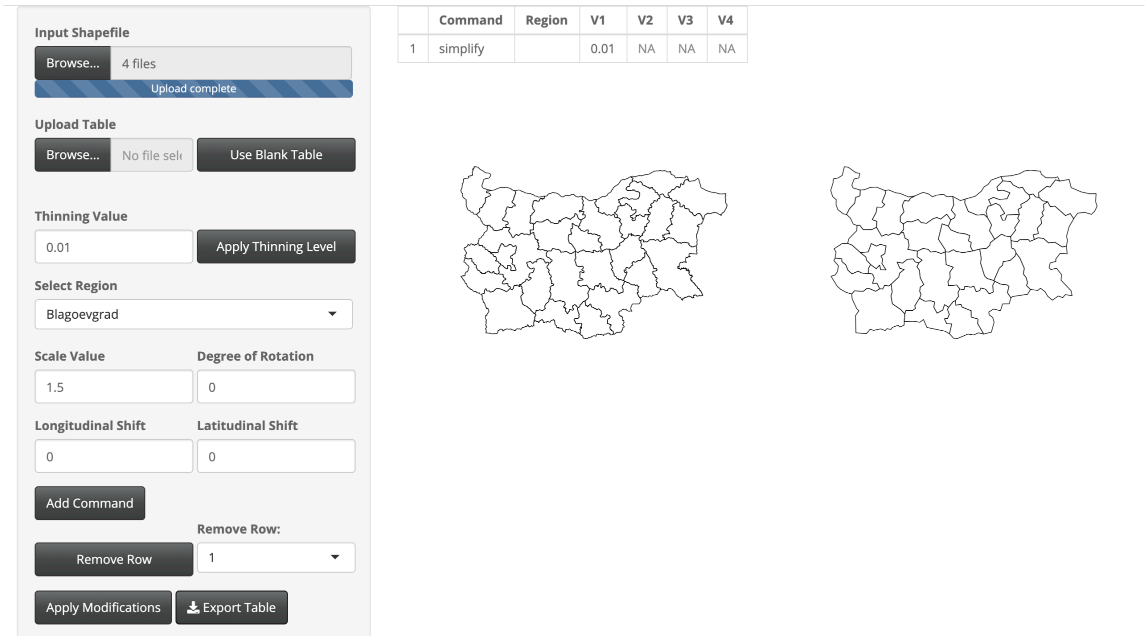


Fig. A.77: Modifications done to Bulgaria.

Thinning was done to keep 1% of the original boundary points.

Croatia

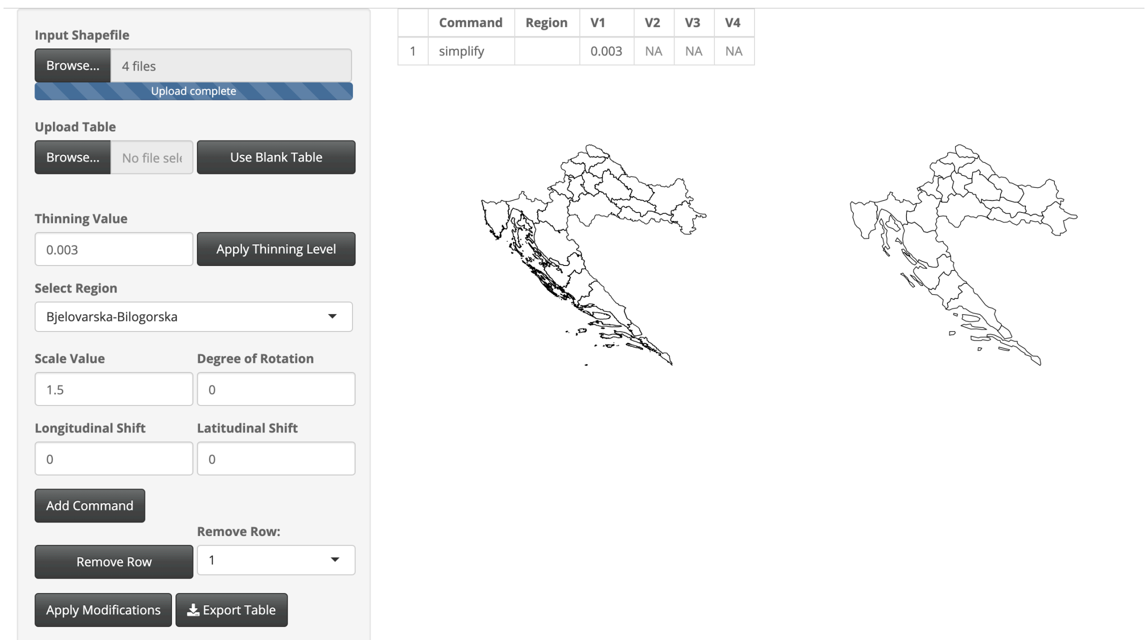


Fig. A.78: Modifications done to Croatia.

Thinning was done to keep 0.3% of the original boundary points.

Czech Republic

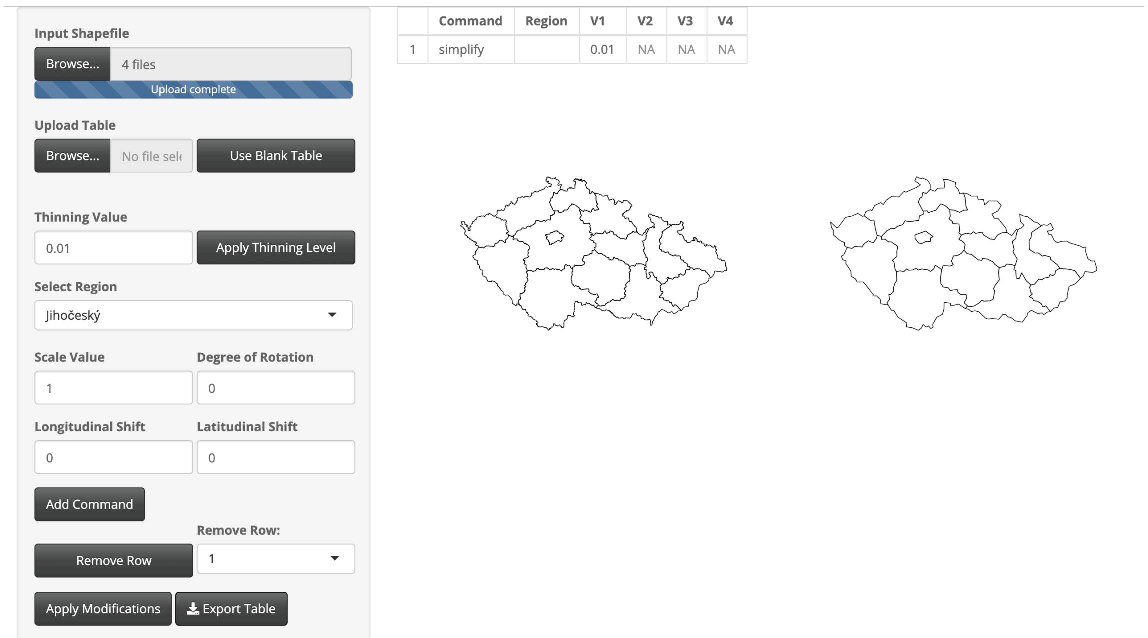


Fig. A.79: Modifications done to Czech Republic.

Thinning was done to keep 1% of the original boundary points.

Estonia

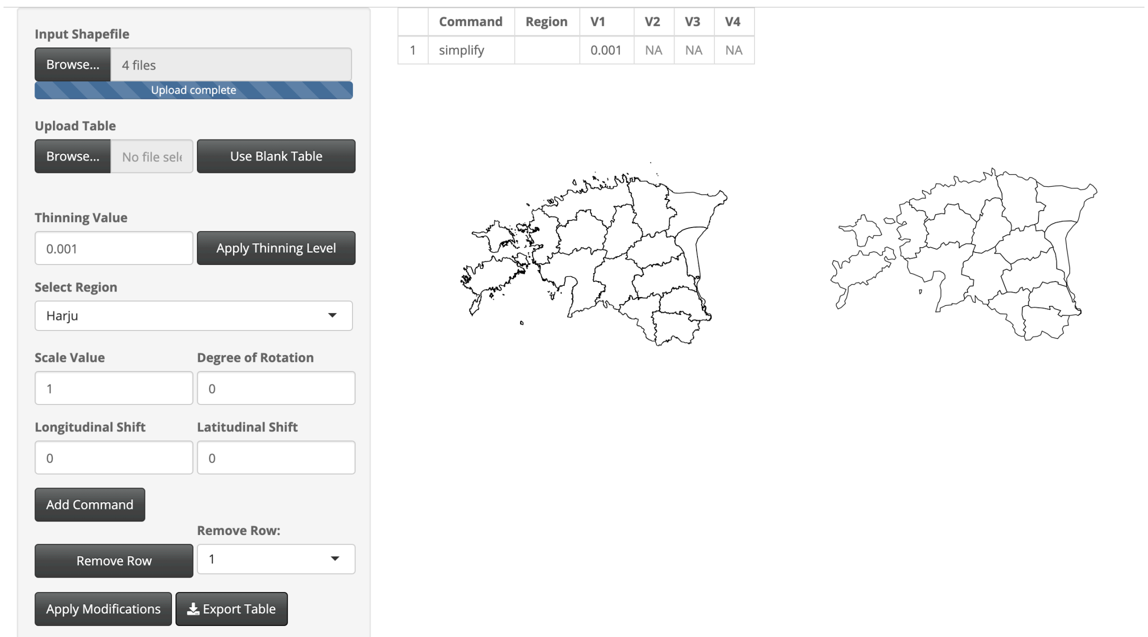


Fig. A.80: Modifications done to Estonia.

Thinning was done to keep 0.1% of the original boundary points.

France

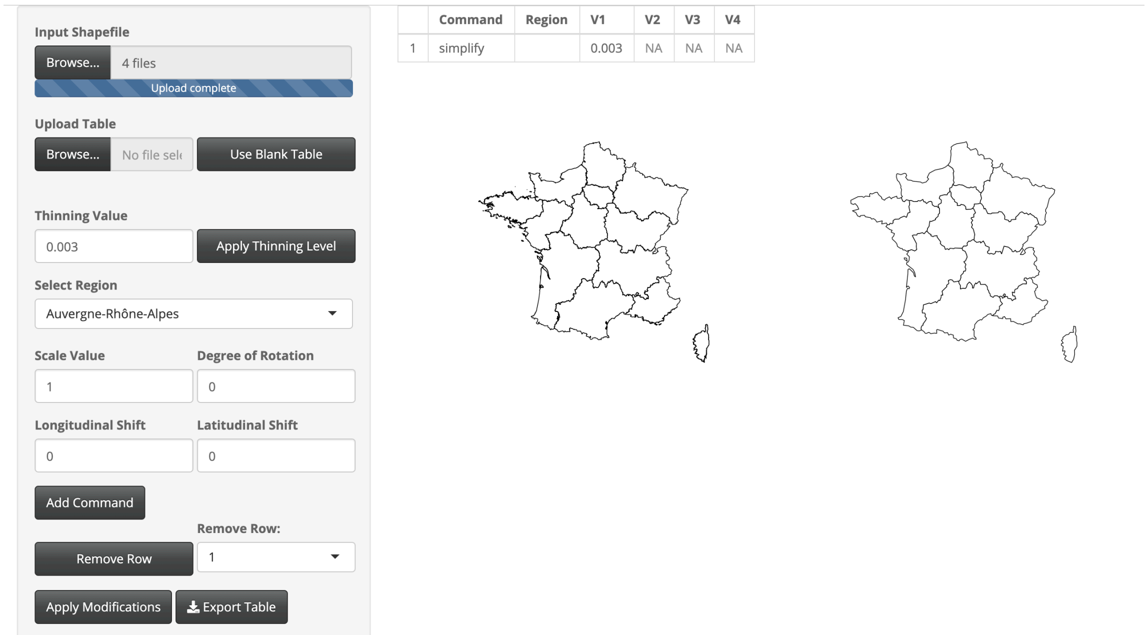


Fig. A.81: Modifications done to France.

Thinning was done to keep 0.3% of the original boundary points.

Germany

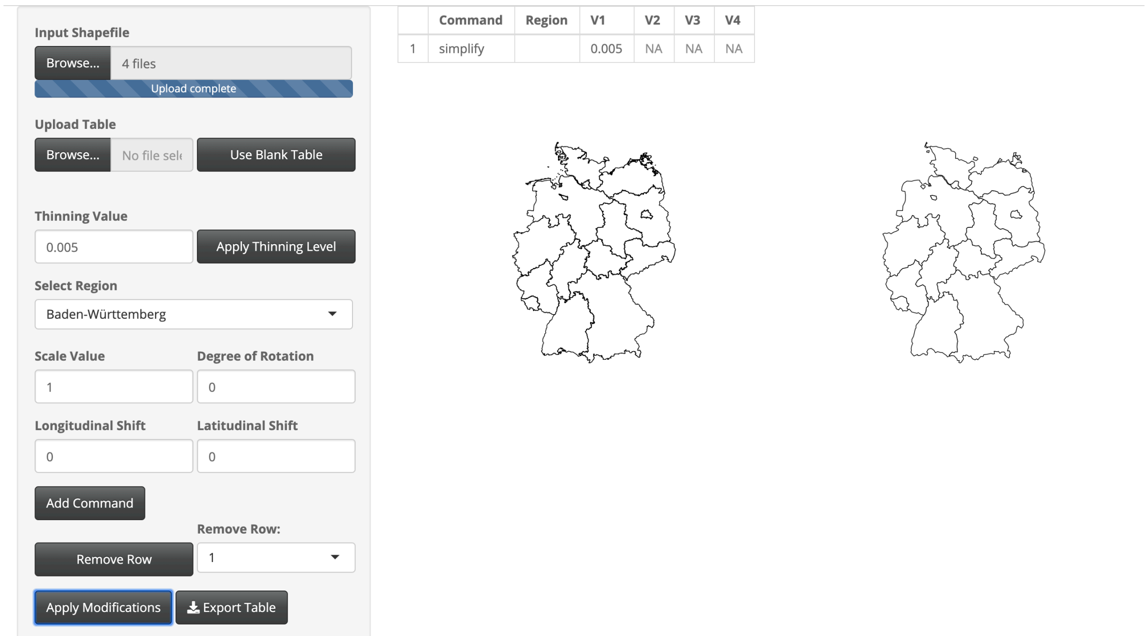


Fig. A.82: Modifications done to Germany.

Thinning was done to keep 0.5% of the original boundary points.

Greece

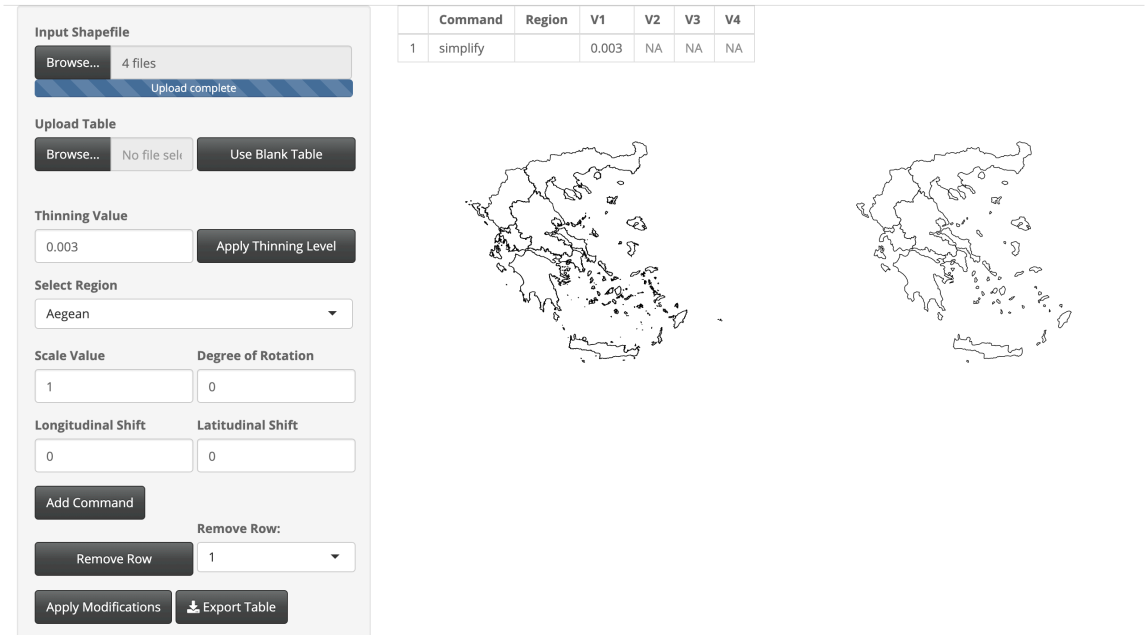


Fig. A.83: Modifications done to Greece.

Thinning was done to keep 0.3% of the original boundary points.

Hungary

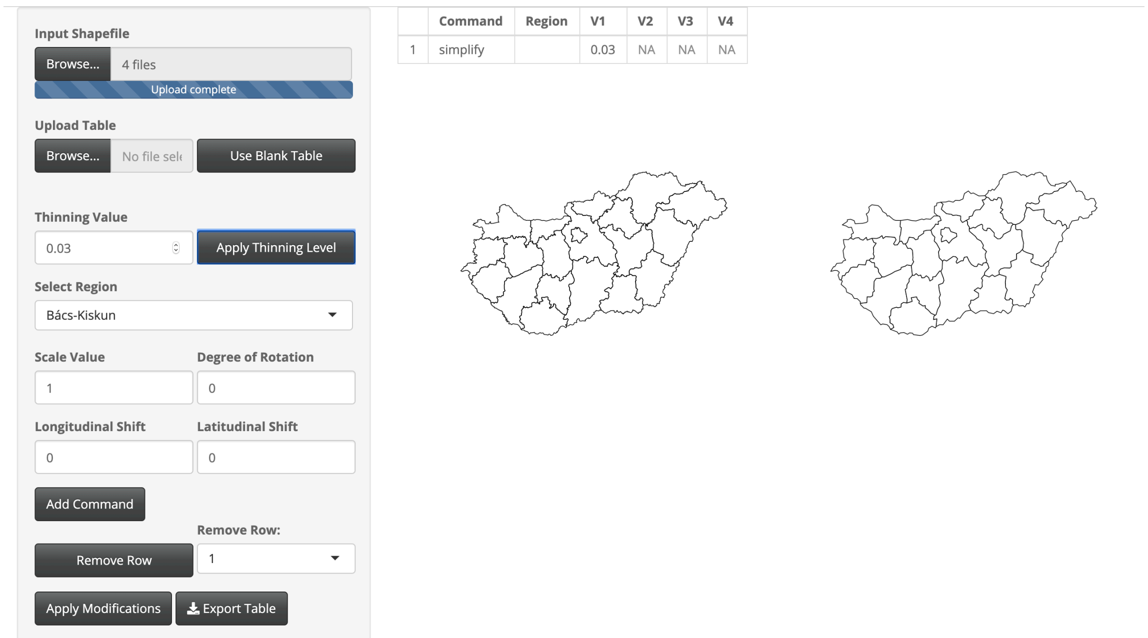


Fig. A.84: Modifications done to Hungary.

Thinning was done to keep 3% of the original boundary points.

Iceland

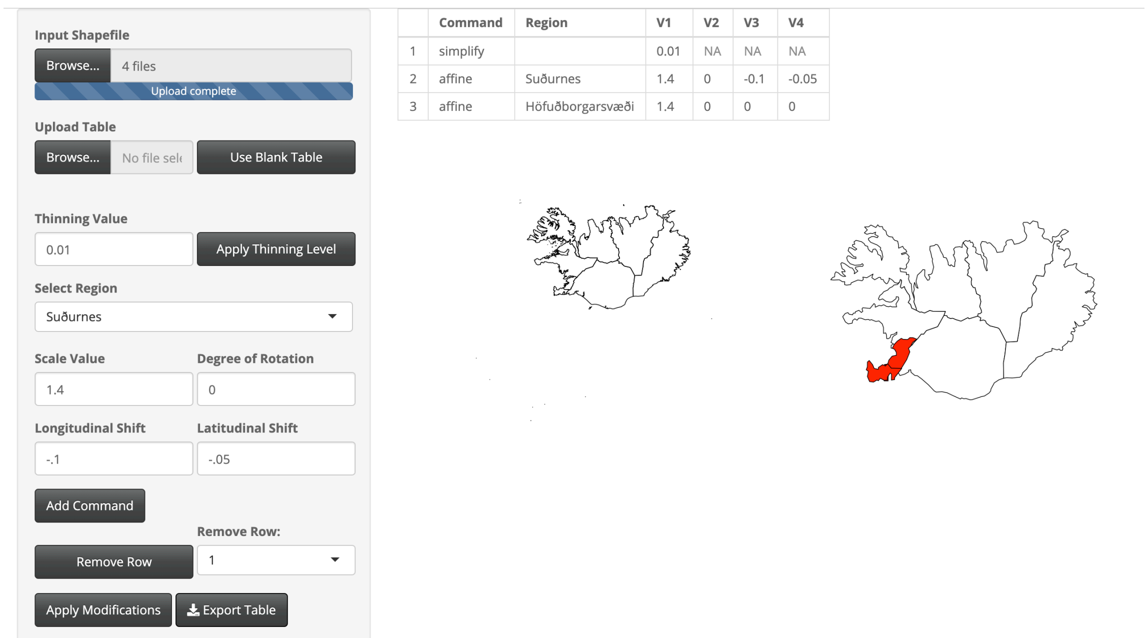


Fig. A.85: Modifications done to Iceland.

Thinning was done to keep 1% of the original boundary points. The Suðurnes (Southern Peninsula) region was scaled larger with a scale factor of 1.4, shifted west 0.1 degrees longitude and south 0.05 degrees latitude. The Höfuðborgarsvæðið (Greater Reykjavík) region was scaled larger with a scale factor of 1.4.

Ireland

Fig. A.86: Modifications done to Ireland.

Thinning was done to keep 0.5% of the original boundary points.

Italy

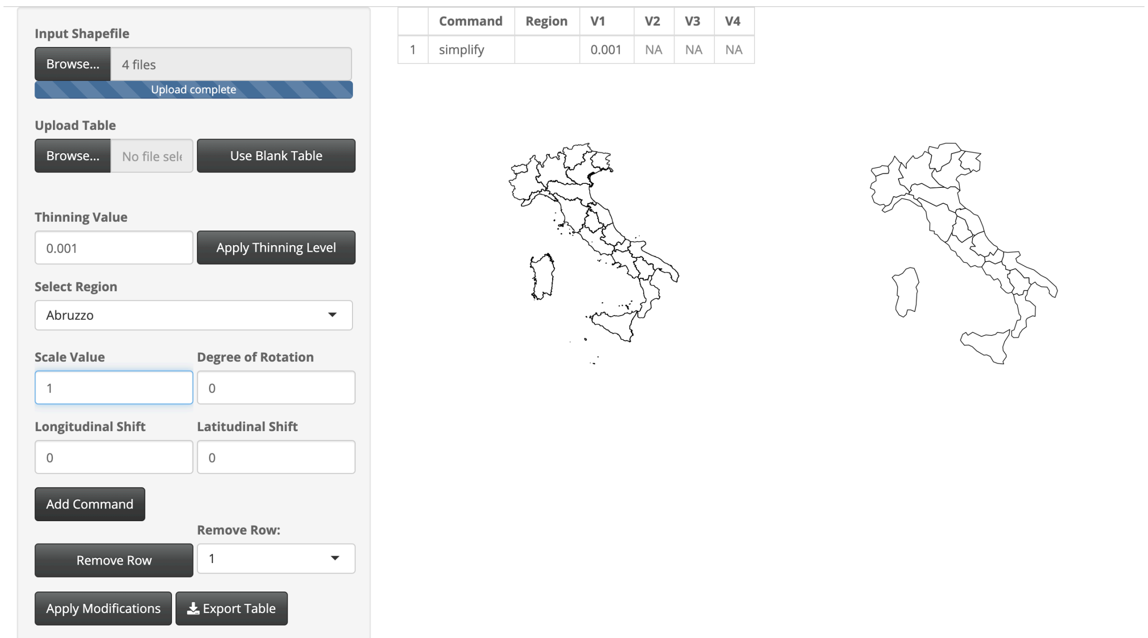


Fig. A.87: Modifications done to Italy.

Thinning was done to keep 0.1% of the original boundary points.

Kosovo

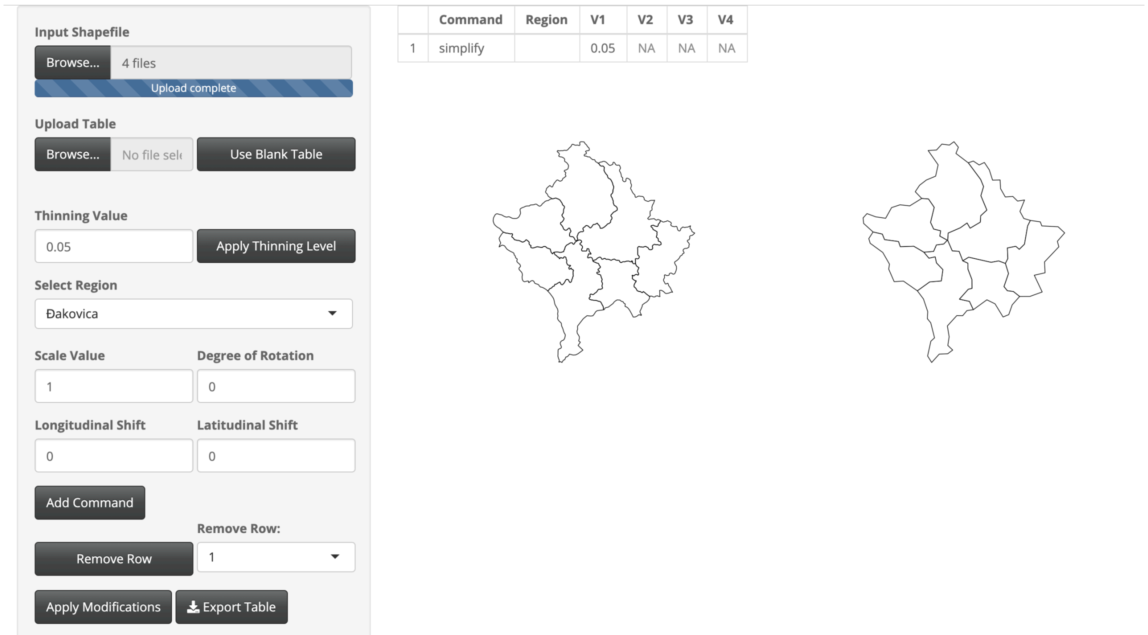


Fig. A.88: Modifications done to Kosovo.

Thinning was done to keep 5% of the original boundary points.

Lithuania

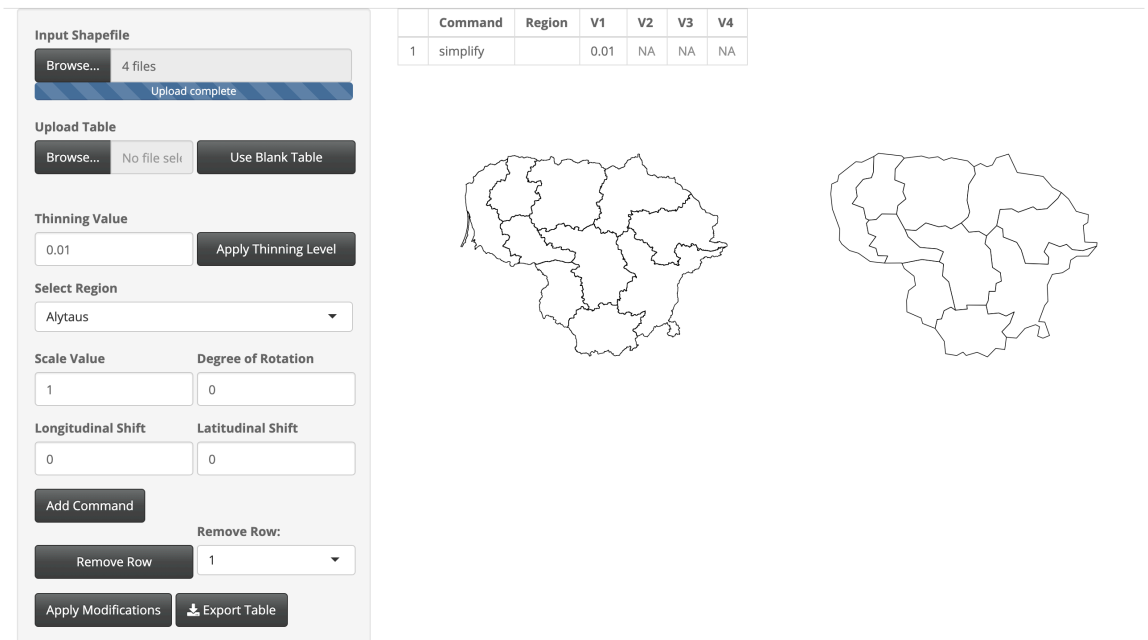


Fig. A.89: Modifications done to Lithuania.

Thinning was done to keep 1% of the original boundary points.

Moldova

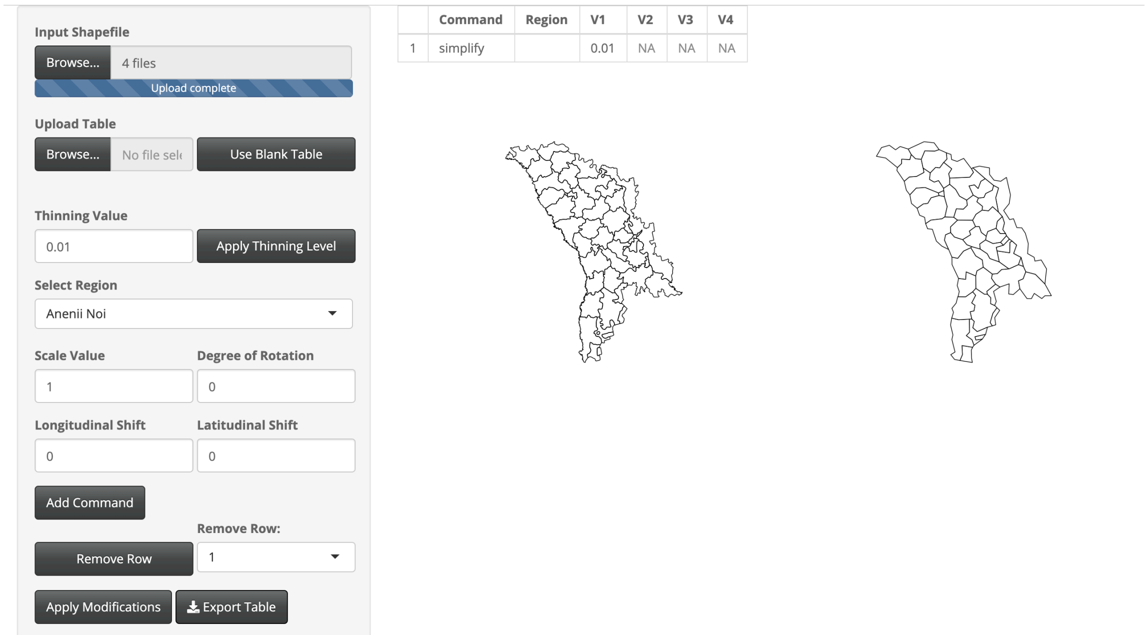


Fig. A.90: Modifications done to Moldova.

Thinning was done to keep 1% of the original boundary points.

Montenegro

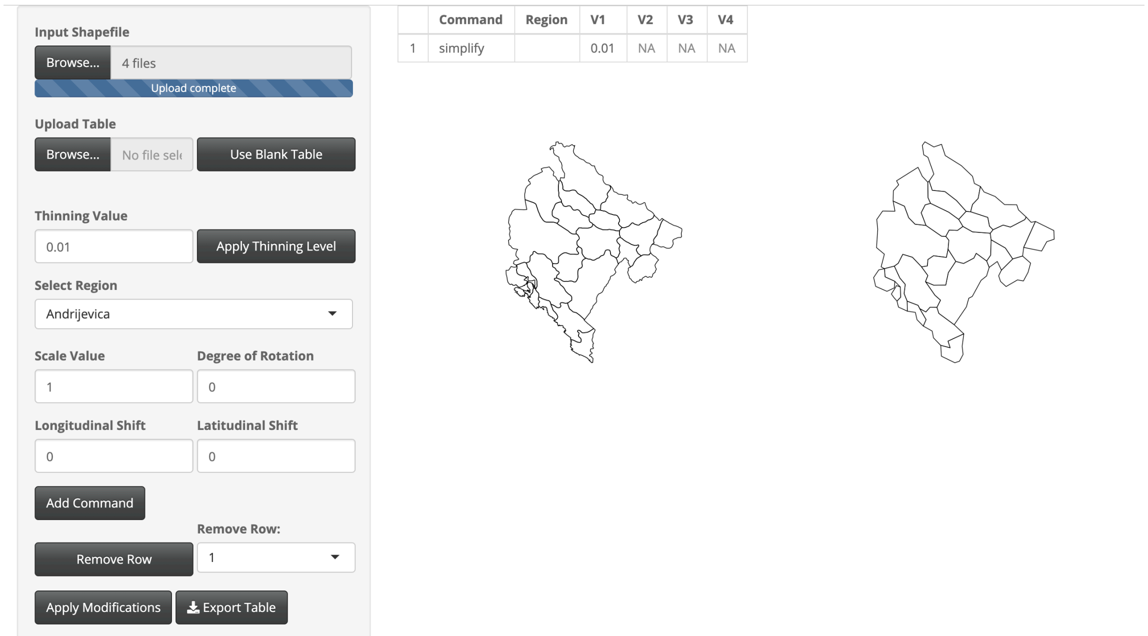


Fig. A.91: Modifications done to Montenegro.

Thinning was done to keep 1% of the original boundary points.

The Netherlands

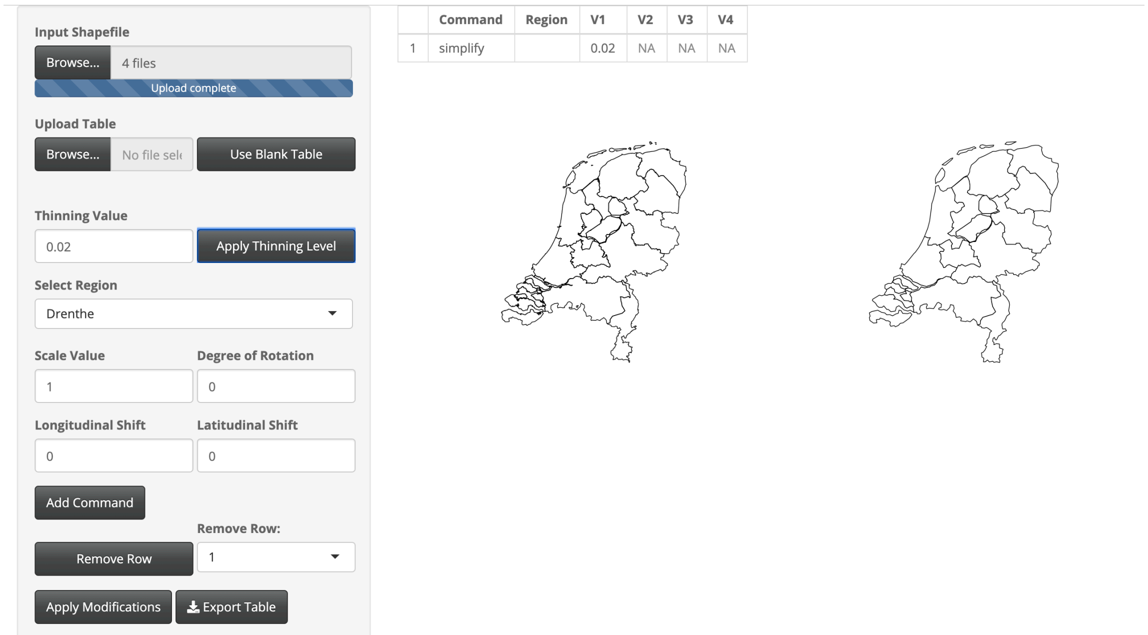


Fig. A.92: Modifications done to The Netherlands.

Thinning was done to keep 2% of the original boundary points.

Norway

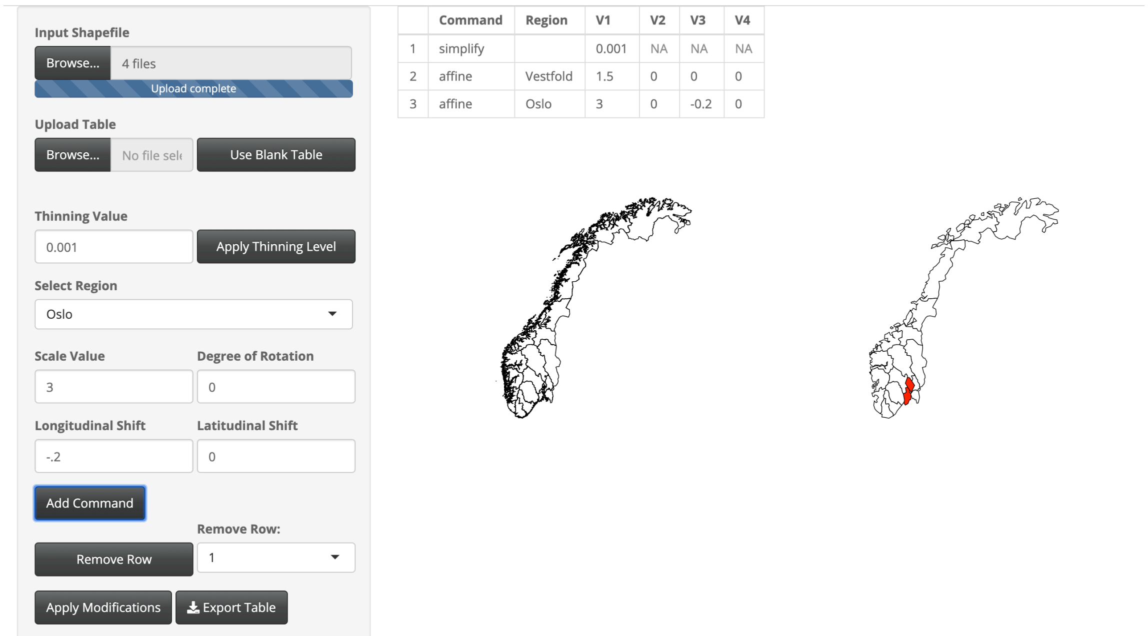


Fig. A.93: Modifications done to Norway.

Thinning was done to keep 0.1% of the original boundary points. The Vestfold region was scaled larger with a scale factor of 1.5. The Oslo region was scaled larger with a scale factor of 3 and shifted west 0.2 degrees longitude.

Poland

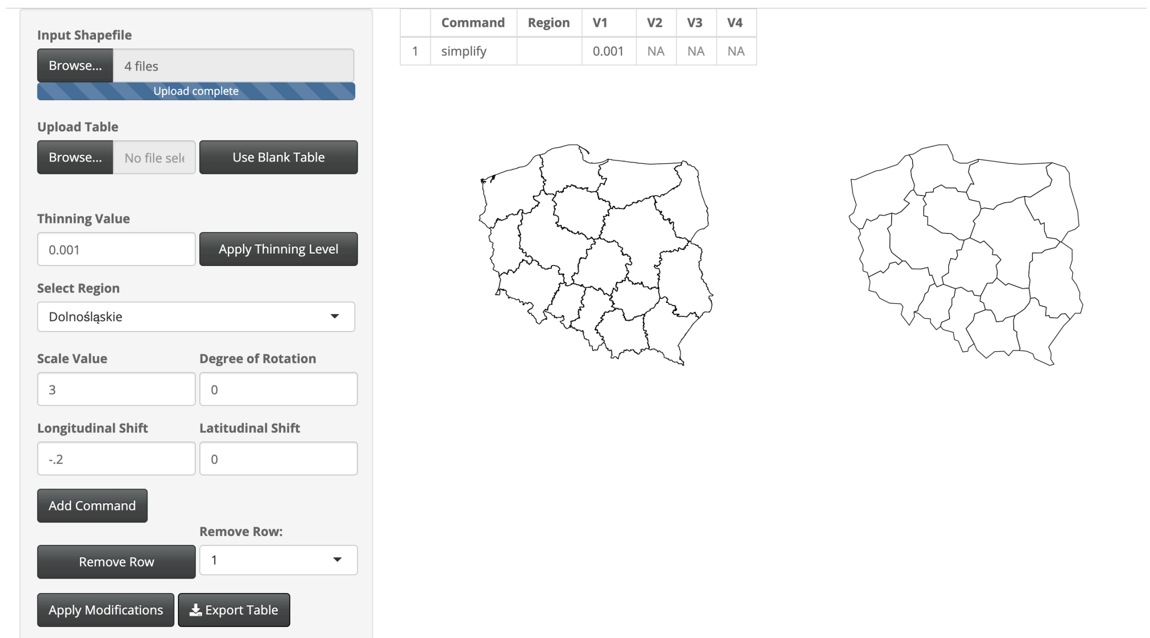


Fig. A.94: Modifications done to Poland.

Thinning was done to keep 0.1% of the original boundary points.

San Marino

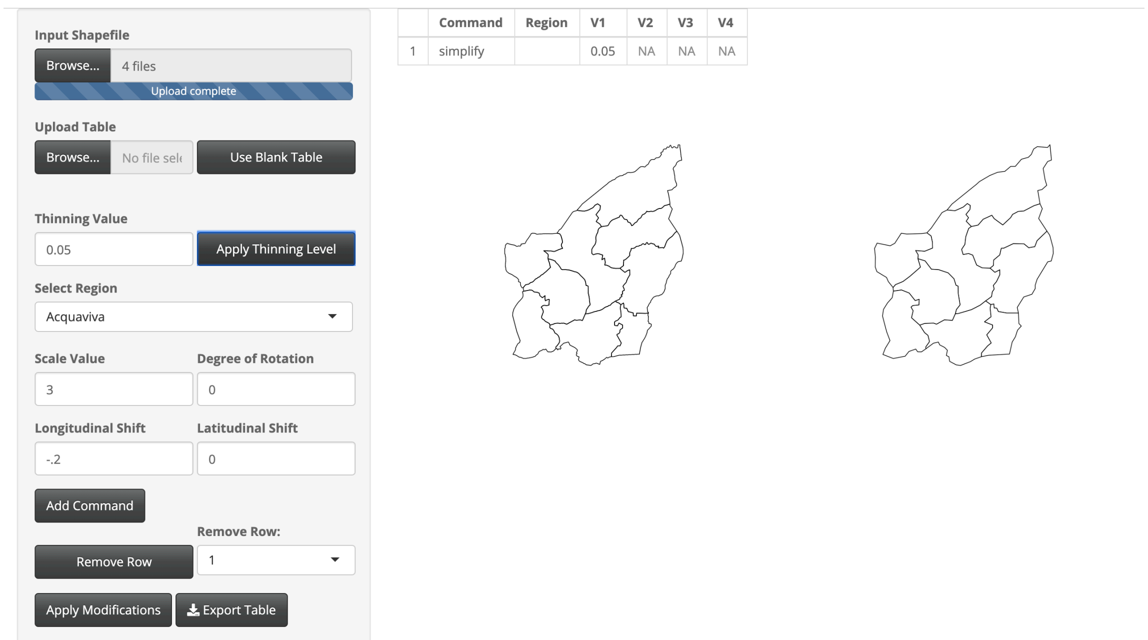


Fig. A.95: Modifications done to San Marino.

Thinning was done to keep 5% of the original boundary points.

Serbia

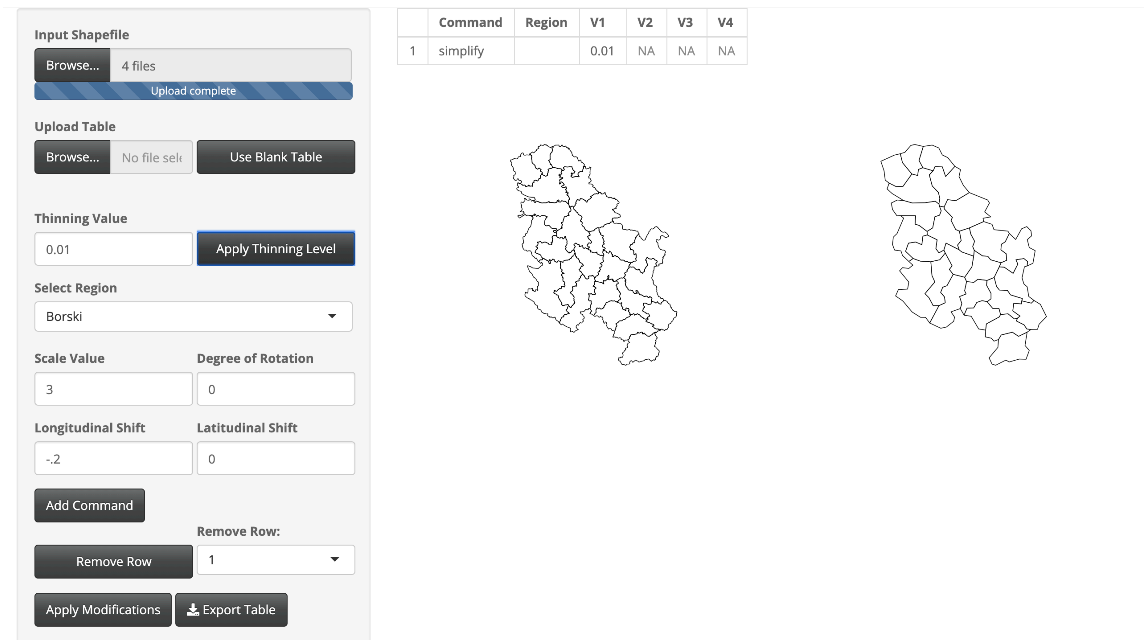


Fig. A.96: Modifications done to Serbia.

Thinning was done to keep 1% of the original boundary points.

Slovakia

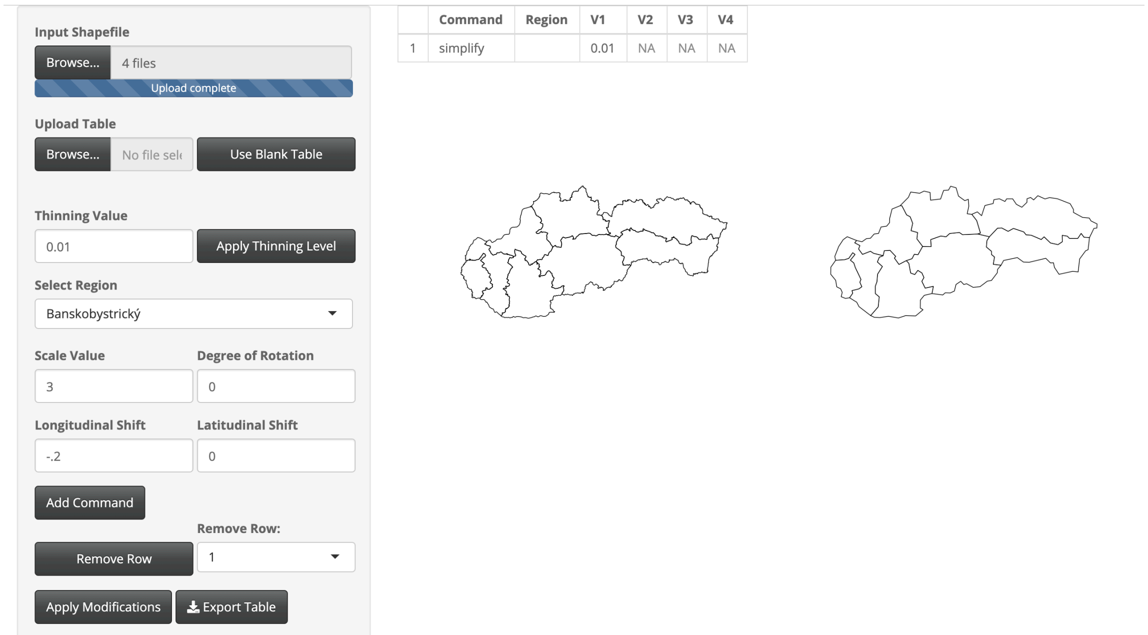


Fig. A.97: Modifications done to Slovakia.

Thinning was done to keep 1% of the original boundary points.

Slovenia

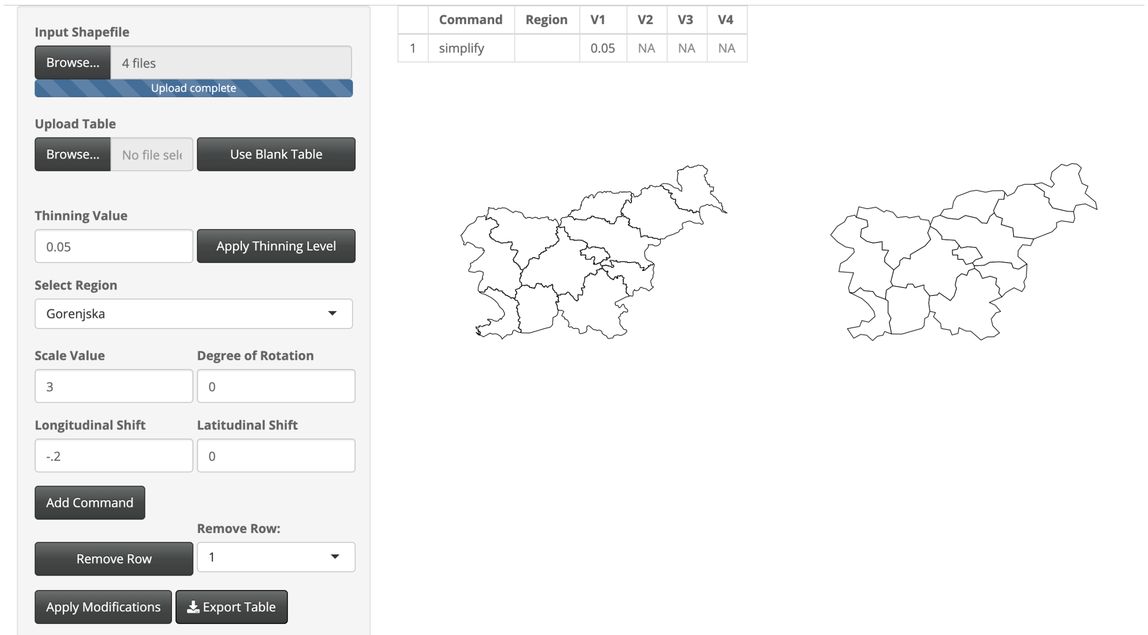


Fig. A.98: Modifications done to Slovenia.

Thinning was done to keep 5% of the original boundary points.

Spain

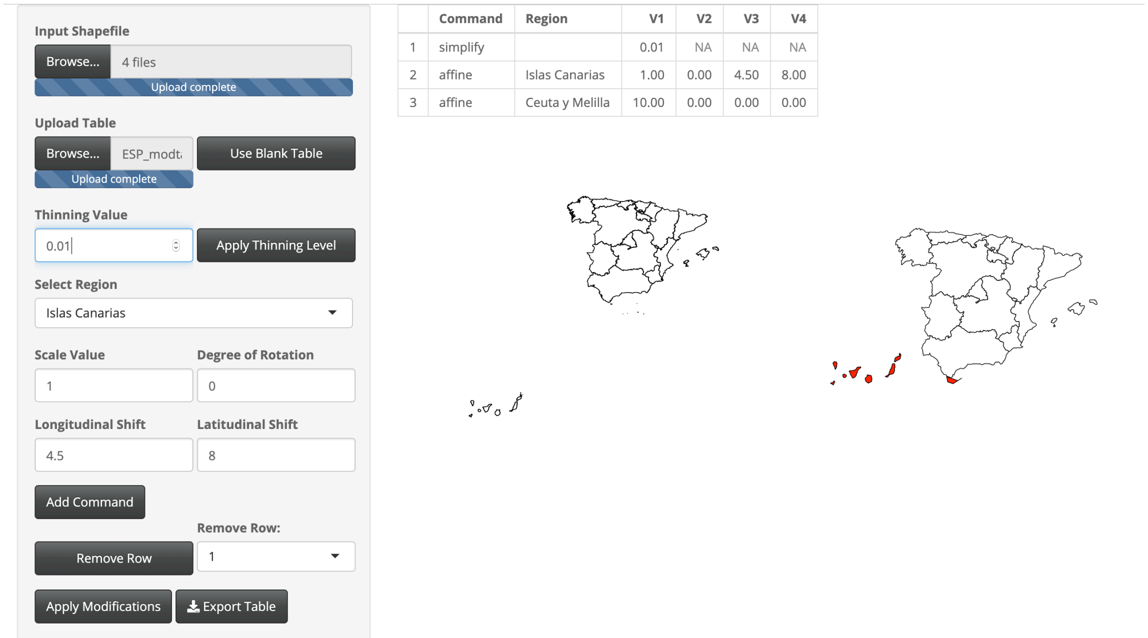


Fig. A.99: Modifications done to Spain.

Thinning was done to keep 1% of the original boundary points. The Islas Canarias region was shifted east 4.5 degrees longitude and north 8 degrees latitude. The Ceuta y Melilla region was scaled larger with a scale factor of 10.

Sweden

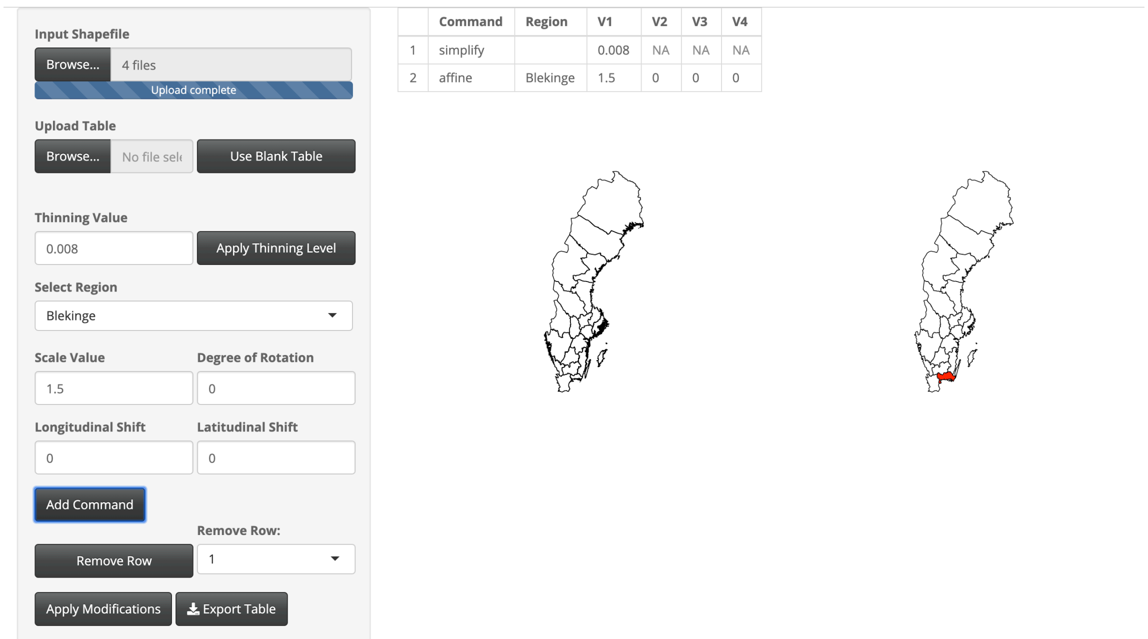


Fig. A.100: Modifications done to Sweden.

Thinning was done to keep 0.8% of the original boundary points. The Blekinge region was scaled larger with a scale factor of 1.5.

Switzerland

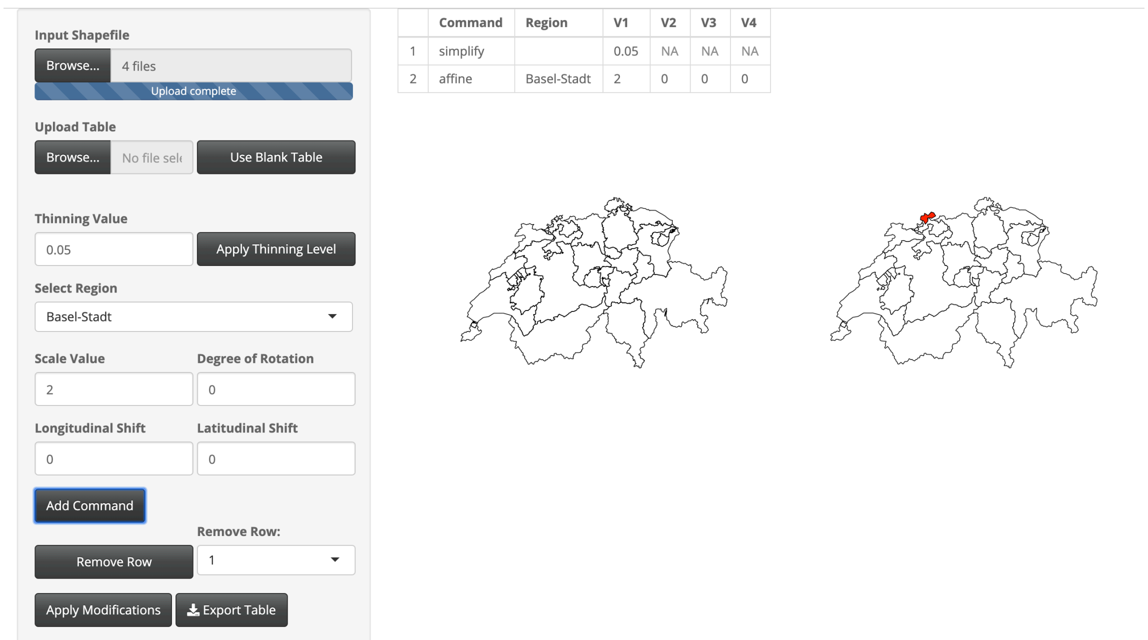


Fig. A.101: Modifications done to Switzerland.

Thinning was done to keep 5% of the original boundary points. The Basel-Stadt region was scaled larger with a scale factor of 2.

Ukraine

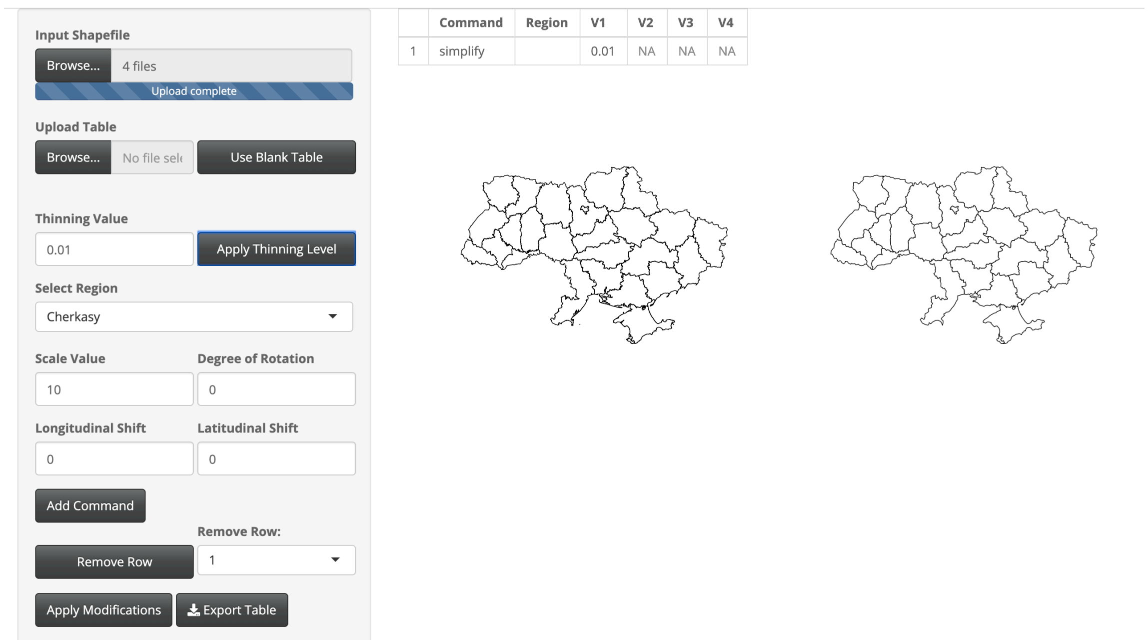


Fig. A.102: Modifications done to Ukraine.

Thinning was done to keep 1% of the original boundary points.

A.4 North America

Table A.4: Table listing North American countries modified and included in the 'LMshapemaker' R package. Countries indicated with a * were created at the command level outside of the 'LMshapemaker' Shiny app.

Country	**Page Number**
Antigua and Barbuda	172
Barbados	173
Belize	174
Canada*	175
Costa Rica	177
Cuba	178
Dominica	179
Dominican Republic	180
El Salvador	181
Grenada	182
Guatemala	183
Haiti	184
Honduras	185
Jamaica	186
Mexico	187
Nicaragua*	188
Panama	190
Saint Kitts and Nevis	191
Saint Lucia	192
Saint Vincent and the Grenadines	193
Trinidad and Tobago	194

The following countries have been excluded from the ‘LMshapemaker’ R package for the reasons indicated.

- The Bahamas were excluded for complicated archipelagic geography.
- The United States of America was excluded for already being available for use in the ‘micromap’ R package.
- Greenland and Puerto Rico were excluded for not being members of the UN Member States.

Antigua and Barbuda

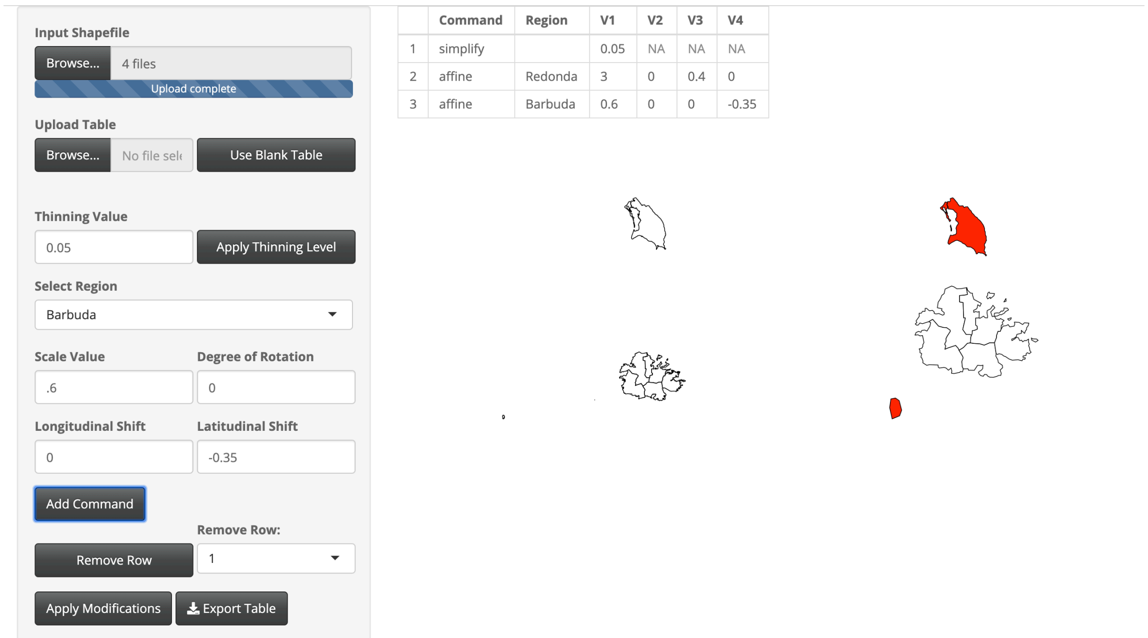


Fig. A.103: Modifications done to Antigua and Barbuda.

Thinning was done to keep 5% of the original boundary points. The Redonda region was scaled larger with a scale factor of 3 and shifted east 0.4 degrees longitude. The Barbuda region was scaled smaller with a scale factor of 0.6 and shifted south 0.35 degrees latitude.

Barbados

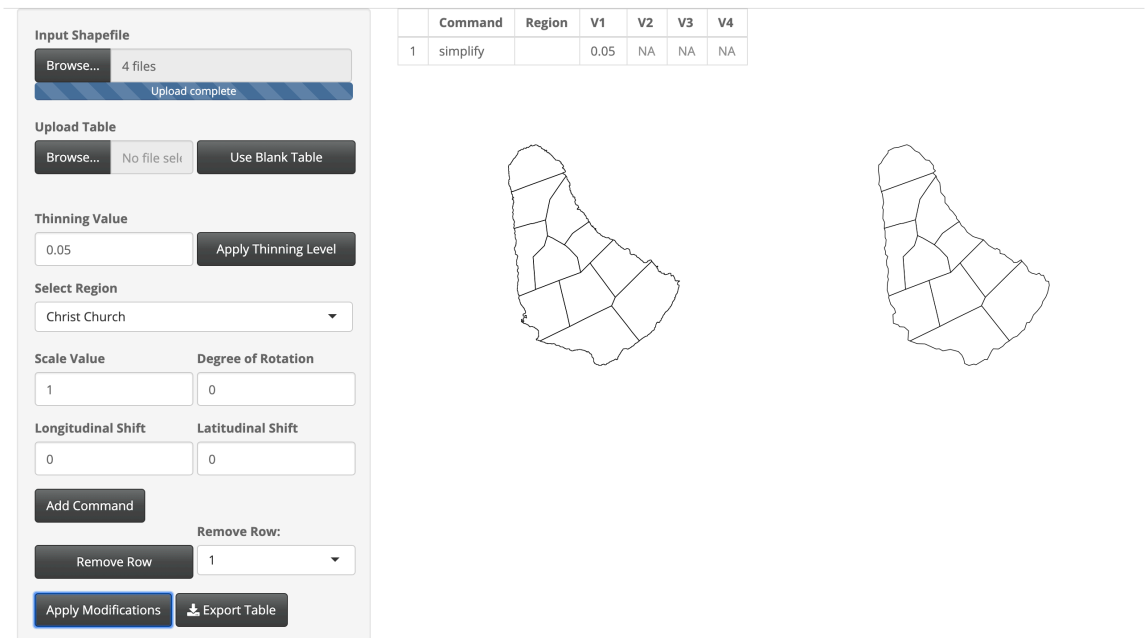


Fig. A.104: Modifications done to Barbados.

Thinning was done to keep 5% of the original boundary points.

Belize

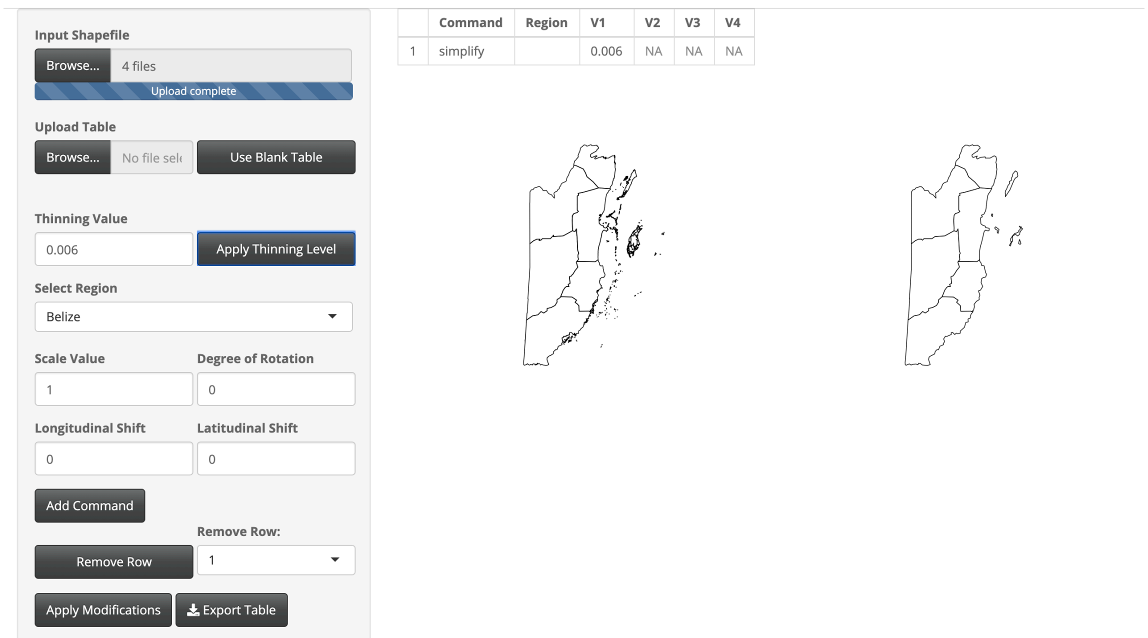


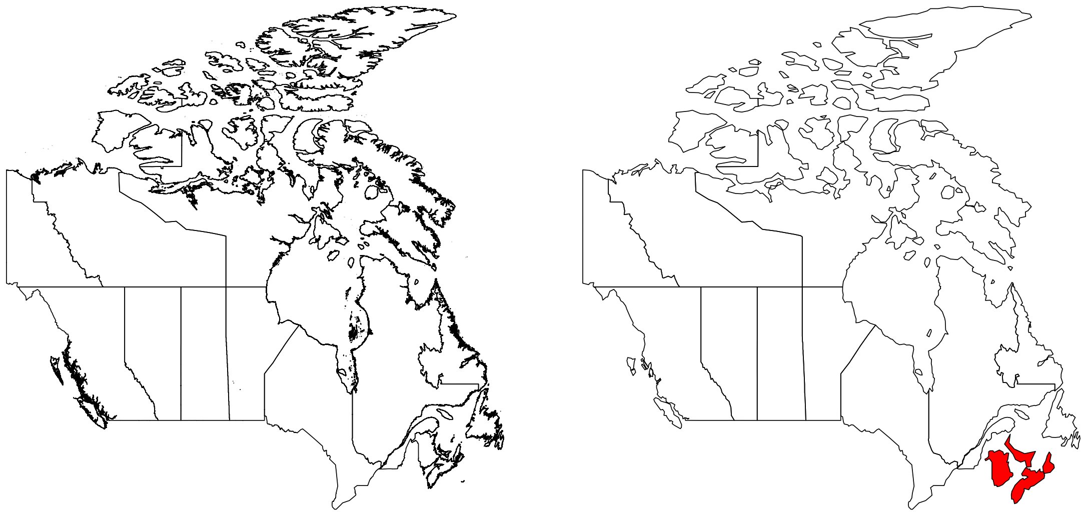
Fig. A.105: Modifications done to Belize.

Thinning was done to keep 0.6% of the original boundary points.

Canada

```
library(rgdal)
library(rgeos)
library(geojsonio)
library(rmapshaper)
library(sp)
library(dplyr)

shape <- readOGR(dsn = "gadm36_CAN_shp", layer = "gadm36_CAN_1", verbose = TRUE)
CAN_json <- geojson_json(shape, geometry = "polygon", group = "group")

# Unmodified Shapefile
CAN_sp <- geojson_sp(CAN_json)

# Modifed Shapefile
CAN_simplified <- CAN_sp %>%
  ms_simplify(keep = .00045) %>%
  geojson_json() %>%
  apply_mapshaper_commands("-affine scale=1.2 shift=2,-1 where='NAME_1 == \"Nova
      Scotia\"'
                           -affine shift=-.5,-1.5 where='NAME_1 == \"New Brunswick\"'
                           -affine scale=2.5 where='NAME_1 == \"Prince Edward Island\
                               "'",
                           force_FC = TRUE) %>%
  geojson_sp()
```

Thinning was done to keep 0.045% of the original boundary points. The Nova Scotia region was scaled larger with a scale factor of 1.2, shifted east 2 degrees longitude, and south 1 degree latitude. The New Brunswick region was shifted west 0.5 degrees longitude and south 1.5 degrees latitude. The Prince Edward Island region was scaled larger with a scale factor of 2.5.



(a) (b)

Fig. A.106: Canada shown (a) with no modification and (b) with simplification to boundaries and modifications to some regions (colored in red).

Costa Rica

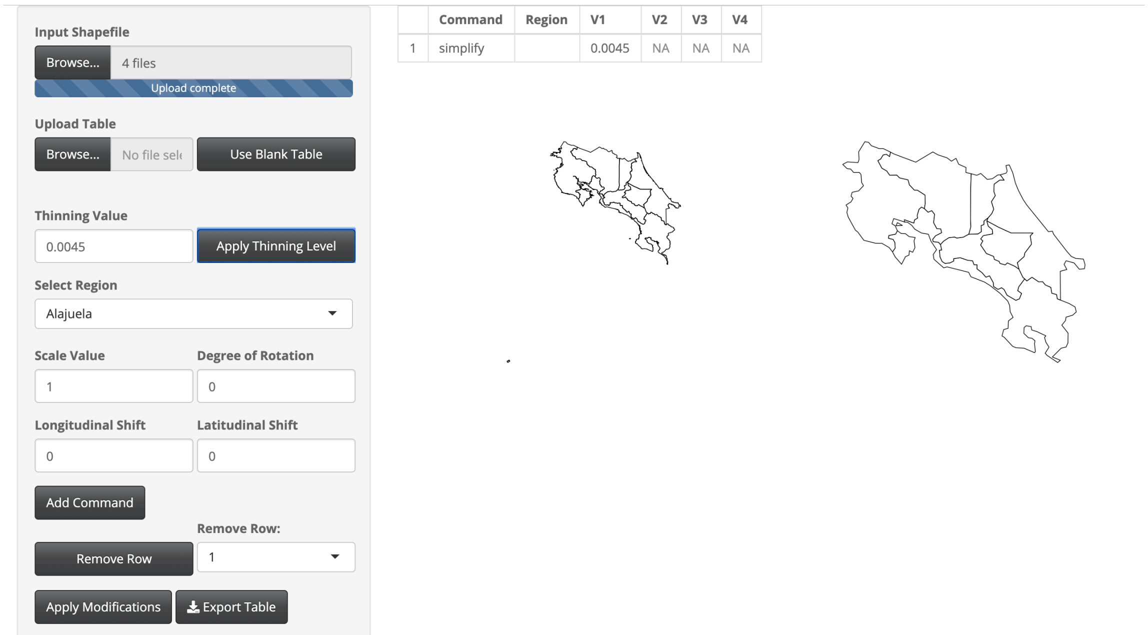


Fig. A.107: Modifications done to Costa Rica.

Thinning was done to keep 0.45% of the original boundary points.

Cuba

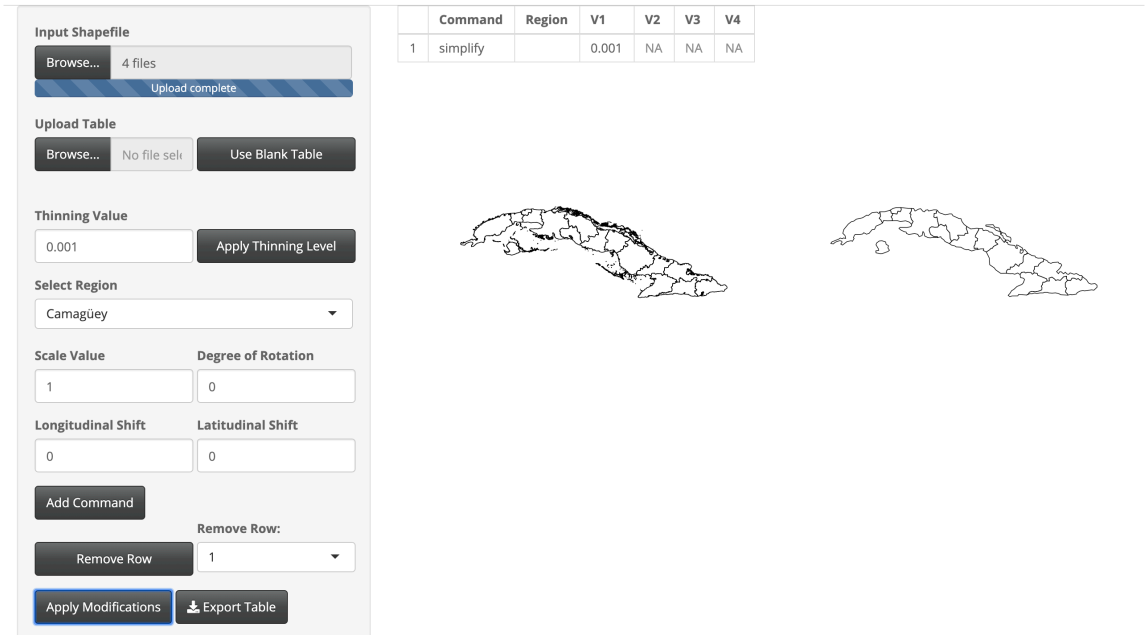


Fig. A.108: Modifications done to Cuba.

Thinning was done to keep 0.1% of the original boundary points.

Dominica

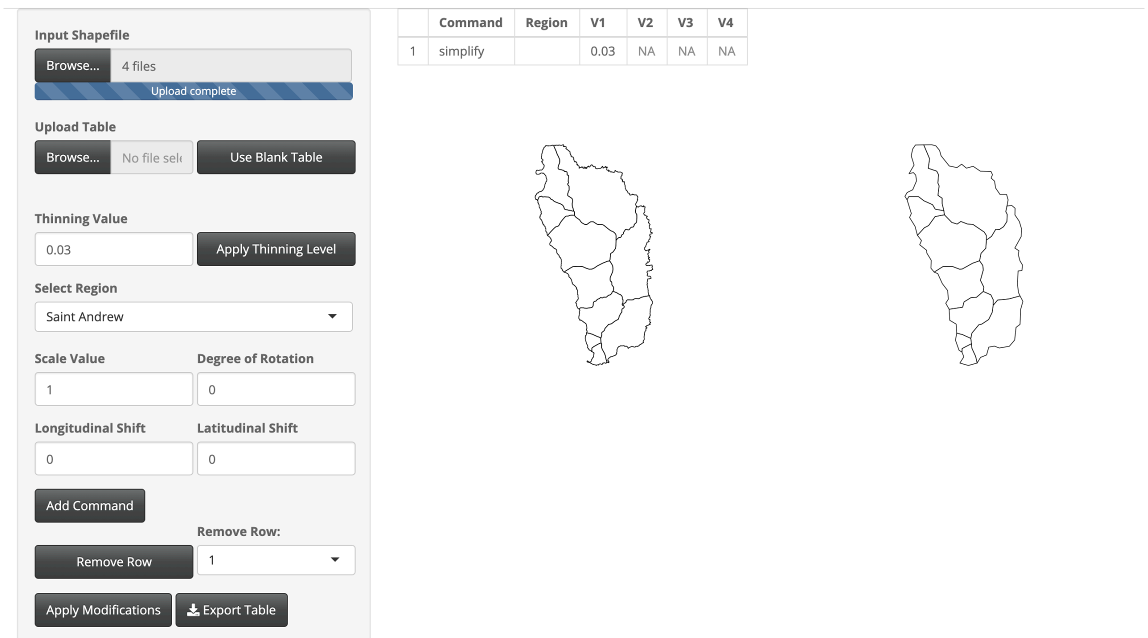


Fig. A.109: Modifications done to Dominica.

Thinning was done to keep 3% of the original boundary points.

Dominican Republic

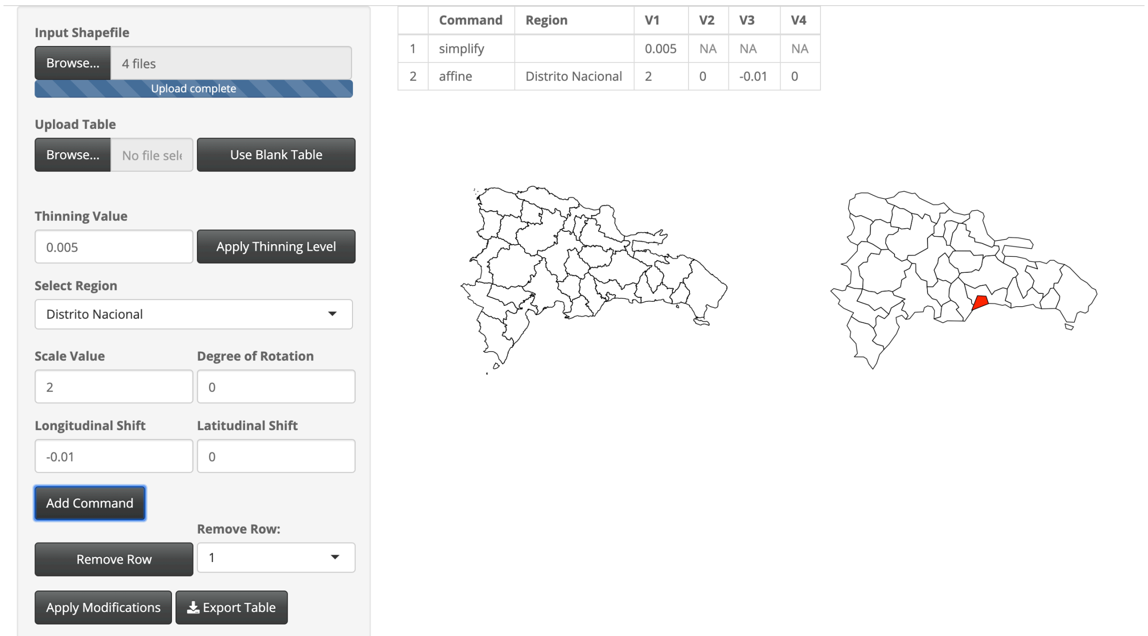


Fig. A.110: Modifications done to Dominican Republic.

Thinning was done to keep 0.5% of the original boundary points. The Distrito Nacional region was scaled larger with a scale factor of 2 and shifted west 0.01 degrees longitude.

El Salvador

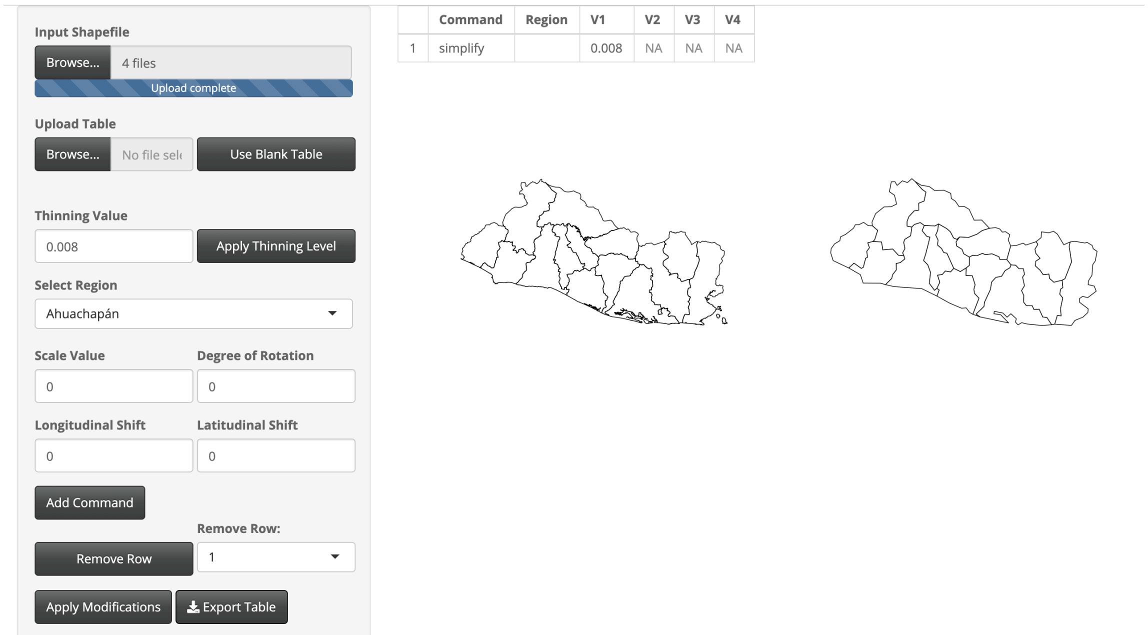


Fig. A.111: Modifications done to El Salvador.

Thinning was done to keep 0.8% of the original boundary points.

Grenada

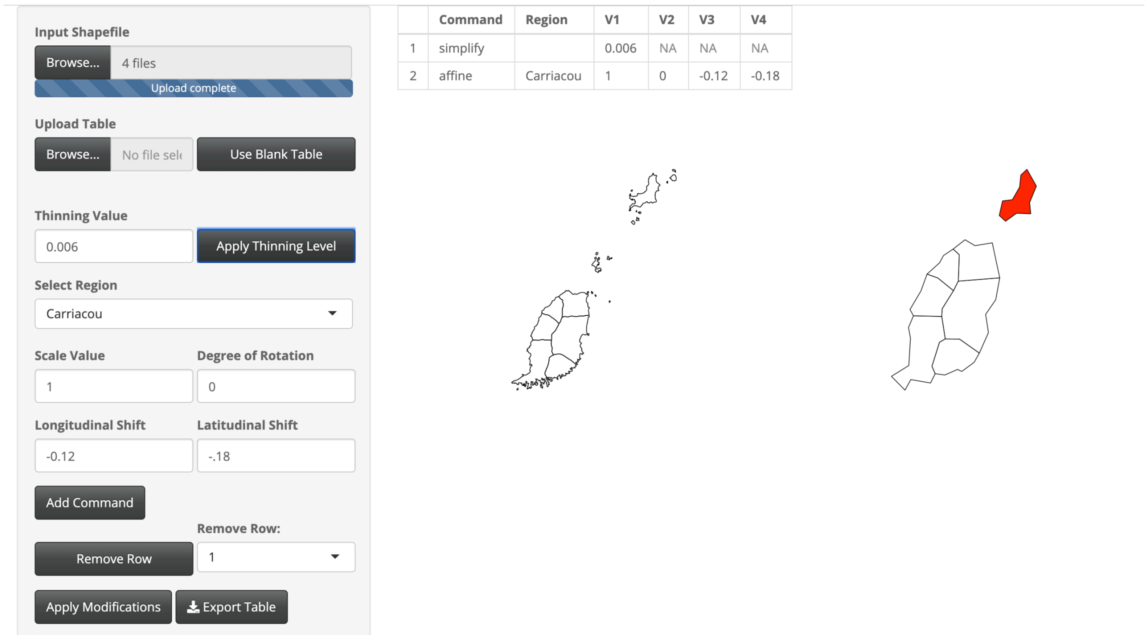


Fig. A.112: Modifications done to Grenada.

Thinning was done to keep 0.6% of the original boundary points. The Carriacou region was shifted west 0.12 degrees longitude and south 0.18 degrees latitude.

Guatemala

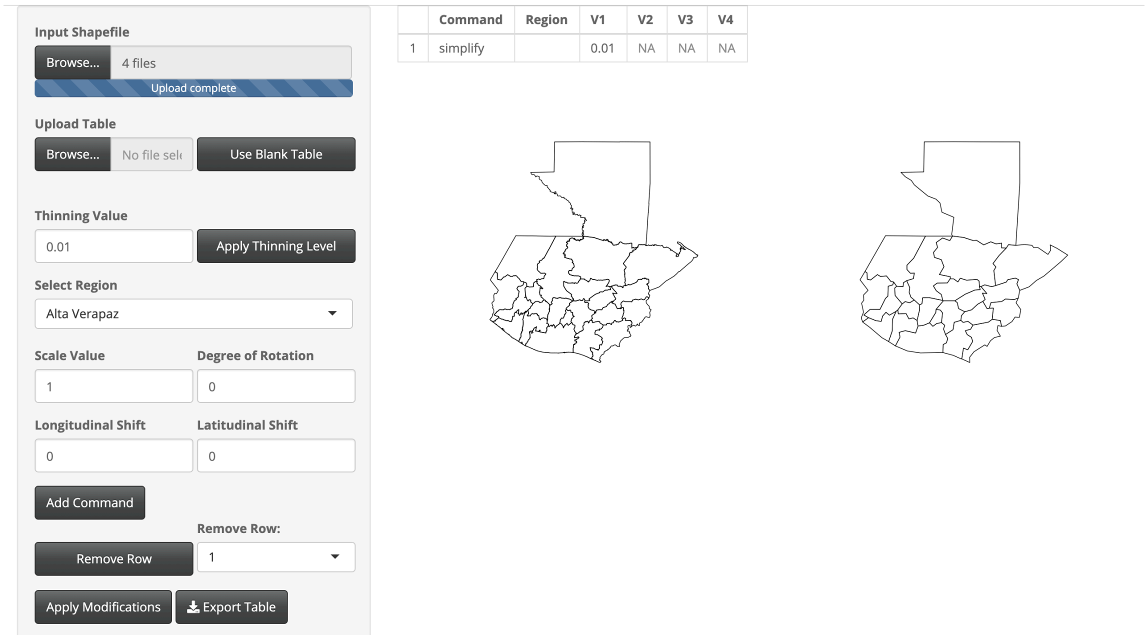


Fig. A.113: Modifications done to Guatemala.

Thinning was done to keep 1% of the original boundary points.

Haiti

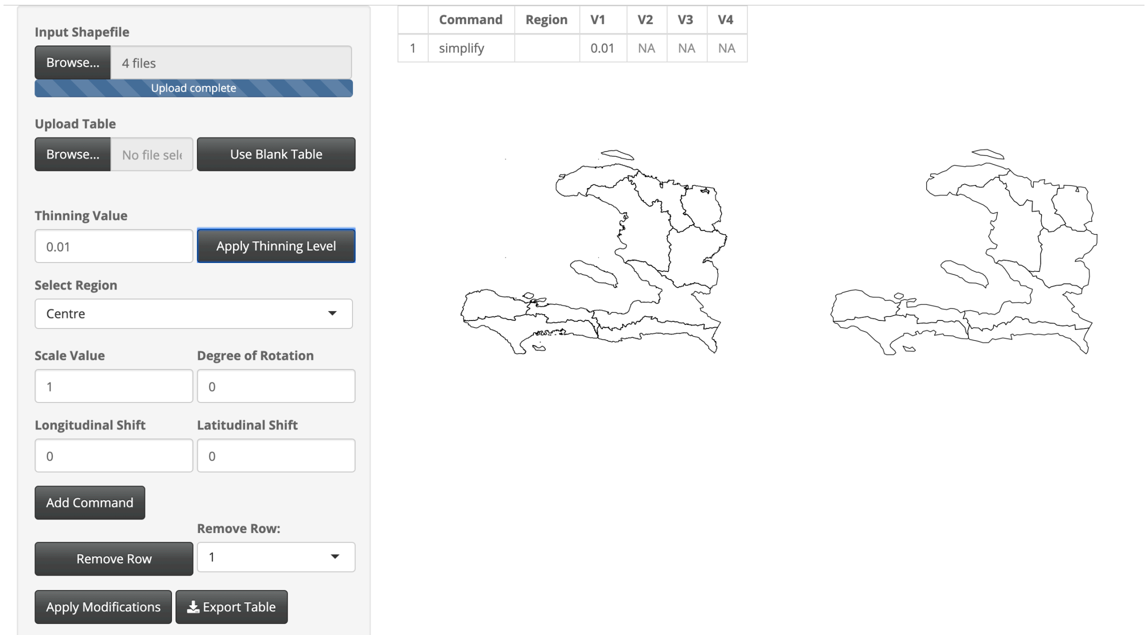


Fig. A.114: Modifications done to Haiti.

Thinning was done to keep 1% of the original boundary points.

Honduras

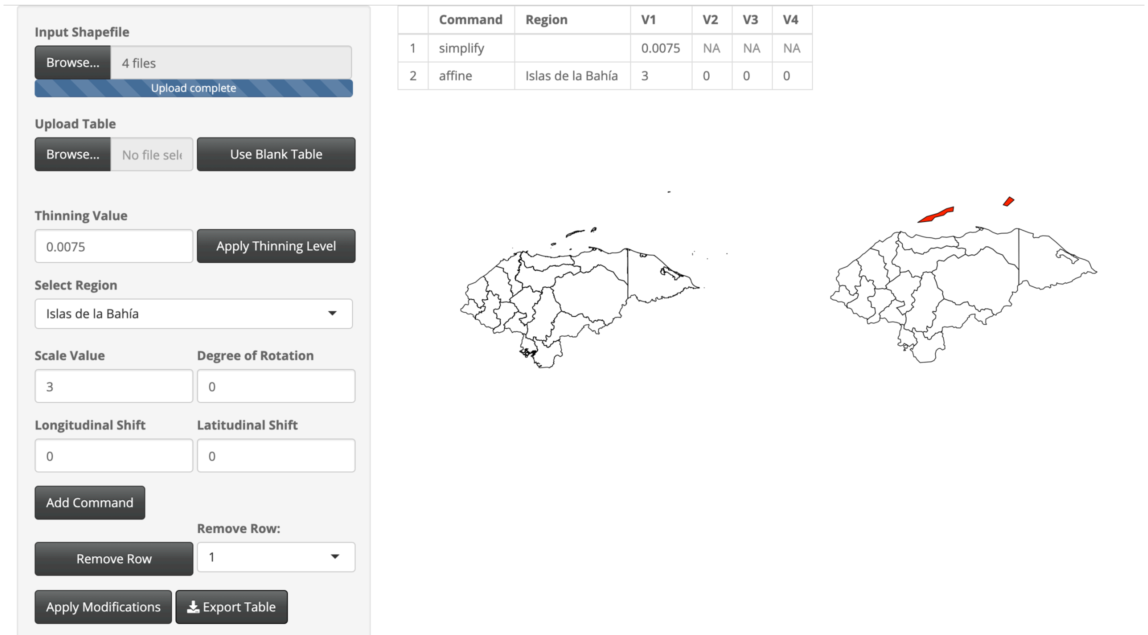


Fig. A.115: Modifications done to Honduras.

Thinning was done to keep 0.75% of the original boundary points. The Islas de la Bahía region was scaled larger with a scale factor of 3.

Jamaica

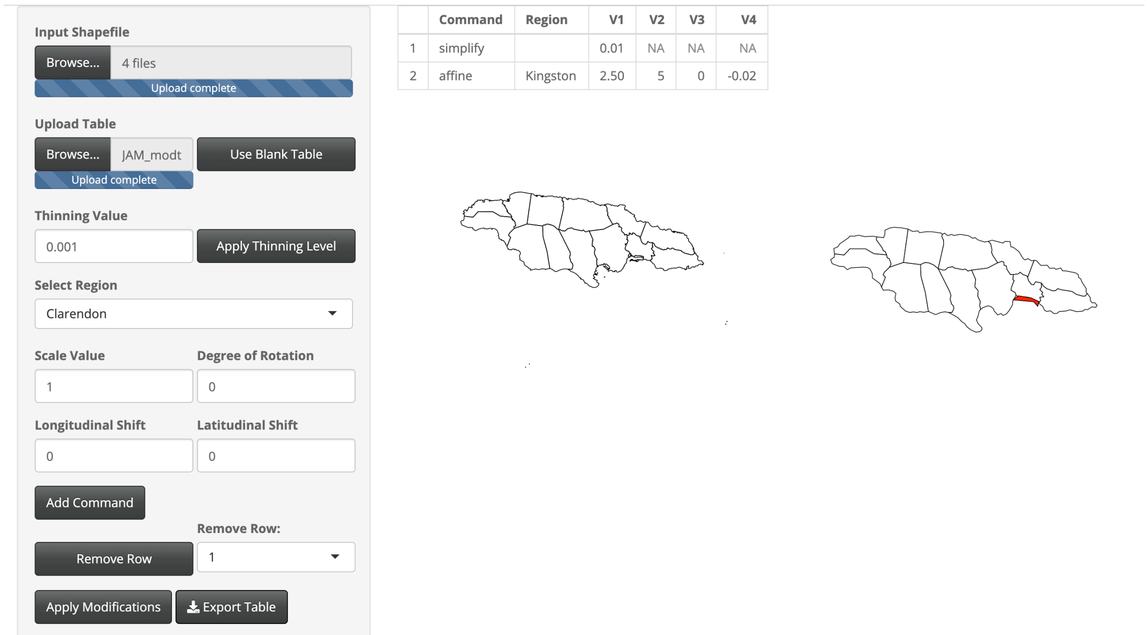
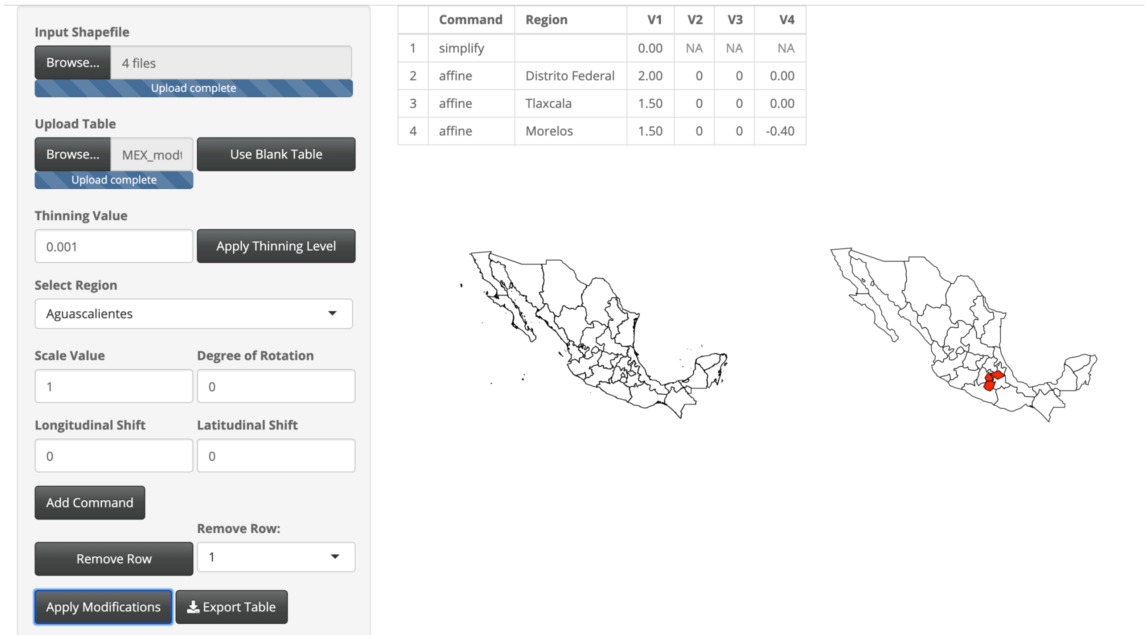


Fig. A.116: Modifications done to Jamaica.

Thinning was done to keep 1% of the original boundary points. The Kingston region was scaled larger with a scale factor of 2.5, rotated 5 degrees clockwise, and shifted south 0.02 degrees latitude.

Mexico



	Command	Region	V1	V2	V3	V4
1	simplify		0.00	NA	NA	NA
2	affine	Distrito Federal	2.00	0	0	0.00
3	affine	Tlaxcala	1.50	0	0	0.00
4	affine	Morelos	1.50	0	0	-0.40

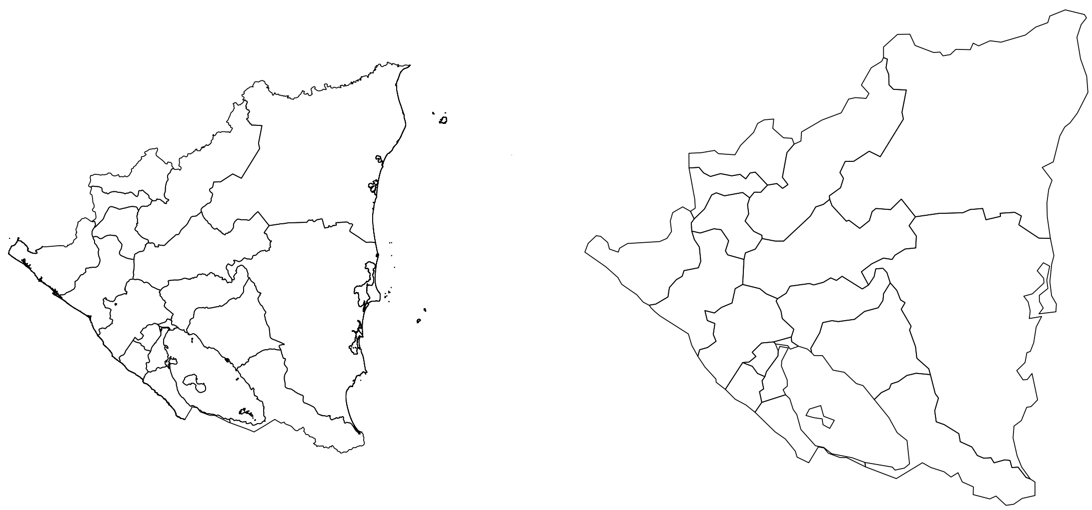
Fig. A.117: Modifications done to Mexico.

Thinning was done to keep 0.01% of the original boundary points. The Distrito Federal region was scaled larger with a scale factor of 2. The Tlaxcala region was scaled larger with a scale factor of 1.5. The Morelos region was scaled larger with a scale factor of 1.5 and shifted south 0.4 degrees latitude.

Nicaragua

```
library(rgdal)
library(rgeos)
library(geojsonio)
library(rmapshaper)
library(sp)
library(dplyr)

shape <- readOGR(dsn = "gadm36_NIC_shp", layer = "gadm36_NIC_1", verbose = TRUE)
NIC_json <- geojson_json(shape, geometry = "polygon", group = "group")

# Unmodifed Shapefile
NIC_sp <- geojson_sp(NIC_json)

# Modified Shapefile
NIC_simplified <- NIC_sp %>%
  ms_simplify(keep = .006)

NIC_simplified <- NIC_simplified[-10, ]
```

Thinning was done to keep 0.6% of the original boundary points. The Lago Nicaragua region was removed from the data.



(a) (b)

Fig. A.118: Nicaragua shown (a) with no modifications and (b) with simplification to boundaries. Additionally, an extraneous region was removed from the GADM shapefile.

Panama

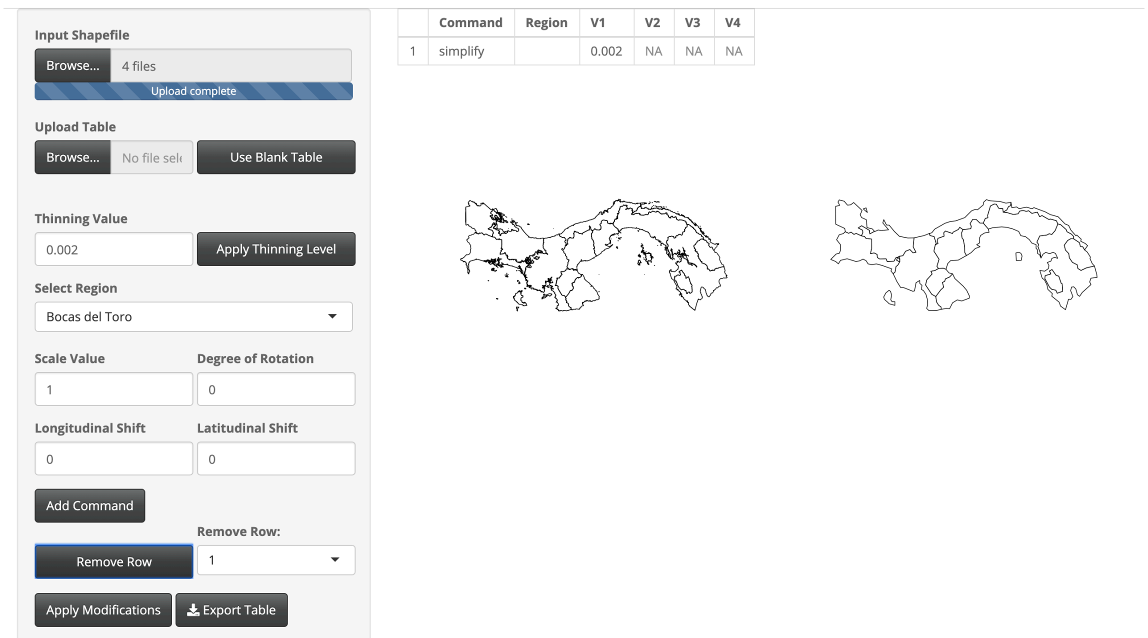


Fig. A.119: Modifications done to Panama.

Thinning was done to keep 0.2% of the original boundary points.

Saint Kitts and Nevis

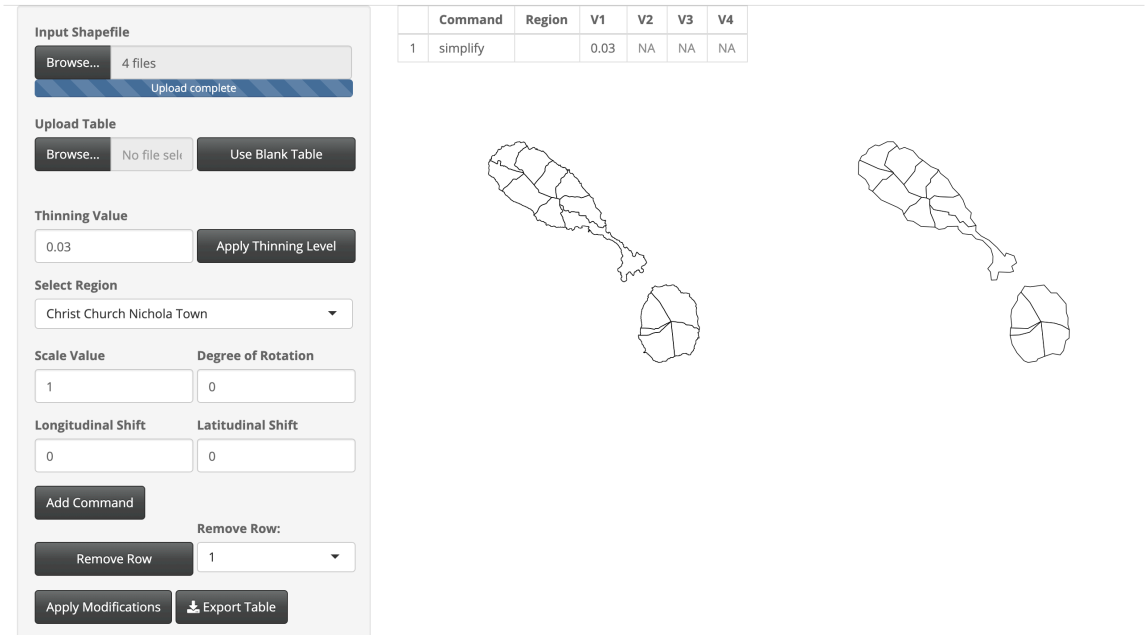


Fig. A.120: Modifications done to Saint Kitts and Nevis.

Thinning was done to keep 3% of the original boundary points.

Saint Lucia

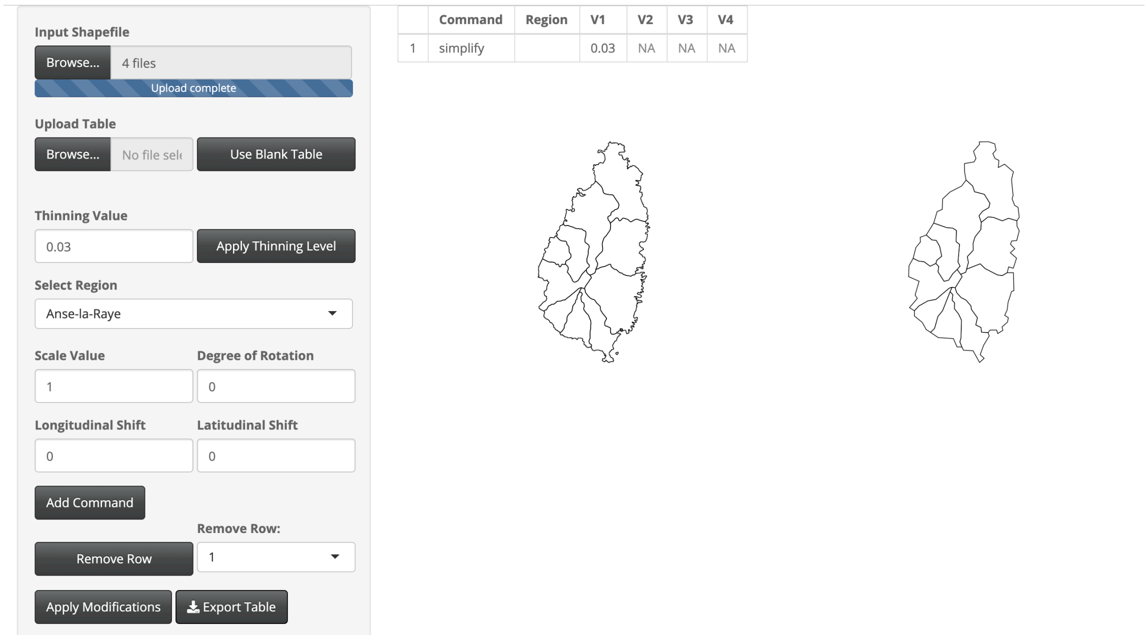


Fig. A.121: Modifications done to Saint Lucia.

Thinning was done to keep 3% of the original boundary points.

Saint Vincent and the Grenadines

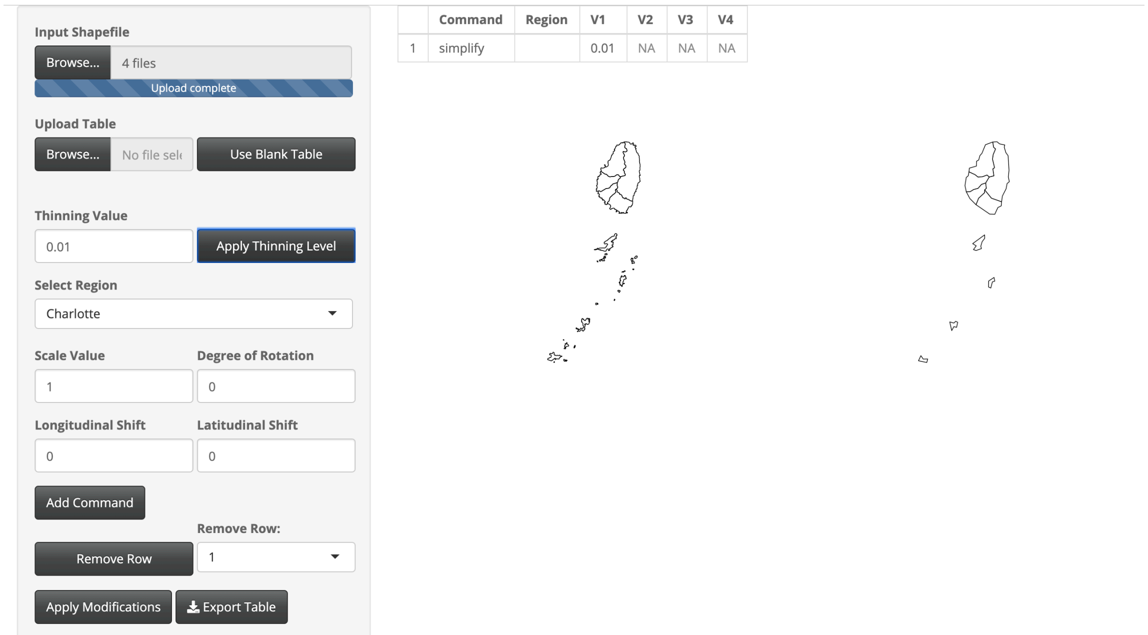


Fig. A.122: Modifications done to Saint Vincent and the Grenadines.

Thinning was done to keep 1% of the original boundary points.

Trinidad and Tobago

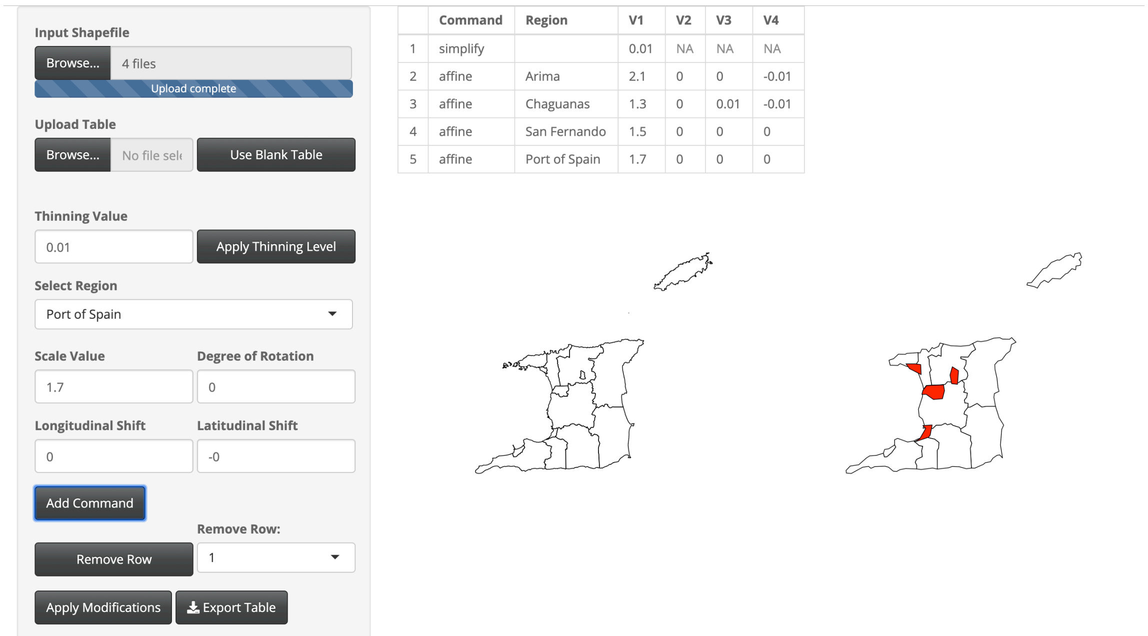


Fig. A.123: Modifications done to Trinidad and Tobago.

Thinning was done to keep 1% of the original boundary points. The Arima region was scaled larger with a scale factor of 2.1. The Chaguanas region was scaled larger with a scale factor of 1.3, shifted east 0.01 degrees longitude, and shifted south 0.01 degrees latitude. The San Fernando region was scaled larger with a scale factor of 1.5. The Port of Spain regions was scaled larger with a scale factor of 1.7.

A.5 Oceania

Table A.5: Table listing Oceanic countries modified and included in the 'LMshapemaker' R package. Countries indicated with a * were created at the command level outside of the 'LMshapemaker' Shiny app.

Country	Page Number
Nauru	196
New Zealand*	197
Papua New Guinea	199

The following countries have been excluded from the 'LMshapemaker' R package for the reasons indicated.

- The Republic of Kiribati, the Marshall Islands, the Republic of Palau, Samoa, the Solomon Islands, Tuvalu, and the Republic of Vanuatu were excluded for consisiting of complex archipelagic geography.
- The Federated States of Micronesia, the Republic of Fiji, and the Kingdom of Tonga were excluded for having five or less first level administrative districts.
- Australia was excluded due to the complex structure of its administrative districts, with regard to geography and the GADM shapefile contents.

Nauru

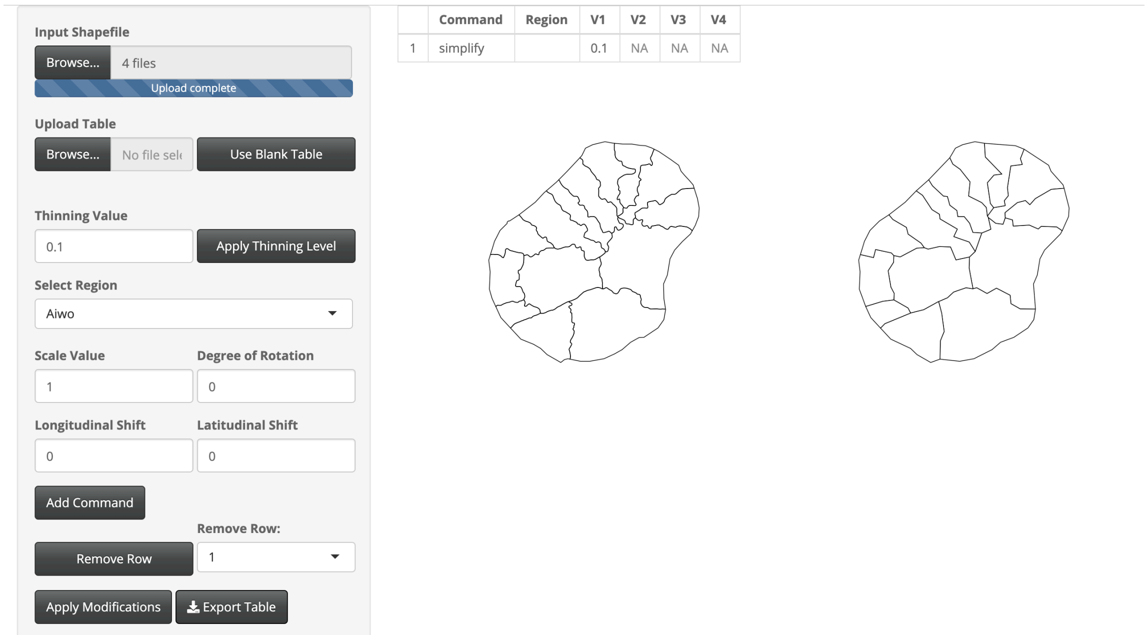
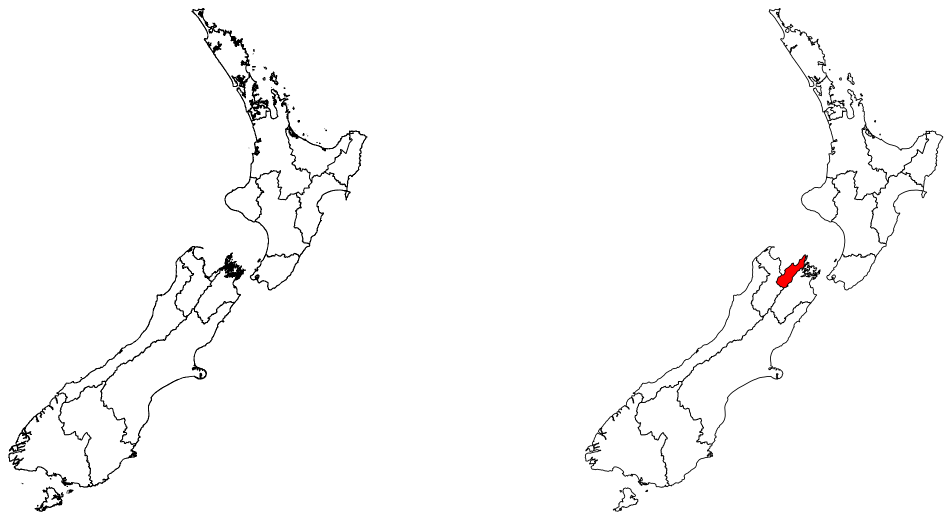


Fig. A.124: Modifications done to Nauru.

Thinning was done to keep 10% of the original boundary points.

New Zealand

```
library(rgdal)
library(rgeos)
library(geojsonio)
library(rmapshaper)
library(sp)
library(dplyr)

shape <- readOGR(dsn = "gadm36_NZL_shp", layer = "gadm36_NZL_1", verbose = TRUE)
NZL_json <- geojson_json(shape, geometry = "polygon", group = "group")

# Unmodifed Shapefile
NZL_sp <- geojson_sp(NZL_json)

# Remove the Chatham Islands (4), Northern Islands (10), and Southern Islands(13)
    polygons.
NZL_sp <- NZL_sp[-c(4, 10, 13), ]

# Modified Shapefile
NZL_simplified <- NZL_sp %>%
  ms_simplify(keep = .003) %>% geojson_json() %>%
  apply_mapshaper_commands("-affine scale=2.5 shift=0,0.09 where='NAME_1 == \"Nelson\
      "'",
                           force_FC = TRUE) %>%
  geojson_sp()
```



(a) (b)

Fig. A.125: New Zealand shown (a) with no modifications and (b) with simplification to boundaries and modifications to the regions (modified regions colored in red).

Thinning was done to keep 0.3% of the original boundary points. The Chatham Islands, Northern Islands, and Southern Islands regions were removed from the shapefile. The Nelson region was scaled larger witha scale factor or 2.5.

Papua New Guinea

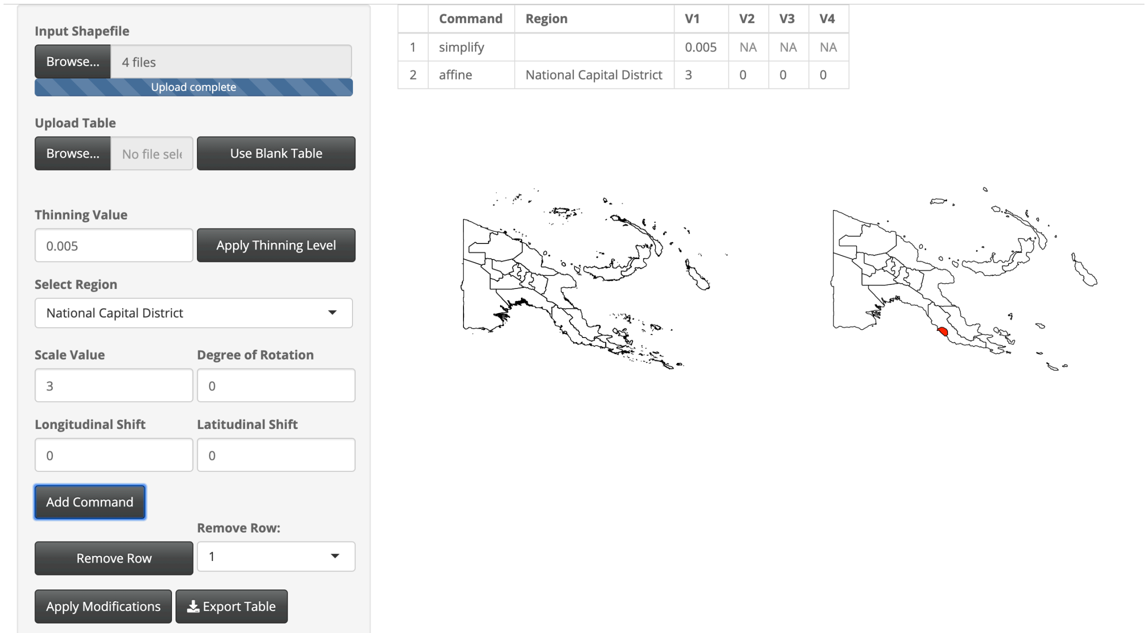


Fig. A.126: Modifications done to Papua New Guinea.

Thinning was done to keep 0.5% of the original boundary points. The National Capital District region was scaled larger with a scale factor of 3.

A.6 South America

Table A.6: Table listing South American countries modified and included in the 'LMshapemaker' R package. Countries indicated with a * were created at the command level outside of the 'LMshapemaker' Shiny app.

Country	Page Number
Argentina	201
Bolivia	202
Brazil	203
Colombia*	204
Ecuador	206
Guyana	207
Paraguay	208
Peru	209
Suriname	210
Uruguay*	211
Venezuela	213

The following country has been excluded from the 'LMshapemaker' R package for the reasons indicated.

- Chile for consisting of complex geography due to the north-south orientation of the country.

Argentina

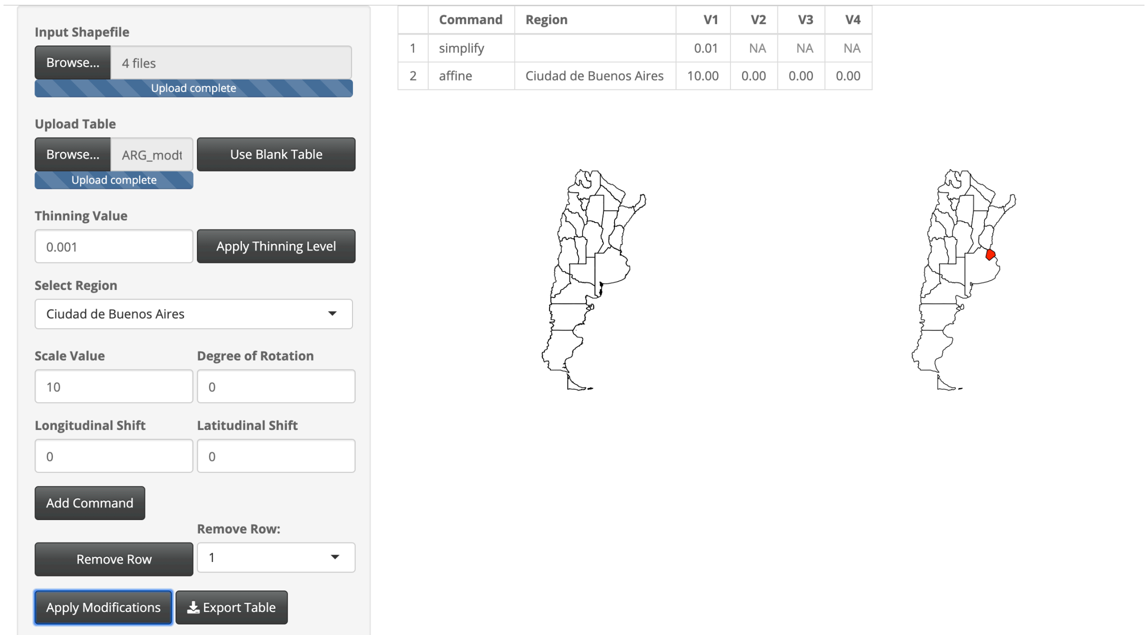


Fig. A.127: Modifications done to Argentina.

Thinning was done to keep 0.05% of the original boundary points. The Ciudad de Buenos Aires region was scaled larger with a scale factor of 10.

Bolivia

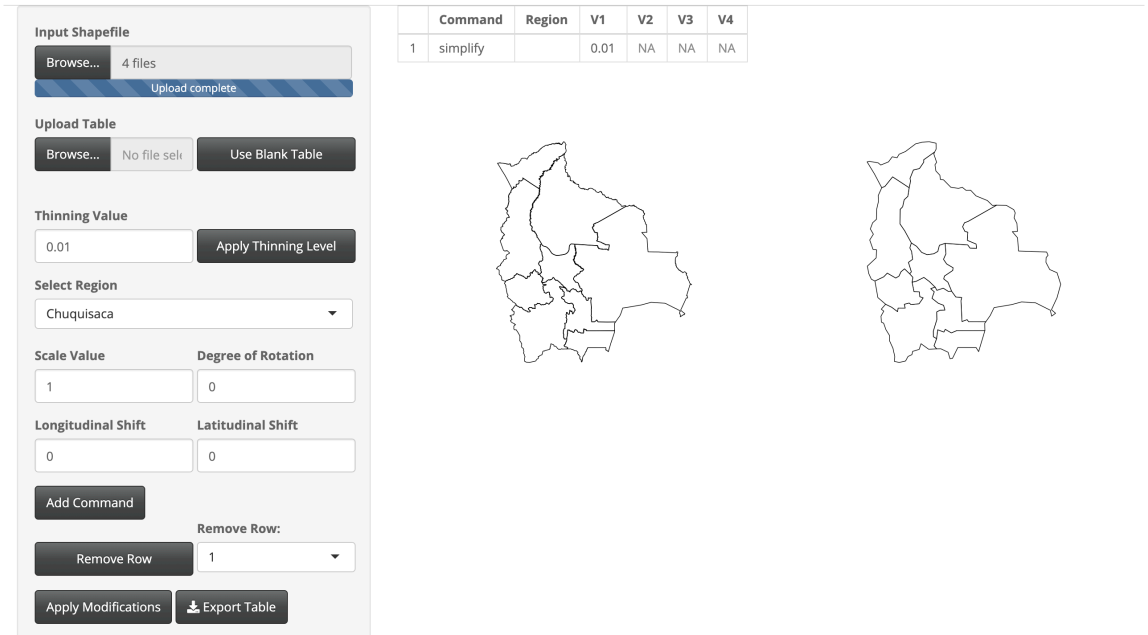


Fig. A.128: Modifications done to Bolivia.

Thinning was done to keep 1% of the original boundary points.

Brazil

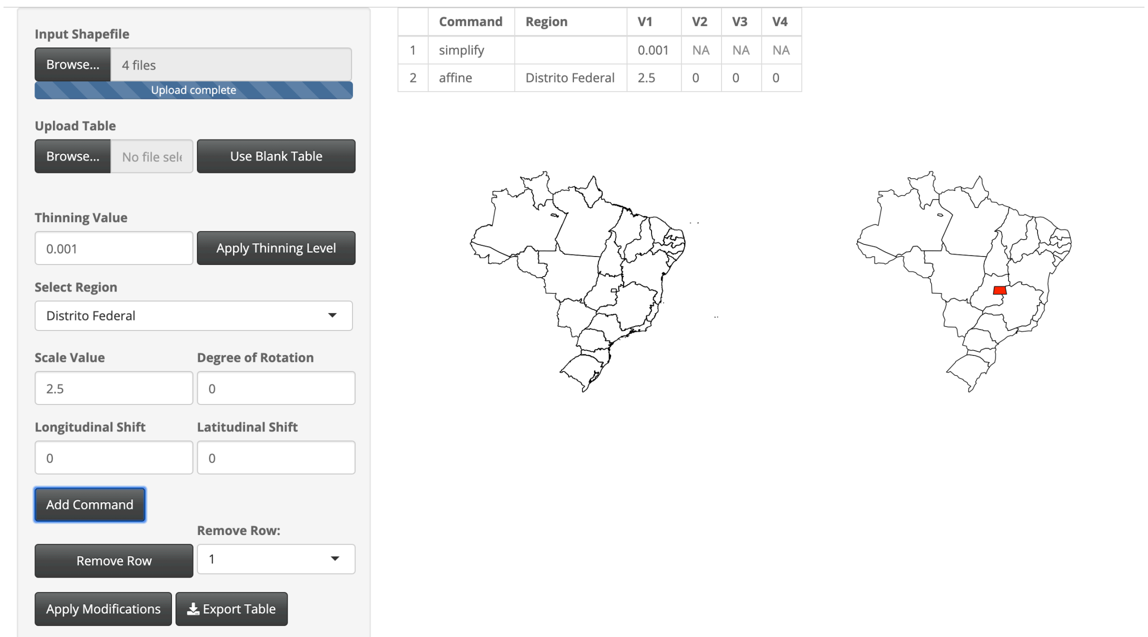


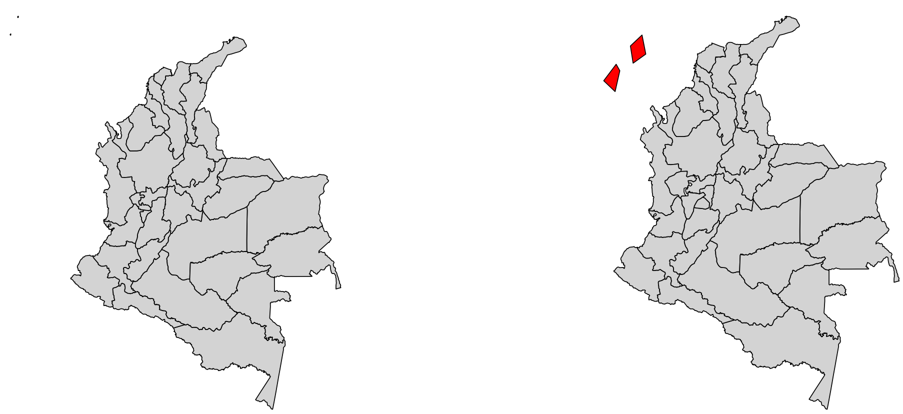
Fig. A.129: Modifications done to Brazil.

Thinning was done to keep 0.1% of the original boundary points. The Distrito Federal region was scaled larger with a scale factor of 2.5.

Colombia

```
library(rgdal)
library(rgeos)
library(geojsonio)
library(rmapshaper)
library(sp)
library(dplyr)

shape <- readOGR(dsn = "gadm36_COL_shp", layer = "gadm36_COL_1", verbose = TRUE)
COL_json <- geojson_json(shape, geometry = "polygon", group = "group")
# Unmodified Shapefile
COL_sp <- geojson_sp(COL_json)

# Modified Shapefile
COL_simplified <- COL_sp %>%
  ms_simplify(keep = .02) %>%
  geojson_json() %>%
  apply_mapshaper_commands("-affine scale=1.5 shift=0,0 where='NAME_1 == \"Atlantico\
      "'
                            -affine scale=18 shift=2.5,-5 where='NAME_1 == \"San
                                Andres y Providencia\"'
                            -affine scale=2 shift=0,0 where='NAME_1 == \"Risaralda\"'
                            -affine scale=2 shift=0,0 where='NAME_1 == \"Quindio\"'",
                           force_FC = TRUE) %>%
  geojson_sp()
# Break up the polygon consisting of the two islands San Andres and Providencia
d <- disaggregate(COL_simplified[26, ])
COL_simplified <- rbind(d, COL_simplified[-26, ])
COL_simplified$GID_1[2] <- "COL.26_2"
# Move the islands separately
COL_simplified_2 <- geojson_json(COL_simplified)
COL_simplified_2 <- COL_simplified_2 %>%
  apply_mapshaper_commands("-affine scale=1 shift=3,9 where='GID_1 == \"COL.26_1\"'
                            -affine scale=1 shift=-2,-4 where='GID_1 == \"COL.26_2\"'
                                ",
                           force_FC = TRUE) %>%
  geojson_sp()
```

Thinning was done to keep 2% of the original boundary points. The Atlántico region was scaled larger with a scale factor of 1.5. The Risaralda region was scaled larger with a scale factor of 2. The Quindío region was scaled larger with a scale factor of 2. The San Andrés y Providencia regions are combined in the same administrative district. They were enlarged by a scale factor of 18, shifted east 2.5 degrees longitude, and shifted south 5 degrees latitude. Using the `disaggregate()` function, further modifications were made to the islands independently. The island of San Andrés was shifted an additional 2 degrees east and 4 degrees south.



(a) (b)

Fig. A.130: Colombia shown (a) with no modification and (b) with simplification to boundaries and modifications to the regions (modified regions colored in red).

Ecuador

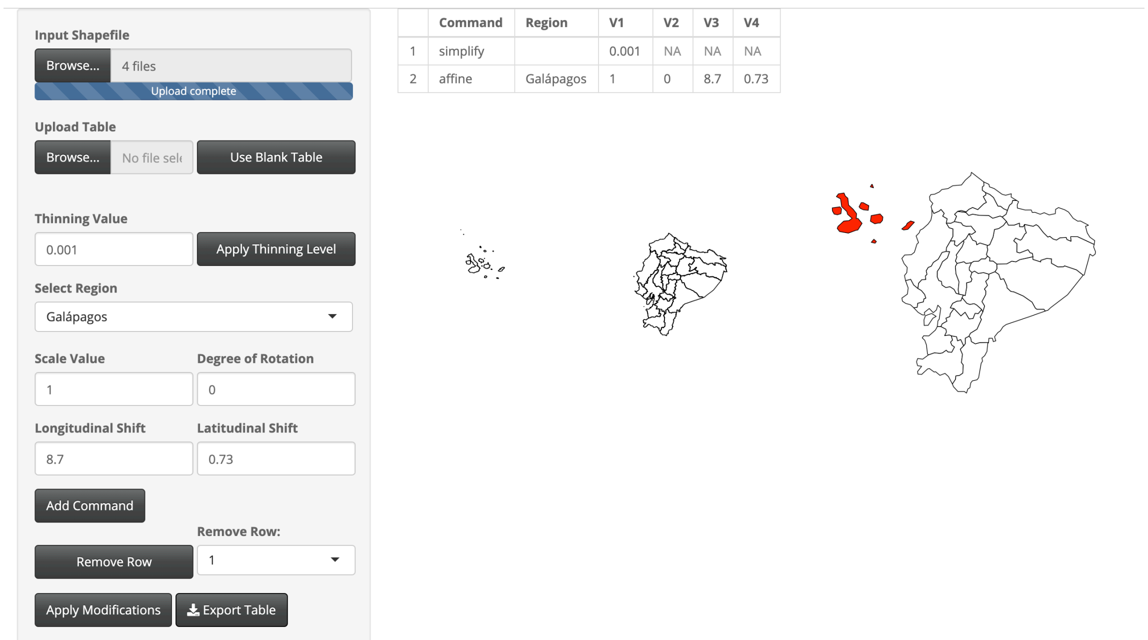


Fig. A.131: Modifications done to Ecuador.

Thinning was done to keep 0.1% of the original boundary points. The Galápagos region was shifted east 8.7 degrees longitude and north 0.73 degrees latitude.

Guyana

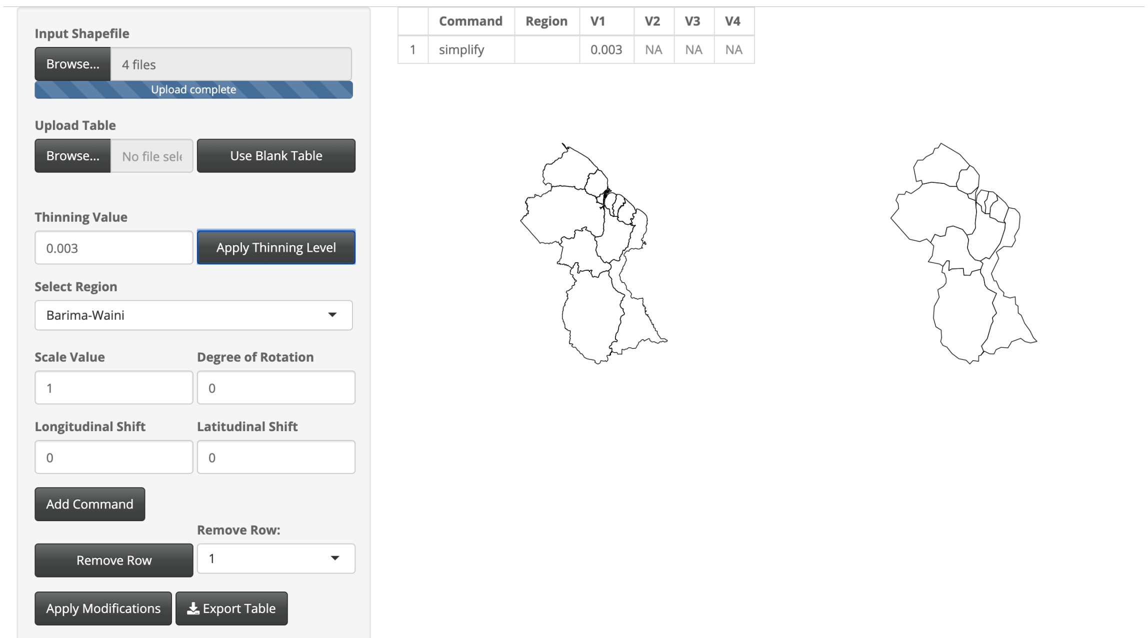


Fig. A.132: Modifications done to Guyana.

Thinning was done to keep 0.3% of the original boundary points.

Paraguay

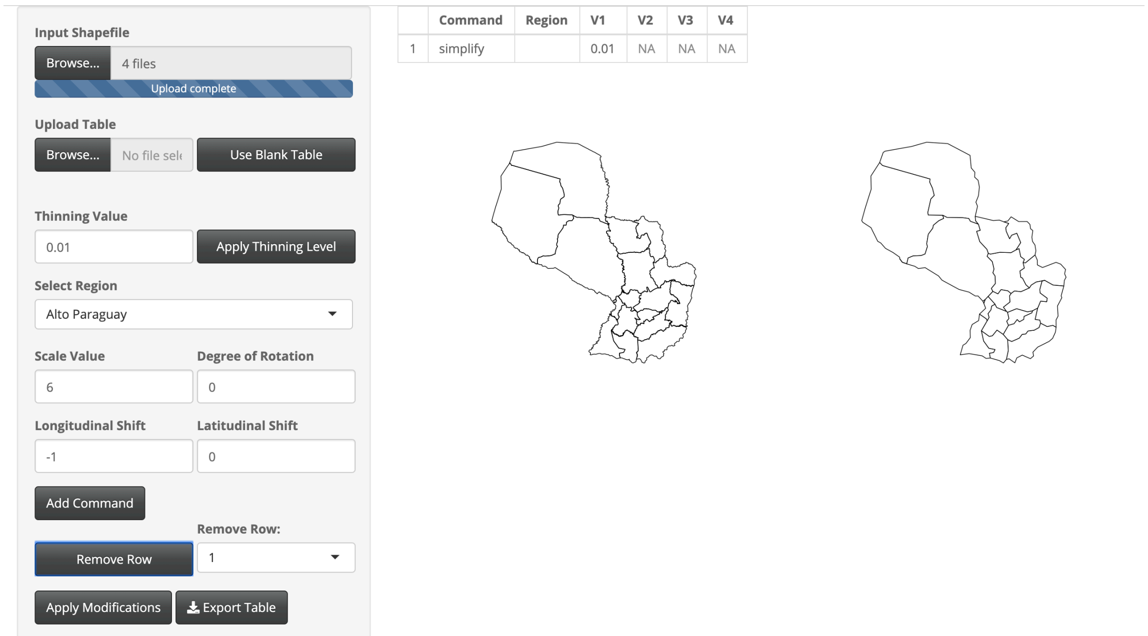
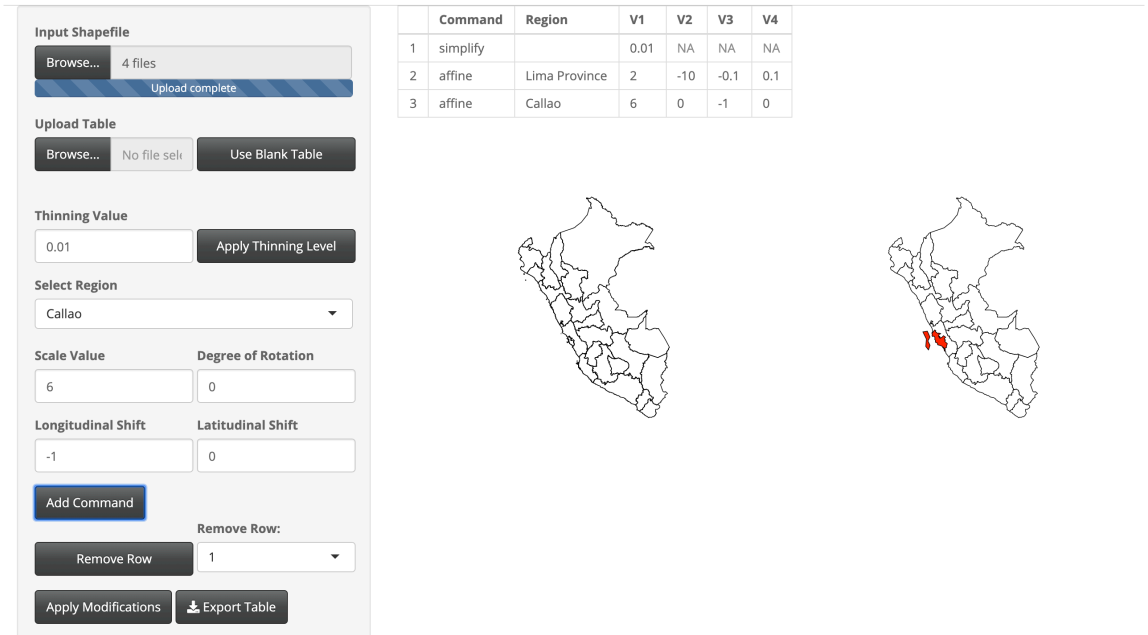


Fig. A.133: Modifications done to Paraguay.

Thinning was done to keep 1% of the original boundary points.

Peru



	Command	Region	V1	V2	V3	V4
1	simplify		0.01	NA	NA	NA
2	affine	Lima Province	2	-10	-0.1	0.1
3	affine	Callao	6	0	-1	0

Fig. A.134: Modifications done to Peru.

Thinning was done to keep 1% of the original boundary points. The Lima Province region was scaled larger with a scale factor of 2, rotated 10 degree counter-clockwise, and shifted west 0.1 degrees longitude and north 0.1 degrees latitude. The Callao region was scaled larger with a scale factor of 6 and shifted west 1 degree longitude.

Suriname

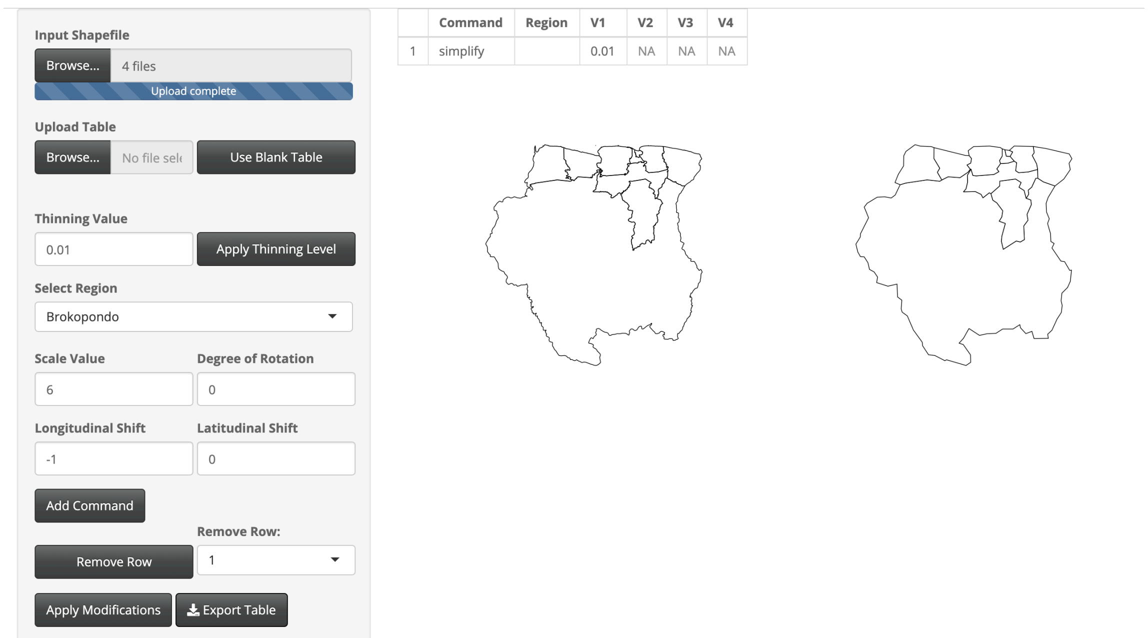


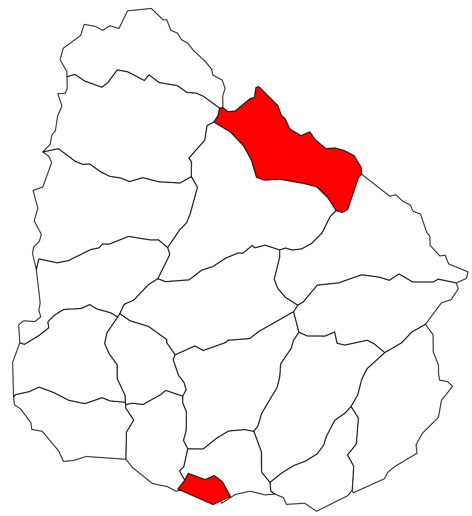
Fig. A.135: Modifications done to Suriname.

Thinning was done to keep 1% of the original boundary points.

Uruguay

```
library(rgdal)
library(rgeos)
library(geojsonio)
library(rmapshaper)
library(sp)
library(dplyr)

shape <- readOGR(dsn = "gadm36_URY_shp", layer = "gadm36_URY_1", verbose = TRUE)
URY_json <- geojson_json(shape, geometry = "polygon", group = "group")

# Unmodifed Shapefile
URY_sp <- geojson_sp(URY_json)

# Modified Shapefile
URY_simplified <- URY_sp %>%
  ms_simplify(keep = .01) %>%geojson_json() %>%
  apply_mapshaper_commands("-affine scale=1.5 shift=0,0.09 where='NAME_1 == \"
      Montevideo\"'",
                           force_FC = TRUE) %>%
  geojson_sp()

# Remove the extraneous polygon on region 13 (Rivera)
URY_simplified@polygons[[13]]@Polygons[[1]] <- NULL
```

(a)



(b)

Fig. A.136: Uruguay shown (a) with no modifications and (b) with simplification to boundaries and modifications to the regions (modified regions colored in red).

Thinning was done to keep 1% of the original boundary points. The Montevideo region was scaled larger with a scale factor of 1.5 and shifted north 0.09 degrees latitude. An extraneous polygon on the eastern border was removed from the Rivera region.

Venezuela

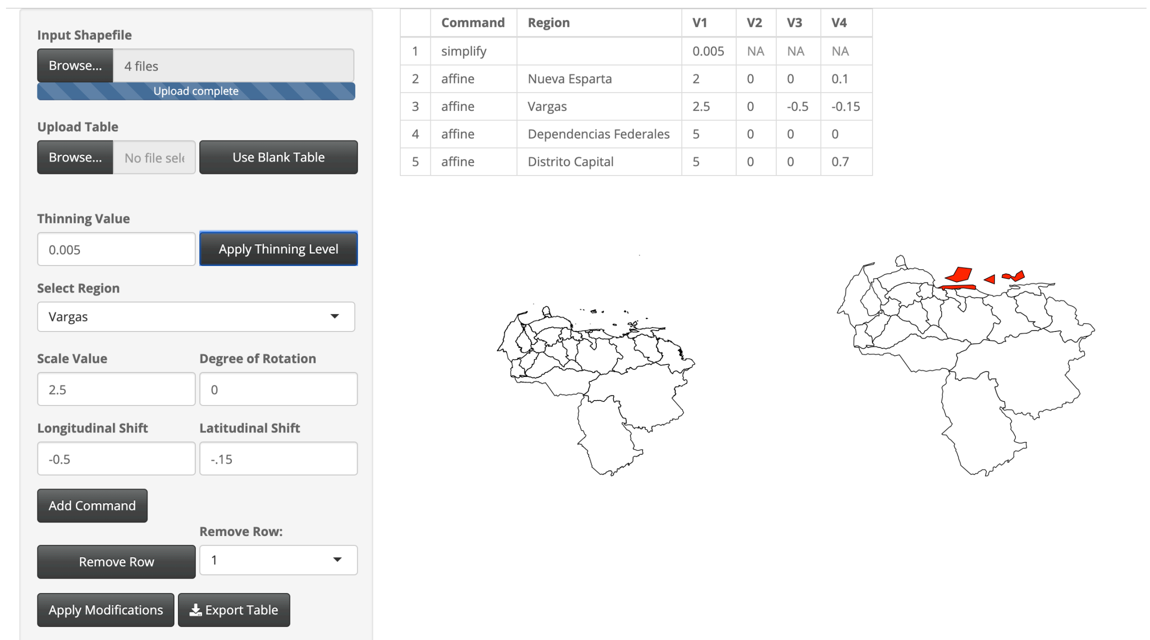


Fig. A.137: Modifications done to Venezuela.

Thinning was done to keep 0.5% of the original boundary points. The Nueva Esparta region was scaled larger with a scale factor of 2 and shifted north 0.1 degrees latitude. The Vargas region was scaled larger with a scale factor of 2.5 and shifted west 0.5 degrees longitude and south 0.15 degrees latitude. The Dependencias Federales region was scaled larger with a scale factor of 5. The Distrito Capital region was scaled larger with a scale factor of 5 and shifted north 0.7 degrees latitude.

APPENDIX B

'LMshapemaker' Documentation

Included in Appendix B is the R documentation for the 'LMshapemaker' R package. This includes the help pages for the functions, an example of the the data call functions for countries included in the R package, and the vignette as an extended guide for the R package.

B.1 Help Pages

Package ‘LMshapemaker’

December 2, 2019

Title Prepare Shapefiles for Use in Linked Micromap Plots

Version 0.0.0.9000

Description LMshapemaker provides ready-to-use shapefiles for use in linked micromap (LM) plots. Through an included Shiny app, users can modify GADM shapefiles in real time to produce a modification table for the shape file in question. With the mod_shape() function, a user can apply these modifications to a shapefile for use in spatial visualizations in R.

Depends R (>= 3.5.0)

License BSD

Encoding UTF-8

LazyData false

Imports dplyr, geojsonio, rgdal, rmapshaper, shiny, shinythemes, sp

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Braden Probst [aut, cre]

Maintainer Braden Probst <braden.probst@gmail.com>

R topics documented:

`mexico`	*Modified Mexico Shapefile*

Description

A list object including a modified shapefile for creating a LM plot using Mexico. Additionally included is a table, detailing the specific modifications used in creating this shapefile.

Usage

```
data("mexico")
```

Format

An object of class ‘"list"‘. [[1]] is the modification table. [[2]] is the modifed shapefile.

`mod_shape`	*Use Modification Table to Create Modified Shapefiles*

Description

mod_shape allows modifications specified in a table object to be applied to a given shapefile.

Usage

```
mod_shape(shape, mod_table)
```

Arguments

`shape`	A GADM shapefile.
`mod_table`	A table of modifications as provided by the ’LMshapemaker’ Shiny app.

Details

mod_table needs to be formatted in the same way that the tables provided by the ShinyShapemaker Shiny app provide in this R package. The columns need to be Command, Region, V1, V2, V3, and V4. The shapefile needs to be from GADM and specified to be the first administrative level.

Value

A shapefile with the specified modifications in the modification table.

`runShinyShapemaker` *'LMshapemaker' Shiny App*

Description

Launch the 'LMshapemaker' Shiny app in a web browser to customize GADM shapefiles and export modification tables. Use mod_shape to apply modifications to a shapefile in R.

Usage

```
runShinyShapemaker()
```

`vignette_data` *Mexico Murder Rate Data for Vignette*

Description

Murder rate data for Mexico from the years 2010 and 2017.

Usage

```
data("vignette_data")
```

Format

A 32 by 3 data frame. Columns are [1] states of Mexico, [2] murder rate in 2010, and [3] murder rate in 2017.

Index

B.2 Vignette

‘LMshapemaker’ Basics

Introduction to the ‘LMshapemaker’ R Package

The purposes of the ‘LMshapemaker’ R package is to use the ‘rmapshaper’ functionality to provide access to ready-to-use shapefiles for creating linked micromap (LM) plots and to allow users to customize shapefiles for their uses. The shapefiles provided for over 130 countries in the world were created with an objective perspective to ensure each region could be colored and meaningfully represented in the context of LM plots. As such, the shapefiles in this package are not 1:1 representations of any countries existing borders. In every case, the boundary points were simplified to allow for quick plotting. In many more cases, entire administrative districts were resized and/or shifted in the latitude and longitude directions. For shapefiles that do not adequately represent the users’ needs, the `runShinyShapemaker` function allows users to modify GADM shapefiles to their specifications.

At this time, `LMshapemaker` provides the following functions:

`mod_shape` - apply modifications from a ShinyShapemaker modification table to a GADM shapefile

`runShinyShapemaker` - launch a browser based Shiny R app for modifying shapefiles in real time. Can export modification table to use with the `mod_shape` function to recreate the shapefile in R.

`data` - Can be used to obtain a list object containing the provided modifications in table format for a given country and the ready-to-use modified shapefiles for creating LM plots.

This short vignette focuses on how to make changes to provided shapefiles and use the final modified product to create a useable LM plot.

Usage

```
library(LMshapemaker)
```

```
## Registered S3 method overwritten by 'crul':
##   method                 from
##   as.character.form_file httr
```

```
## Registered S3 method overwritten by 'geojsonlint':
##   method         from
##   print.location geojsonio
```

```
library(rmapshaper)
library(tidyverse)
```

```
## Registered S3 method overwritten by 'dplyr':
##   method         from
##   print.location geojsonlint
```

```
## ── Attaching packages
──────────────────────────────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.2.0     ✔ purrr   0.3.2
## ✔ tibble  2.1.3     ✔ dplyr   0.8.3
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0
```

```
## ── Conflicts ───────────────────────────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(raster)
```

```
## Loading required package: sp
```

```
##
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

Load the provided data and modified shapefile for Mexico.

```
data("mexico")
```

Let's examine the two objects returned with the `data` call.

```
class(mexico)
```

```
## [1] "list"
```

```r
class(mexico[[1]])
```

```
## [1] "data.frame"
```

```r
head(mexico[[1]])
```

```
##    Command           Region    V1 V2 V3   V4
## 1 simplify                  0.001 NA NA   NA
## 2   affine Distrito Federal 2.000  0  0  0.0
## 3   affine         Tlaxcala 1.500  0  0  0.0
## 4   affine          Morelos 1.500  0  0 -0.4
```

```r
class(mexico[[2]])
```

```
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

```r
plot(mexico[[2]])
```

These look pretty good. The modification table specified that there were four modifications done to the boundaries and regions of mexico. The `simplify` line works with the `ms_simplify` function from `rmapshaper`. This applies the Douglas-Peuker algorithm to remove a proportion of total points (V1) used to make up the boundaries in order to make the boundaries. This process is known as thinning. The next three rows indicate that three regions were modified in the shapefile, specifically the Distrito Federal, Tlaxcala, and Morelos regions. V1, V2, V3, and V4 refer do different parameters when paired with the `affine` value in the Command column. V1 is the scaling value to resize the region. In the case of the Distrito Federal region, it is being scaled larger by a factor of 2. V2 would be the value in degrees that you rotate a region. V3 and V4 represent shifts to the longitude and latitude coordiniates. In this example we have shifted the Morelos regions south by 0.4 degrees latitude. Lets see the three regions in question.

```
MEX_mod_table <- mexico[[1]]
MEX_shp <- mexico[[2]]
MEX_shp@data$color  <- ifelse(MEX_shp$NAME_1 %in% MEX_mod_table$Region,
                        "red", NA)
plot(MEX_shp, col = MEX_shp@data$color)
```

The three highlighted regions were originally too small to see when plotted in the context of a LM plot. A user could go forward with this shapefile to create LM plots for Mexico. While this shapefile was made from an objective perspective to provide each region with enough area to be meaningfully colored, someone more familiar with the country of Mexico may want to represent it differently. In many maps in the United States, the District of Columbia (Washington D.C.) is represented as an island off the coast of Virginia. This is to empahsize a small region that would otherwise not show up. This same approach could be applied to the Distrito Federal of Mexico. To do so, a user would save this shapefile externally and run `runShinyShapmaker`. Upon moving in to the browser for the app, the user would load this shapefile in and specify they are using a new table. The user would select Distrito Federal from the drop down region list and apply different values for longitudinal and latitudinal shifts until it was in the right place. From the app they could export the table. For this vignette, we will build it manually.

```
alt_table <- MEX_mod_table[-c(1, 3, 4), ]
alt_table[1, 3:6] <- c(1, 0, 5, 1)
alt_table
```

```
##   Command           Region V1 V2 V3 V4
## 2  affine Distrito Federal  1  0  5  1
```

Now we can use the `mod_shape` command to apply this new change to the shapefile. Note that we are applying a new modification to the existing modified shapefile. That is, the Morelos and Tlaxcala regions will still be modified. We are just shifting the Distrito Federal region to be represented as an island in the Gulf of Mexico.

```
MEX_shp_new <- mod_shape(shape = MEX_shp, mod_table = alt_table)
plot(MEX_shp_new, col = MEX_shp@data$color)
```

Create a LM Plot

Now that we have a newly modified shapefile, let's use some real data from Wikipedia.org to see how a LM plot would look using this shapefile.

```
data("vignette_data")
```

This data is pulled from the Time Series data from
https://en.wikipedia.org/wiki/List_of_Mexican_states_by_homicides. We have pulled the murder rates for each Mexican state for the years of 2010 and 2017.

```
library(micromap)
```

```
## Loading required package: maptools
```

```
## Checking rgeos availability: TRUE
```

```
## Loading required package: RColorBrewer
```

```
## Loading required package: rgdal
```

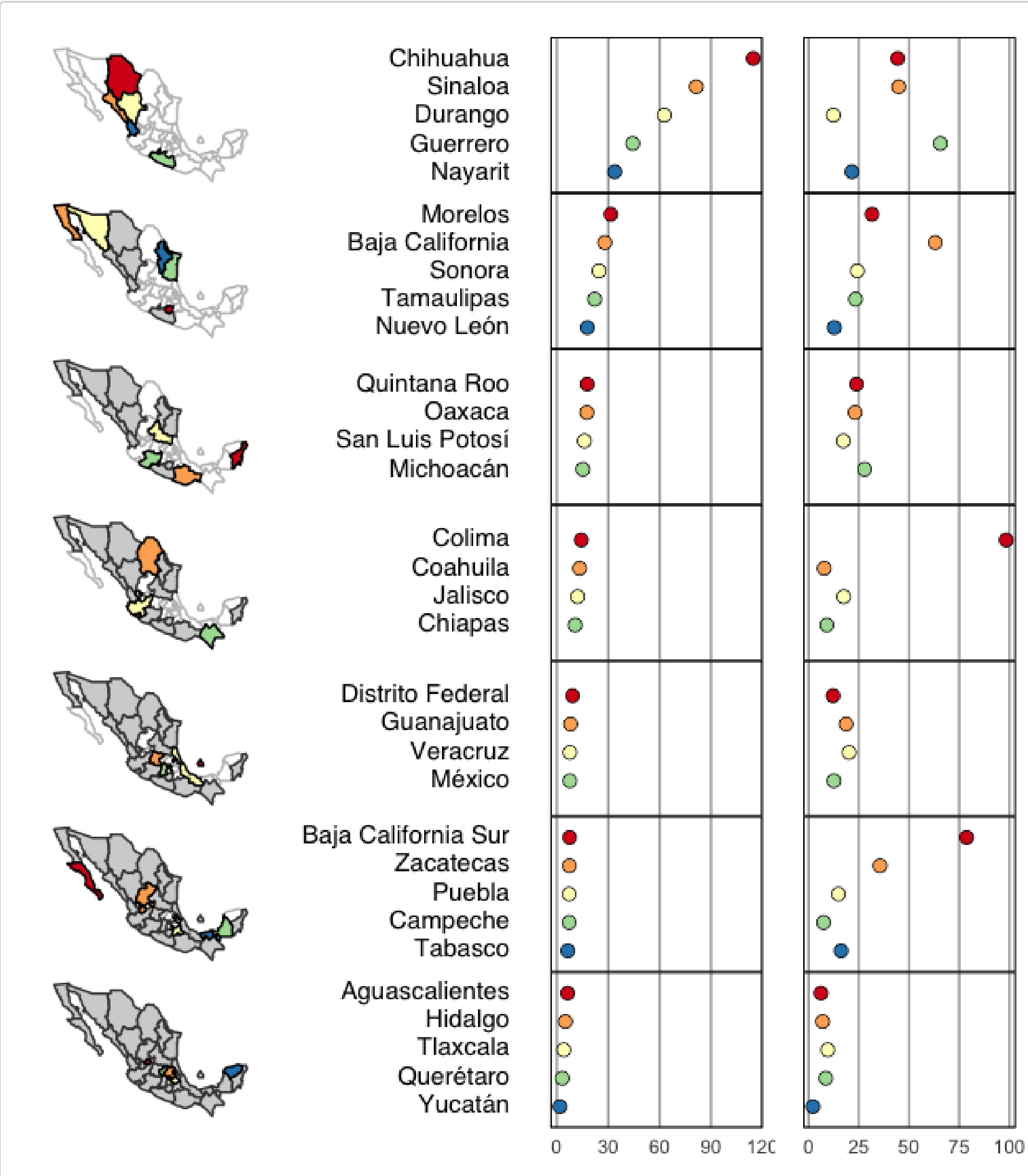
```
## rgdal: version: 1.4-4, (SVN revision 833)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.1.3, released 2017/20/01
##  Path to GDAL shared files:
/Library/Frameworks/R.framework/Versions/3.6/Resources/library/sf/gdal
##  GDAL binary built with GEOS: FALSE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files:
/Library/Frameworks/R.framework/Versions/3.6/Resources/library/sf/proj
##  Linking to sp version: 1.3-1
```

```
Data <- vignette_data
Polys <- create_map_table(MEX_shp_new, "NAME_1")
```

The `micromap` package allows users to easily make LM plots. Below is a basic example. We will further modify it later.

```
BasicPlot <- mmplot(stat.data = Data,
                    map.data = Polys,
                    panel.types = c("map", "labels", "dot", "dot"),
                    panel.data = list(NA, "State",
                                      "MurderRate 2010",
                                      "MurderRate 2017"),
                    ord.by = "MurderRate 2010",
                    median.row = FALSE,
                    rev.ord = TRUE,
                    grouping = c(5, 5, 4, 4, 4, 5, 5),
                    vertical.align = "center",
```

```
                   map.link = c("State", "ID")
                   )
```



Let's create a new variable in our data that shows the change in murder rate.

```
vignette_data <- vignette_data %>%
  mutate(Murderchange = `MurderRate 2017` - `MurderRate 2010`) %>%
  mutate(MurderchangePercent = ((`MurderRate 2017` - `MurderRate
 2010`)/`MurderRate 2010`)*100)
```

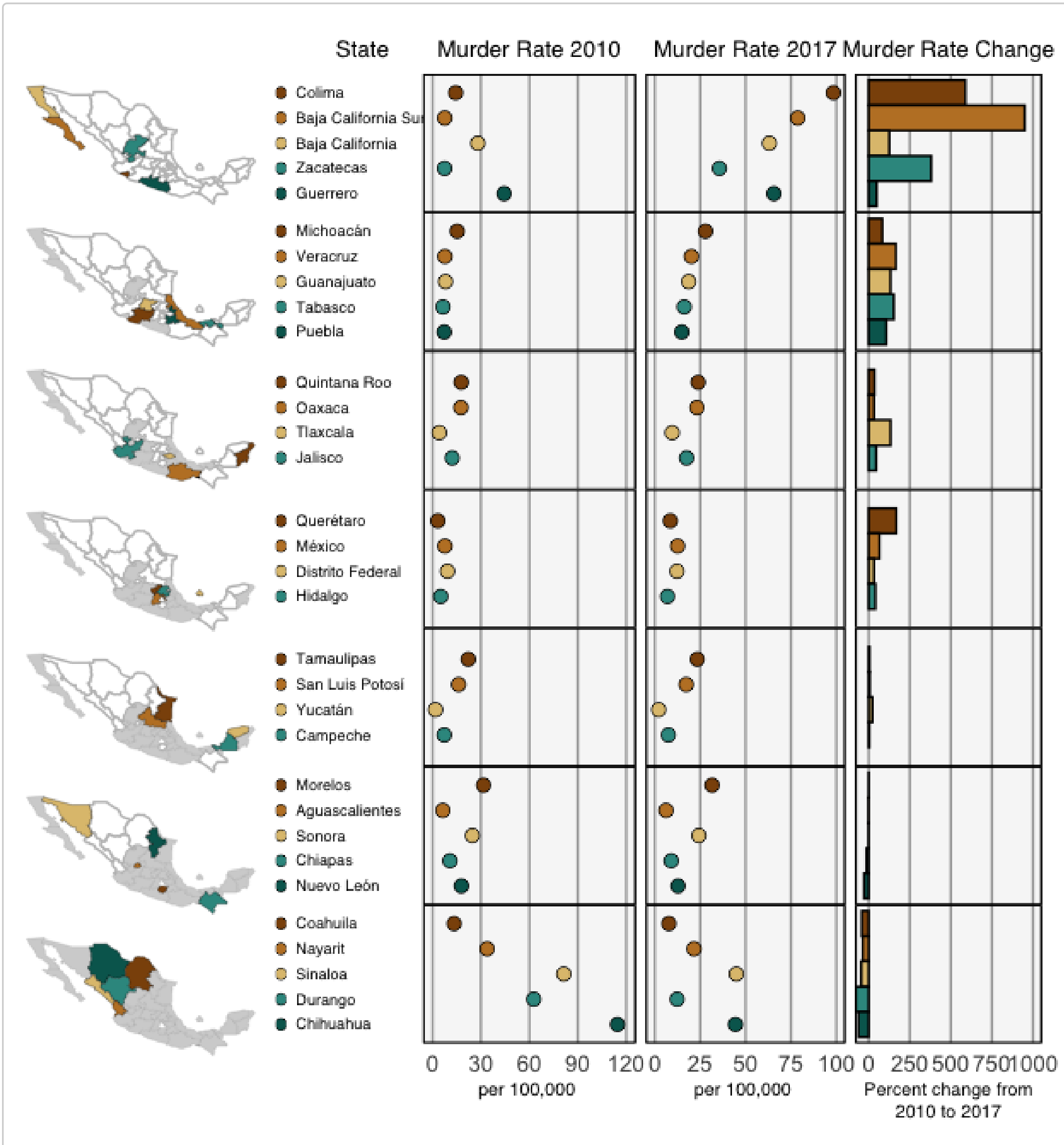
```
Data <- vignette_data

head(vignette_data)
```

```
##                 State MurderRate 2010 MurderRate 2017 Murderchange
## 1      Aguascalientes             6.3             6.2         -0.1
## 2     Baja California            28.0            63.1         35.1
## 3 Baja California Sur             7.5            78.7         71.2
## 4            Campeche             7.2             7.5          0.3
## 5             Chiapas            10.7             9.1         -1.6
## 6           Chihuahua           114.6            44.4        -70.2
##   MurderchangePercent
## 1           -1.587302
## 2          125.357143
## 3          949.333333
## 4            4.166667
## 5          -14.953271
## 6          -61.256545
```

To better understand this LM plot, let's modify it further and add some labels to make it clear what we are seeing.

```
RefinedPlot <- mmplot(stat.data = Data,
                      map.data = Polys,
                      panel.types = c("map", "dot_legend", "labels", "dot",
  "dot", "bar"),
                      panel.data = list(NA, NA, "State",
                                        "MurderRate 2010",
                                        "MurderRate 2017",
                                        "MurderchangePercent"),
                      ord.by = "Murderchange",
                      median.row = FALSE,
                      rev.ord = TRUE,
                      grouping = c(5, 5, 4, 4, 4, 5, 5),
                      vertical.align = "center",
                      map.link = c("State", "ID"),

                      colors = brewer.pal(8, "BrBG")[c(1, 2, 3, 7, 8)],
                      two.ended.maps = FALSE,
                      map.all = TRUE,
                      map.color2 = "lightgray",

                      plot.panel.spacing = 0.70,
                      panel.att = list(
                        list(1,
                             inactive.border.color = gray(0.7),
                             inactive.border.size = .2,
                             nodata.border.size = .5,
```

```
                        active.border.size = .2,
                        active.border.color = 'black',
                        panel.width = .75,
                        left.margin = -1),
                   list(2, point.type = 20,
                        point.size = 1.2,
                        panel.width = .65,
                        left.margin = -1, right.margin = 1),
                   list(3, header = "State",
                        panel.width = 0.3,
                        align = "left",
                        text.size = 0.6,
                        left.margin = -2, right.margin = .05),
                   list(4, header = "Murder Rate 2010",
                        panel.width = .6,
                        panel.header.size = 1,
                        graph.bgcolor=(brewer.pal(7,"Greys"))[1],
                        left.margin = -1.5, right.margin = .15,
                        connected.dots = FALSE,
                        xaxis.ticks = c(0, 30, 60, 90, 120), #seq(.65, .95,
 .10)
                        xaxis.labels = c(0, 30, 60, 90, 120),
                        xaxis.title = "per 100,000"
                   ),
                   list(5, header = "Murder Rate 2017",
                        panel.width = .6,
                        panel.header.size = 1,
                        graph.bgcolor=(brewer.pal(7,"Greys"))[1],
                        connected.dots = FALSE,
                        left.margin = -1.1, right.margin = .15,
                        xaxis.ticks = seq(0, 100, by = 25),
                        xaxis.labels = seq(0, 100, by = 25),
                        xaxis.title = "per 100,000"
                   ),
                   list(6, header = "Murder Rate Change",
                        panel.width = .6,
                        panel.header.size = 1,
                        graph.bgcolor=(brewer.pal(7,"Greys"))[1],
                        left.margin = -1.1, right.margin = .55,
                        xaxis.ticks = seq(0, 1000, by = 250),
                        xaxis.labels = seq(0, 1000, by = 250),
                        xaxis.title = "Percent change from\n2010 to 2017"
                   )
                  )
)
```



The picture is a little more clear now. We can see that some of the safest states in 2010 had huge spikes (up to over 500%) in murder rates in 2017. The opposite side of that is that the more dangerous states saw decreases in murder rate.