Regionalization preprocessing procedures for spatial interaction modeling

Tyler Hoffman

December 16, 2020

1 Introduction

Human mobility is an important, highly dynamic factor to consider in urban planning problems. Activities that require travel through physical space – like commuting, dining, and tourism – are forces that drive a city [2]. Spatial interaction (SI) models, commonly used by academics and urban planners alike, provide a framework for analyzing the varied dimensions of human movement. Among other applications, these models have been used to optimize cities, raising the natural question of what makes a city optimal. The COVID-19 pandemic, for example, has revealed that previous notions of optimality required citizens to be too close together and brought about rapid disease transmission.

With new sources of spatiotemporal data, we are now able to leverage insights from urban activity data with a higher temporal volume and spatial coverage than ever before. Yet such a fine resolution and detail often comes at the cost of computational complexity and fitness for use in urban models. One option that balances these concerns is to aggregate these individual level data in a way that ensures anonymity, maximizes data quality, and maintains tractability for real-time decision-making. This aggregation problem is known by many names (such as region-building or spatial clustering) and algorithms that solve it are called regionalization algorithms. Therefore, the overall goal of this work is to interface SI models and regionalization algorithms to better estimate human movement patterns and reimagine criteria for building optimal cities while balancing data quality, privacy, and representativeness.

SI modeling provides a conceptual and technical foundation for explaining and predicting the flow of human movement [4, 12]. Importantly, SI models require only data at the aggregate level in order to make accurate predictions about movement patterns between places. This can be a mixed blessing: from a privacy standpoint, data aggregation means that an individuals behavior can no longer be traced back to them; but it raises the new issue of defining appropriate spatial units or aggregation procedures. An ideal

definition would successfully anonymize the individual level data and create a set of regions that accurately represents the communities at the heart of important policy goals. Algorithms to accomplish this task fall under the umbrella of regionalization.

Regionalization is the process of spatially aggregating areas into homogeneous regions ([7]) subject to various constraints, such as contiguity, number and scale of aggregated regions, or determining which areas need to be analyzed. In the context of SI models, regionalization procedures could be used to define novel functional regions—e.g., central business districts versus predominantly residential districts—that fulfill multiple criteria, potentially diverging from traditional boundaries, such as census geographies. This yields a meso-scale community-driven approach to building meaningful regions as opposed to a purely top-down administrative approach or a bottom-up data-driven approach. Viewing SI problems through this new lens could empower analysts to extract more pertinent insights from movement data and create more diverse solutions to urban problems.

In particular, this work begins the process of applying regionalization methods to SI problems. It seeks to answer the following pair of questions:

- What, if any, benefits can be incurred as a result of regionalizing a dataset before running spatial interaction models?
- How much regionalization should be done?

The first can be answered in a more qualitative manner, as evidenced by the description above, as well as in a quantitative setting by comparing computational results. The second will be answered by varying the level of regionalization and examining various statistics related to the models. In Section 2, I provide an overview of regionalization and spatial interaction algorithms, focusing on those which are used in this work. Then, in Section 3 I inspect examples of this methodology applied to a sample dataset and their output. I finish by interpreting these results in Section 4 and defining a heuristic criterion for determining a reasonable level of regionalization.

2 Methods

2.1 Regionalization

Precisely, the three regionalization algorithms explored in this work are Ward linkage spatial clustering, the Skater algorithm, and Region k-means (k-means clustering with spatial constraint). The Ward and Region k-means algorithms are both standard (aspatial) clustering methods which have added spatial constraints,

while the Skater algorithm is a spatially explicit method. It will be apparent from the descriptions of these algorithms that the number of clusters is a hyperparameter which is not inherently related to the algorithm or the problem setup. I address how to select this parameter in Section 4.

The Ward clustering procedure is an agglomerative or hierarchical procedure which can be thought of as the opposite of the Skater method [8]. Using the spatial weights matrix and data at each region, it computes the similarity between every pair of regions and merges the two most similar at each iteration until the requested number of clusters are formed. Skater (Spatial "K"luster Analysis by Tree Edge Removal) first creates a minimum spatial spanning tree connecting all the base level units of study [1]. In the `pysal.spopt` library used to carry out the analyses, this tree is created from a spatial weights (adjacency) matrix of the base level units. Then, the algorithm iteratively prunes the most dissimilar edges from the tree until the requested number of disconnected tree components remain. These components are returned as the clusters.

Finally, the spatial k-means regionalization uses the k-means algorithm with network distances as defined by the spatial weights matrix [9]. The aspatial k-means algorithm begins by choosing random vectors as the cluster centers, then iteratively recomputes the centers so as to minimize the intracluster variance, computed as a function of the distances between points and the cluster center. The spatial variant modifies how distances are incorporated into the algorithm, forcing a spatial contiguity constraint on this optimization procedure and leading to a regionalization.

2.2 Spatial interaction models

Only one kind of SI model was explored for this work: the unconstrained gravity model. This model can be expressed in the following equation ([12]):

$$T_{ij} = k\frac{V_i^{\mu} W_j^{\alpha}}{d_{ij}^{\beta}}$$

where T is the matrix of flows, V is a matrix of origin attributes (push factors), W is a matrix of destination attributes (pull factors), d is a matrix of distances between attributes, k is a normalization factor, and μ, α, and β are parameter vectors representing the effects of push factors, pull factors, and distance on the flows. These parameters are estimated in the model fitting phase. While the unconstrained gravity model is far from the most sophisticated method for SI modeling, it is still a reasonable and widely used framework which helped to simplify the implementation of this sample workflow. It could easily be substituted for many other more advanced methods at no cost to the ideas presented in this work.

2.3 Data and implementation

To limit the size of the problem to analyze, I considered county-to-county migration flows in California. I obtained these flows as well as county boundaries and data from the U.S. Census Bureau's 2014-2018 American Communty Survey (ACS; [3]). I considered three variables for each county: total population, percent white population, and median income. This work is more concerned with prototyping the workflow than with gaining new insights on migration data in Califonia; the variables were chosen with this in mind. One other challenge in the implementation was aggregating the flows by the new clustering. My code does this by brute force, considering every flow in the original matrix and constructing a new flow matrix by reassigning this flow to the proper units in the new regionalization.

I implemented the code in Jupyter notebooks using the Python libraries `numpy`, `pandas`, `geopandas`, `matplotlib`, `pysal.spint`, `pysal.spopt`, and `cenpy` ([5, 14, 6]). Each library serves a different purpose: `numpy` is for vector mathematics and computation, `pandas` and `geopandas` are used for data management and manipulation, `matplotlib` is used for visualizations, the Census data is gathered from `cenpy`, and algorithms for spatial interaction and regionalization are taken from `pysal.spint` and `pysal.spopt`, respectively.

3 Results

Figure 1 shows an example of a regionalization on California counties using the Ward algorithm and aggregated by median income in each county. After aggregating the flows by the new regionalization, I ran a gravity model and successfully fit the model to the data with a standardized root mean-square error (SRMSE) of 1.312. From here, I swept in number of clusters for each regionalization algorithm (Ward, Skater, and Region k-means) on median income as well as for Ward regionalization on population (Figures S.1-4). For each sweep step, I recorded the Akaike information criterion, the pseudo-R^2, and the SRMSE model fit statistics as well as the average unit area for the regionalization. Additionally, I ran each cluster level many times and averaged the results at this cluster level to smooth out the effects of randomness inherent in these algorithms.

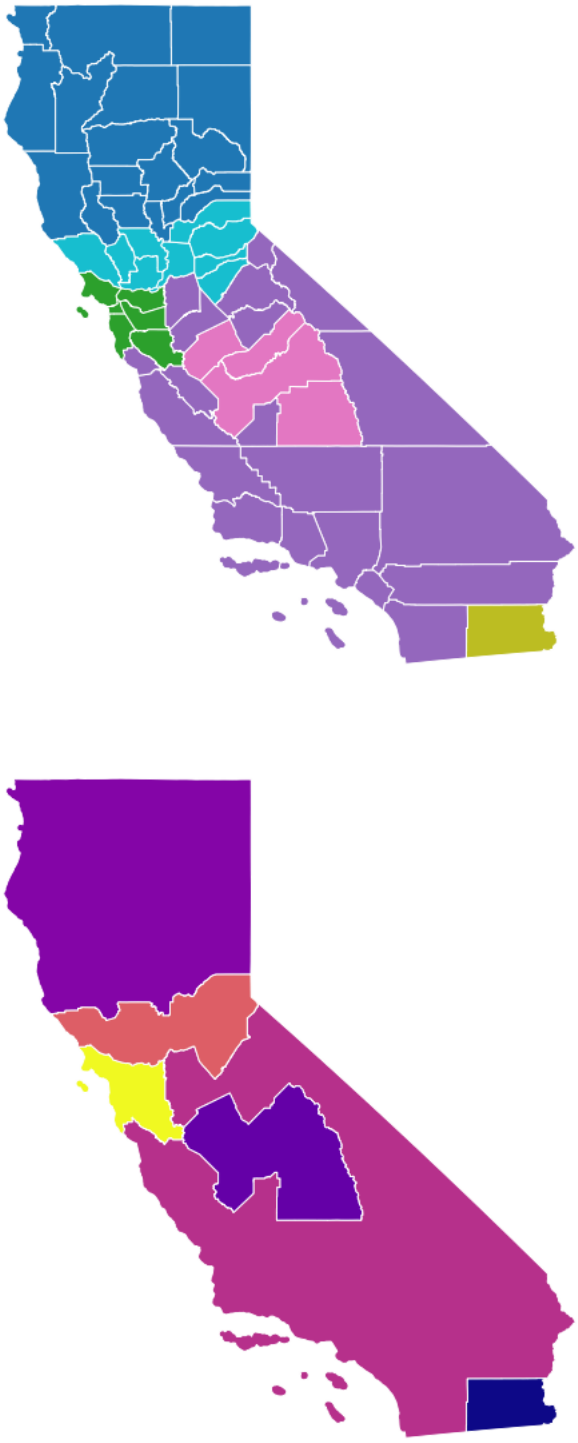

Figure 1: California with Ward linkage regionalization by median income. Top: base counties, pre-aggregation, colored by cluster. Bottom: median income plotted for each cluster (now aggregated).

4 Discussion

For each of these sweep outputs, the results are similar. AIC, pseudo R^2, and SRMSE are all monotonically increasing functions of the number of clusters, while average area per unit is a monotonically decreasing function of the number of clusters. This begs the question of how much regionalization should be done. To develop the answer, we establish our objective: choose a number of clusters that

- maximizes goodness-of-fit of the SI model and
- minimizes the amount of regionalization performed, as more regionalization leads to more information loss.

To do this, I developed the following heuristic:

$$k_{\text{crit}} = \min_{k \in [4,N]} \left[\frac{1}{M_S} \text{SRMSE}(k) + \frac{1}{M_{\bar{A}}} \bar{A}(k) \right]$$

where $M_S = \max_{k \in [4,N]} \text{SRMSE}(k)$ and $M_{\bar{A}} = \max_{k \in [4,N]} \bar{A}(k)$ are normalizing factors. This criterion works by simultaneously minimizing average area and SRMSE, both of which need to be minimized to achieve the outlined objectives. Since it minimizes the sum, it finds the number of clusters that leads to the least amount of both quantities, rather than preferring one or the other. Finally, the normalizations are required so that the two quantities exist on the same scale and are thus comparable in the minimization process.

Figure 2 shows the optimal selection plot and the optimal clustering for Ward linkage regionalization on median income. Figures S.5-7 are similar plots for Skater regionalization, Region k-means regionalization, and Ward regionalization on total population. Interestingly, while the three algorithms have different values of k_{crit}, the clusterings all share similar patterns. This speaks to the robustness of this heuristic to the choice of algorithm and points to some underlying optimal regionalization that all three algorithms are getting close to. However, one important limitation of this heuristic is that it is just that: a heuristic. There is no *proof* of optimality for k_{crit}; it simply appears to perform well with these algorithms. Further work is required to verify if it is indeed a good estimate for the optimal number of clusters.

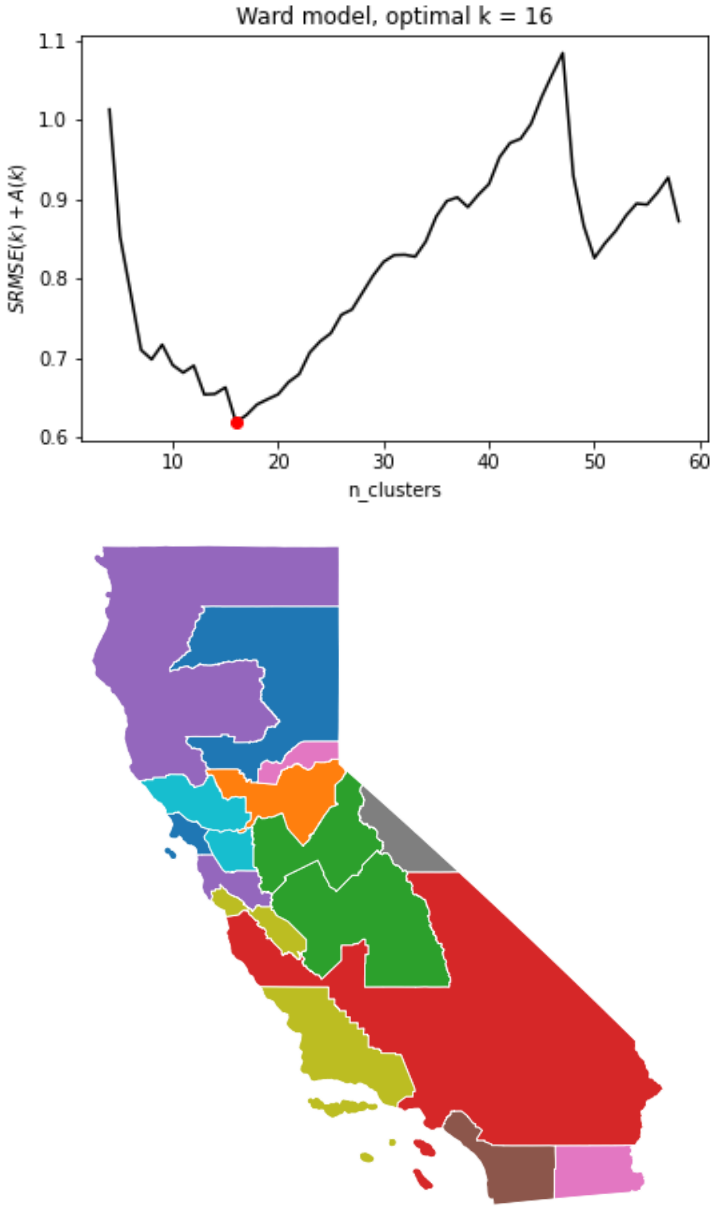

Figure 2: Optimal clustering selection plot for Ward linkage clustering (top) and optimal clustering (bottom).

5 Conclusion

This paper explores the intentional addition of regionalization into the SI modeling workflow and derives a heuristic criterion for determining a reasonable level of regionalization to use. This work contributes

towards several important problems in spatial data science. Firstly, the integration of regionalization algorithms into SI models adds another criterion for judging the quality of a regionalization algorithm: its performance for analyzing movement data and its ability to maintain data anonymity and quality. In this way, this project contributes towards progress on the modifiable areal unit problem (MAUP)—the issue that spatial analysis results are dependent on the way in which they are aggregated—in the context of movement data and SI [10, 11]. By designing theory-driven regions based on multiple criteria, the number of sensible alternative zoning systems based on arbitrary modifications is reduced. Critically, the projects focus on community-driven meso-scale regionalization creates a transparent environment for designing and testing new algorithms. By considering who cities are optimized for at the time of model creation, this workflow promotes representativeness and equity in the modeling process. Lastly, implementing regionalization algorithms into traditional SI workflows provides new means to model urban systems and understand how cities evolve over time.

Additionally, This work motivates the meaningful aggregation of high-resolution data and the fusion of traditional and non-traditional data sources, unlocking previously unimaginable functional regions for use in SI models and facilitating strategic urban development. Importantly, the data privacy and quality concerns of simulating individual level flows are no longer relevant at this new meso-scale aggregation level. This has vast implications in the realm of real-time urban planning: with live feedback and analysis available at their fingertips, planners can better predict how urban systems interact with epidemics, natural disasters, dependence on infrastructure, and threats to national security. Such an implementation will bridge the gap between theory, methods, and applied research, promoting the accessibility of the technological innovations developed in this project.

5.1 Future Work

There are many directions for future work from this paper. To begin, using more realistic data would give more realistic answers and could provide a practical perspective from which to analyze these proposed procedures. Both the regionalization and spatial interaction aspects of this workflow can also be made more complex: this work only considers univariate regionalization, while all of the described regionalization algorithms can very easily be used in on multiple variables at once, potentially leading to more natural clusterings and better model fit. Similarly, there are many SI models that may lead to better results than an unconstrained gravity model, such as production-constrained gravity models and local SI models, and many regionalization modes that may have alike effects such as the max-p regions method [13]. Finally, it would be very interesting to develop a fixed-point iteration which feeds the SI results back into the region-

alization and repeats the procedure until optimality is achieved (the clusters stop significantly changing). This would be a much more significant undertaking, but would represent a highly stable way to run this workflow.

References

[1] R.M. Assunção, M.C. Neves, G. Câmara, and C. Da Costa Freitas. Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographic Information Science*, 20(7):797–811, 2006.

[2] Michael Batty. *The new science of cities*. MIT Press, 2013.

[3] U.S. Census Bureau. County-to-county migration flows: 2014-2018 ACS. census.gov, 2018.

[4] A.S. Fotheringham and M.E. OKelly. *Spatial Interaction Models: Formulations and Applications*. Kluwer Academic Publishers, 1989.

[5] Charles R. Harris, K. Jarrod Millman, Stfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernndez del Ro, Mark Wiebe, Pearu Peterson, Pierre Grard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357362, 2020.

[6] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science Engineering*, 9(3):90–95, 2007.

[7] S.J. Rey J.C. Duque, L. Anselin. The max-p regions problem. *Journal of Regional Science*, 52:397–419, 2012.

[8] J.H. Ward Jr. Hierarchical clustering to optimize an objective function. *Journal of the Amercan Statistical Association*, 58:236–244, 1963.

[9] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[10] S. Openshaw. Optimal zoning systems for spatial interaction models. *Environment and Planning A*, 9:169–184, 1976.

[11] S. Openshaw. *The modifiable areal unit problem*. Norwick: Geo Books, 1983.

[12] T.M. Oshan. A primer for working with the spatial interaction modeling (spint) module in the python spatial analysis library (pysal). *REGION*, 3(2):11, 2016.

[13] R. Wei, S. Rey, and E. Knaap. Efficient regionalization for spatially explicit neighborhood delineation. *International Journal of Geographical Information Science*, 2020.

[14] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

Supplementary Figures

These figures can also all be found in better quality in the repository under `code`.

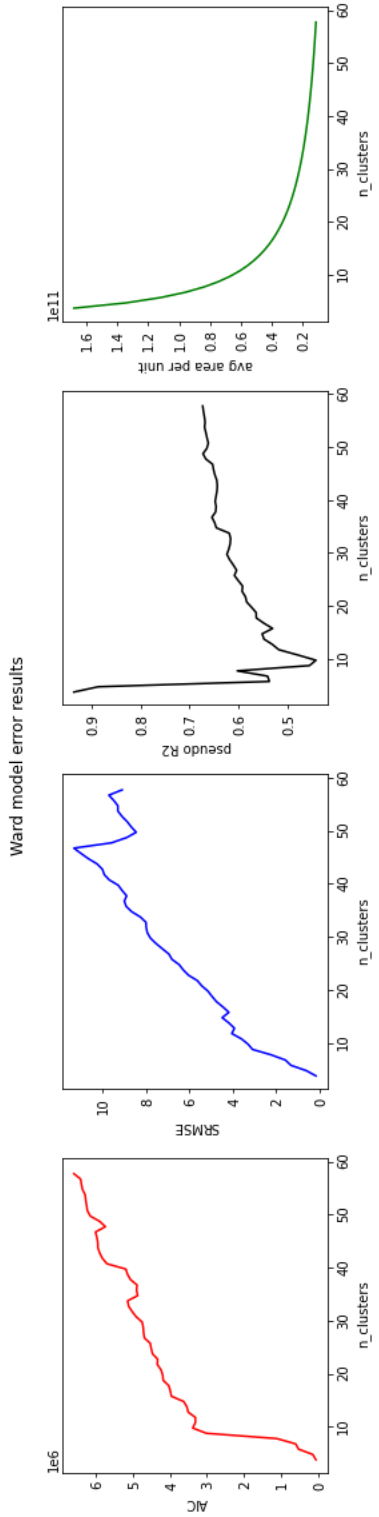

Figure S.1: Sweep results for Ward regionalization.

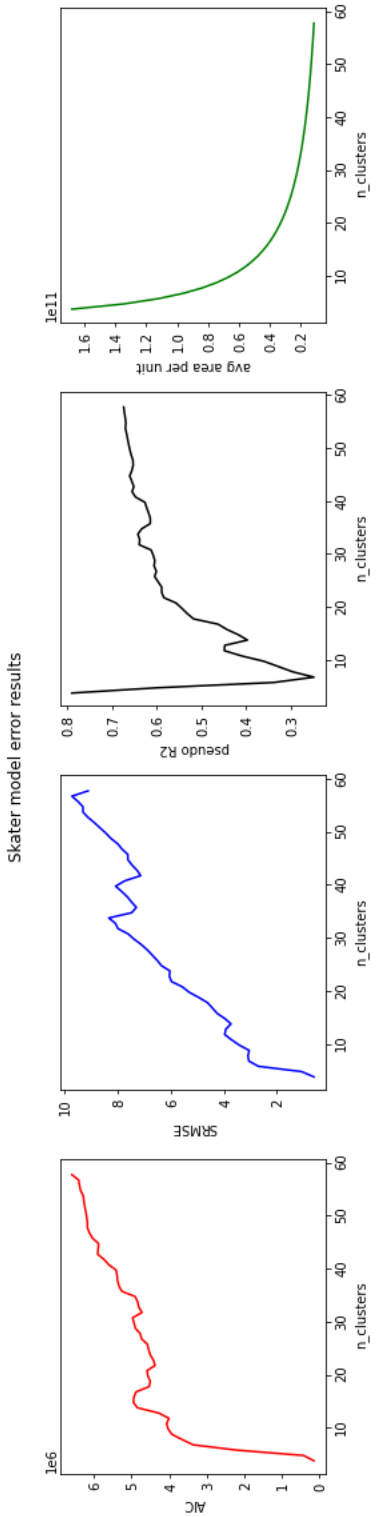

Figure S.2: Sweep results for Skater regionalization.

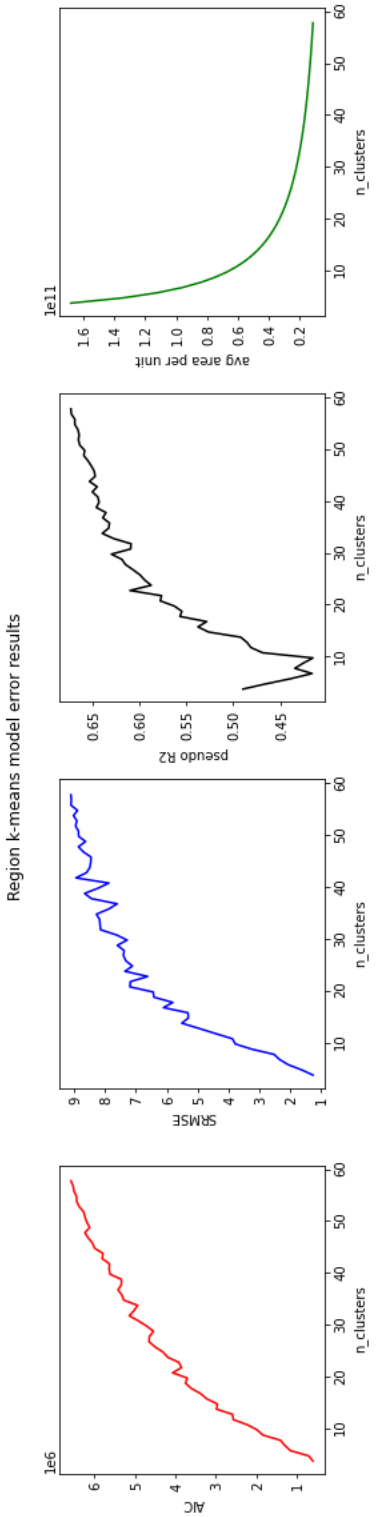

Figure S.3: Sweep results for Region *k*-means regionalization.

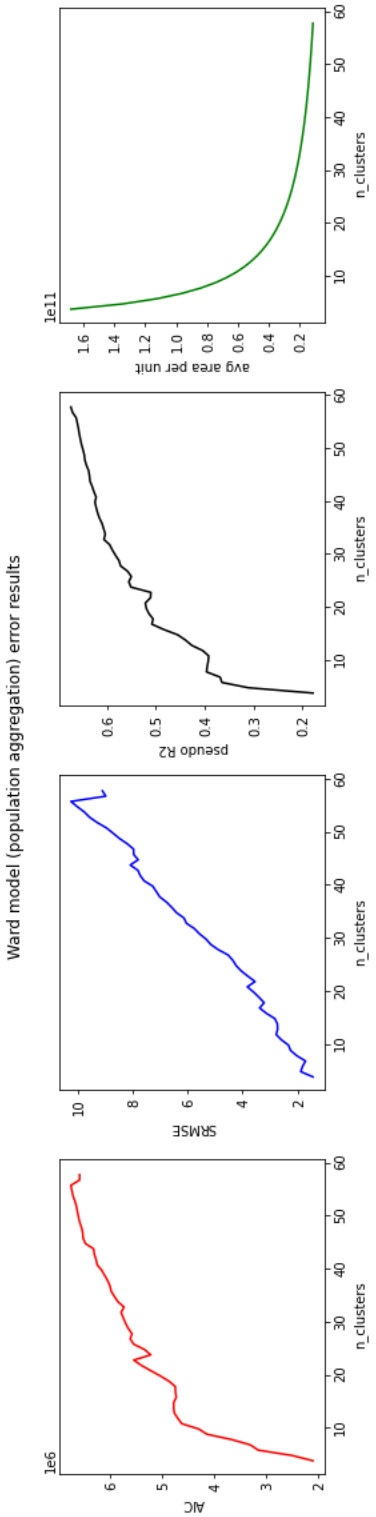

Figure S.4: Sweep results for Ward regionalization on total population.

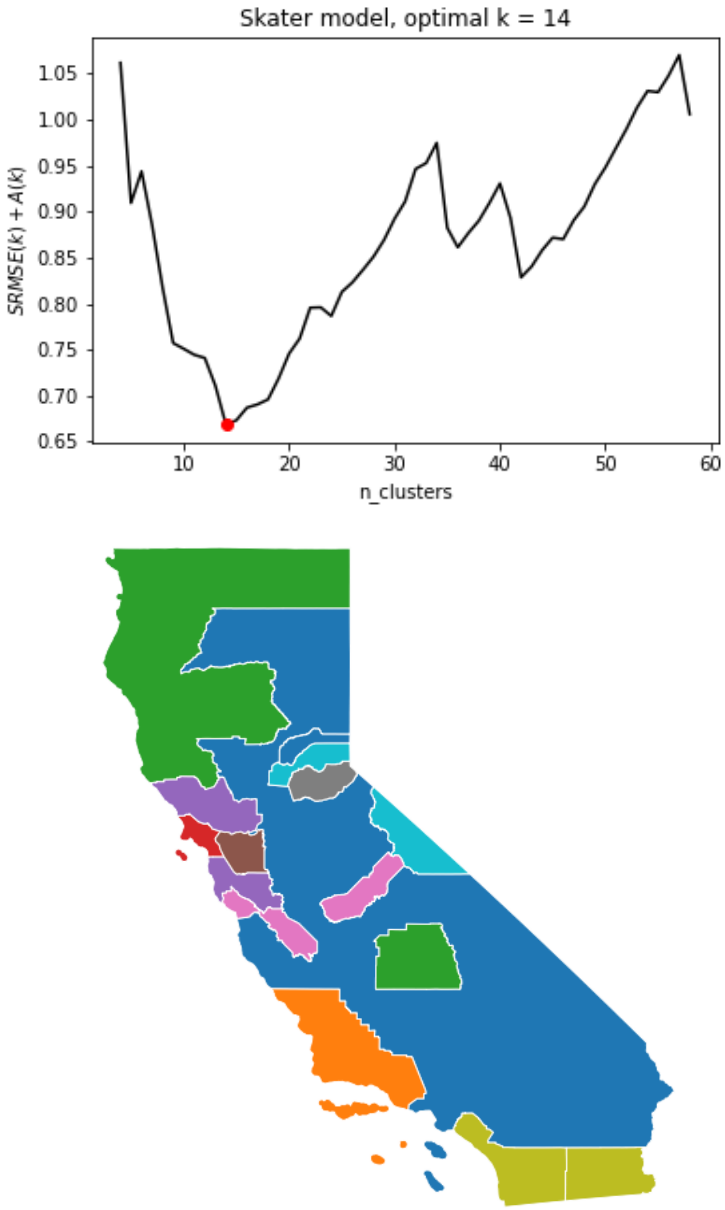

Figure S.5: Optimal clustering selection plot for Skater clustering (top) and optimal clustering (bottom).

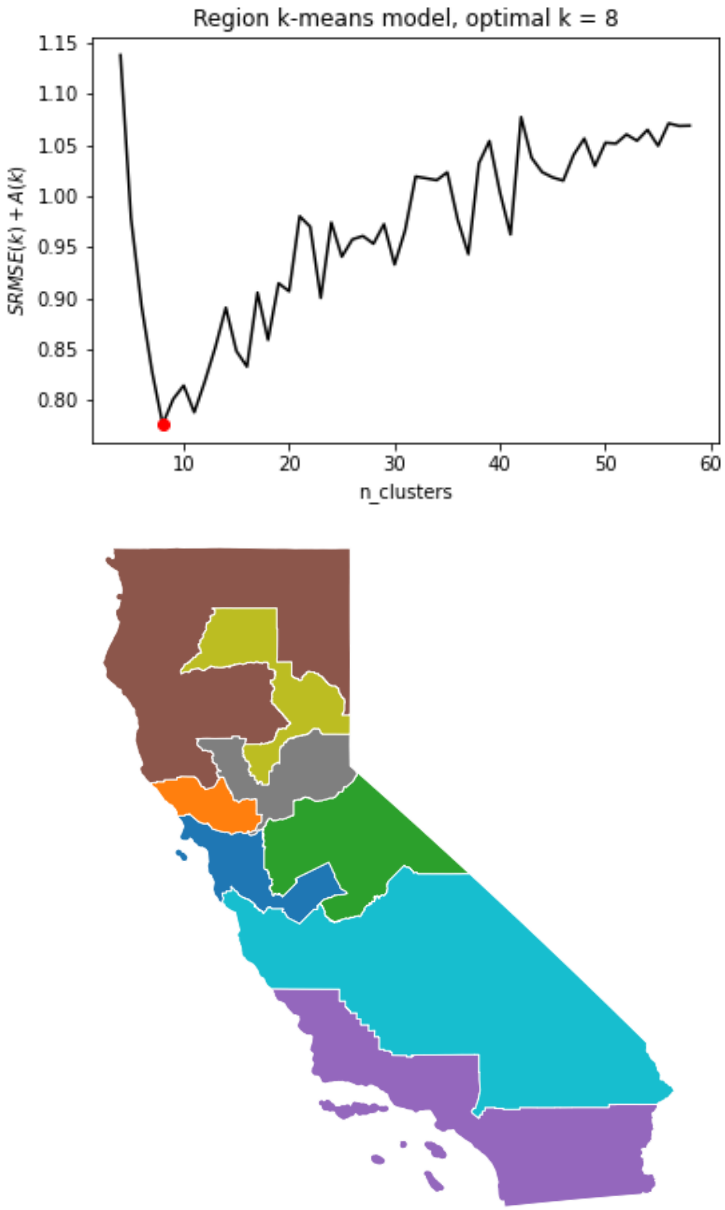

Figure S.6: Optimal clustering selection plot for Region *k*-means clustering (top) and optimal clustering (bottom).

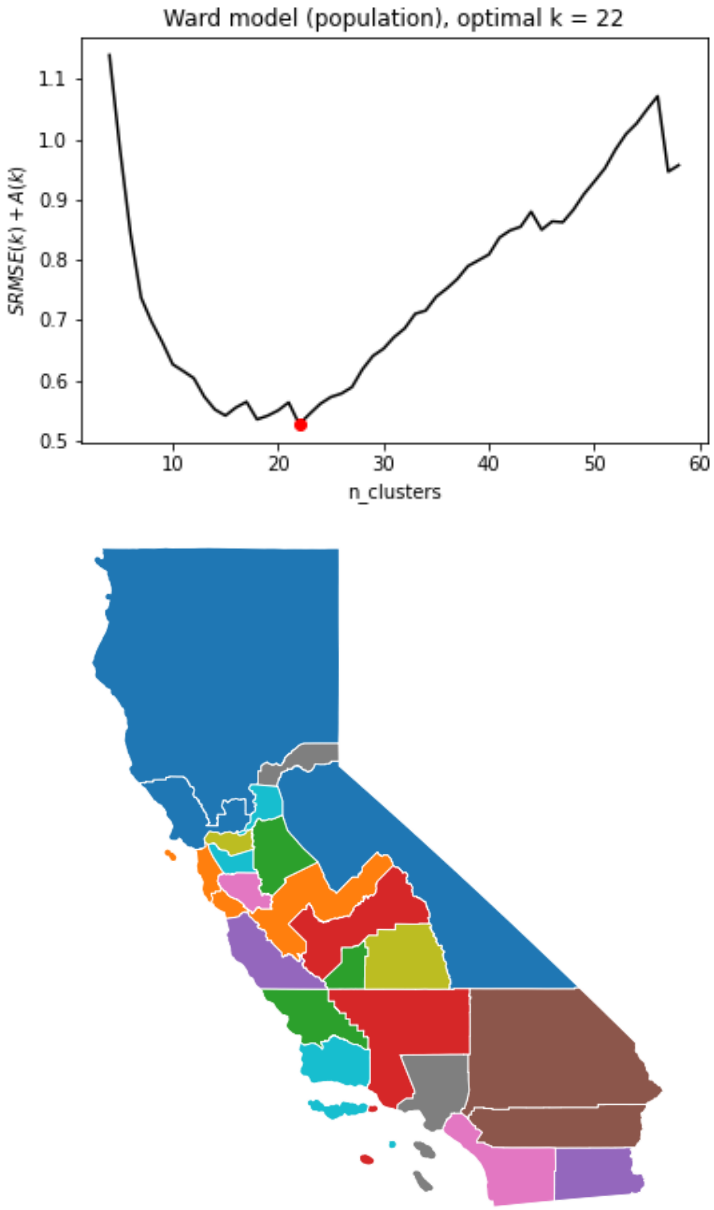

Figure S.7: Optimal clustering selection plot for Ward linkage clustering on total population (top) and optimal clustering (bottom).