Measuring and Mapping Recreation with Social Media

http://www.medialives.com/wp-content/uploads/2013/05/smap.jpg

Matthew Stelmach & Christopher Beddow | GEOG 569 | August 19th, 2016

Recommended Course of Action

Measurement of visitation using social media data opens up new alternatives to traditional survey methods. Trails in Mount Baker-Snoqualmie National Forest see use for a variety activities and can be measured using multiple social media platforms. Comparing usage rates estimated from social media data against traditional visitation survey data from the United States Forest Service serves as a method of validation. This project tests the validity of Flickr, Twitter, and Strava data for measuring visitation.

The results of this project indicate that of the social media data sources used, Flickr is the strongest predictor of national forest permit data. We recommend collection of expanded permit data in order to build a more robust dataset that measures a greater quantity of visitation over a larger scale of time. We also recommend conducting a non-linear geographic regression in order to build a better model of recreational use across the study area. Finally, we recommend continued monitoring of data availability from a variety of sources as new technology emerges and accessibility of existing data changes.

Contents

List of Figures & Tables

List of Abbreviations

AOI – Area of Interest
API – Application Programming Interface
DV – Dependent Variable
FS – Forest Service
GDP – Gross Domestic Product
GWR – Geographically Weighted Regression
InVEST - Integrated Valuation of Ecosystem Services and Tradeoffs
IV – Independent Variable
JSON – JavaScript Object Notation

MBSNF – Mount Baker-Snoqualmie National Forest
NVUM – National Visitor Use Monitoring
PUD – Flickr Photo User Day
TUD – Twitter User Day

1. Introduction

1.1 Scope

Recreation is one of the primary uses of US Forest Service (USFS) land, and can be one of the most challenging to quantify. This project seeks to develop and validate methods which can provide insight concerning which trails are most trafficked and why. Publicly available social media is used as a proxy for visitation to estimate trail usage rate. The project is a contribution to the larger USFS goal of developing a model that makes accurate usage predictions with minimal time, funds, and effort applied toward data collection.

This study focuses specifically on social media data that reflects visitation on hiking trails in Mount Baker-Snoqualmie National Forest (MBSNF) in western Washington. Current USFS data on visitation for hiking trails is incomplete. The current data collection methods are both expensive and time consuming to collect. Current standard for USFS is the National Visitor Use Monitoring (NVUM) method, and a separate permit data collection method, in cases where trailheads require hiker registration. Previous research Wood et al. (other), has shown that social media can be used as a proxy for visitation. Our goal is to establish a method for formatting trail polyline data into a polygon that can be used to spatially query geotagged social media in order to measure trail-usage density. This project considers utilizing the Strava API to supplement the existing InVEST toolkit. Statistical modeling techniques are applied in order to further compare all available data sources and to validate social media derived visitation rates against NVUM and permit collection visitation data.

1.2 Background

Recreation is a major industry in the US. In 2012, the Outdoor Industry of America reported that the sector contributes 6.1 million jobs in addition to $646 billion in outdoor recreation spending. The world travel and tourism council reports that travel and tourism generated $7.6 trillion in 2014, equivalent to 10% of the global GDP. Recreation and tourism can have a large influence on economies, accounting for 20% of the economy in parts of the US (Laney 2009). Spending can overall be treated as a metric for the value people place on a resource. Tourism and recreation come in many forms and purposes, including fitness, leisure, aesthetics, and spiritual values.

Land managers need data on recreational use in order to evaluate policies and management options. Traditional data collection methods can be costly and labor intensive to collect. Mail surveys are estimated to cost between $5000 and $10,000, while on-the-ground personal surveys can cost an extra 50-150% (ASA 1997). In addition, the results of on-the-ground surveys can prove to be unreliable. Variation in time of visit, length of stay, and season of the year can all contribute to a variation in usage estimation. Other researchers (Pergams and Zardic 2008, Warnick et al 2011) have attempted to use surveys and internet search results to understand use rates, with conflicting conclusions.

Social media is closely linked with recreation and tourism, as people share and document their experiences on a variety of platforms. Three sources are used in this study: Flickr, Twitter, and Strava. Each of these has a unique interface, with particular ways of collecting user data and providing metadata about use. Each platform also has a different user base, ranging in size and in type of activity that it is oriented toward. Finally, these platforms are all similar in that they offer an API built on similar principles, allowing data to be searched.

Flickr is a photo sharing platform estimated to have 112 million users (2015). Flickr also hosts a collection of over 10 billion images with an average of 1 million ongoing uploads per day. Photos have rich metadata that includes connections to other users, tags, captions, and geolocations. Twitter is a microblogging platform with an estimated 310 million active users (2016). Twitter users can share up to 140 characters in a public space, with each post referred to as a "tweet". Tweet metadata can include geotagged locations, as well as referenced locations from the text of the tweet. Strava is described as a run, ride and cross-training monitoring and sharing platform with an estimated 8.2 million users (http://markslavonia.com/sampling-strava//). Strava users share "segments" which describe a trail or other use area with attributes for length, elevation profile, location, and time. Strava emphasizes athletic training, with the ability to compare performance against previous efforts on a segment. Strava also has a limited API, with any personally identifiable information being strictly censored unless the user explicitly chooses to share. From Strava, information about segments such as creation date, location, number of efforts and number of athletes can be determined. This is sufficient to provide an average use rate, without the granularity of individually timestamped segment efforts. Figure 1 shows estimated use rate by country. In the US less than 1% of the population is estimated to use Strava.

	Sampled members	Percent of the sample	Estimated Members	National Population	Members as % of population
USA	81	29.24%	2,397,834	318,900,000	0.75%
UK	51	18.41%	1,509,747	64,100,000	2.36%
Brazil	20	7.22%	592,058	200,400,000	0.30%
Australia	20	7.22%	592,058	23,130,000	2.56%
Italy	16	5.78%	473,646	59,830,000	0.79%
Netherlands	12	4.33%	355,235	16,800,000	2.11%
France	10	3.61%	296,029	66,030,000	0.45%
Germany	9	3.25%	266,426	80,620,000	0.33%
Spain	8	2.89%	236,823	47,270,000	0.50%
Canada	6	2.17%	177,617	35,160,000	0.51%

Figure 1. Strava users by country, from random sampling of user population: http://markslavonia.com/stravas-global-reach-confirmed-by-sampling/

	Social				Individual	
Affective	**Mutual Experience.** Images intended to enrich a shared, co-present experience (either in the moment or later as a memento).	103 (35%)	**Absent Friends or Family.** Images intended for communication with absent friends or family (either in the moment or later).	63 (21%)	**Personal Reflection.** Images intended for personal reflection or reminiscing.	120 (41%)
Functional	**Mutual Task.** Images intended to share with people co-present in support of a task (either in the moment or after the event).	11 (4%)	**Remote Task.** Images intended to support a task by sharing with remote family, friends or colleagues (either in the moment or later).	23 (8%)	**Personal Task.** Images intended to support some future task not involving sharing.	29 (10%)

Figure 2. Motivations for sharing photos as a representation for motivations for participating in social media.

Kindbert et al. 2005 describes photography use as shown in figure 2. The expressed values largely correlate to the use of social media and can provide insight into which locations are shared and why. The perceived value of a trip may influence if the experience is shared on social media, leading to a bias toward longer travel distances represented in social media (Wood et. al. 2013).

The social media platforms selected were chosen because they provide rich data and metadata based on user input. The social media data is required to be geocoded, dated, and capable of identification for use rate per day. Additionally, the data must be publicly available at all scales for the development of a public app. Of the sources reviewed, Twitter, Flickr, and Strava met the requirements. Sources review but rejected included Google+, Instagram, and Facebook. Other potential data sources were Cellular Network Providers, and Map Service providers. The sources rejected were primarily fee based or limited in available data, neither of which were appropriate for this project.

1.3 Social-Ecological System

The study area is Mount Baker-Snoqualmie National Forest (MBSNF) in the western part of Washington State shown in figure 3. This area is managed by the USFS, and stretches from the Canadian border near North Cascades National Park to Mount Rainier National Park in the south. MBSNF encompasses over 7000 square kilometers of forest, as well as over 1500 kilometers of trails. Aside from hiking opportunities, the area also includes Snoqualmie Pass, Mount Baker, Crystal Mountain, and Stevens Pass ski resorts. All of these recreational opportunities are a major draw for visitors, especially those coming from the nearby urban areas of the Puget Sound. NVUM data shows a mean visitor travel distance between 51 and 75 miles.

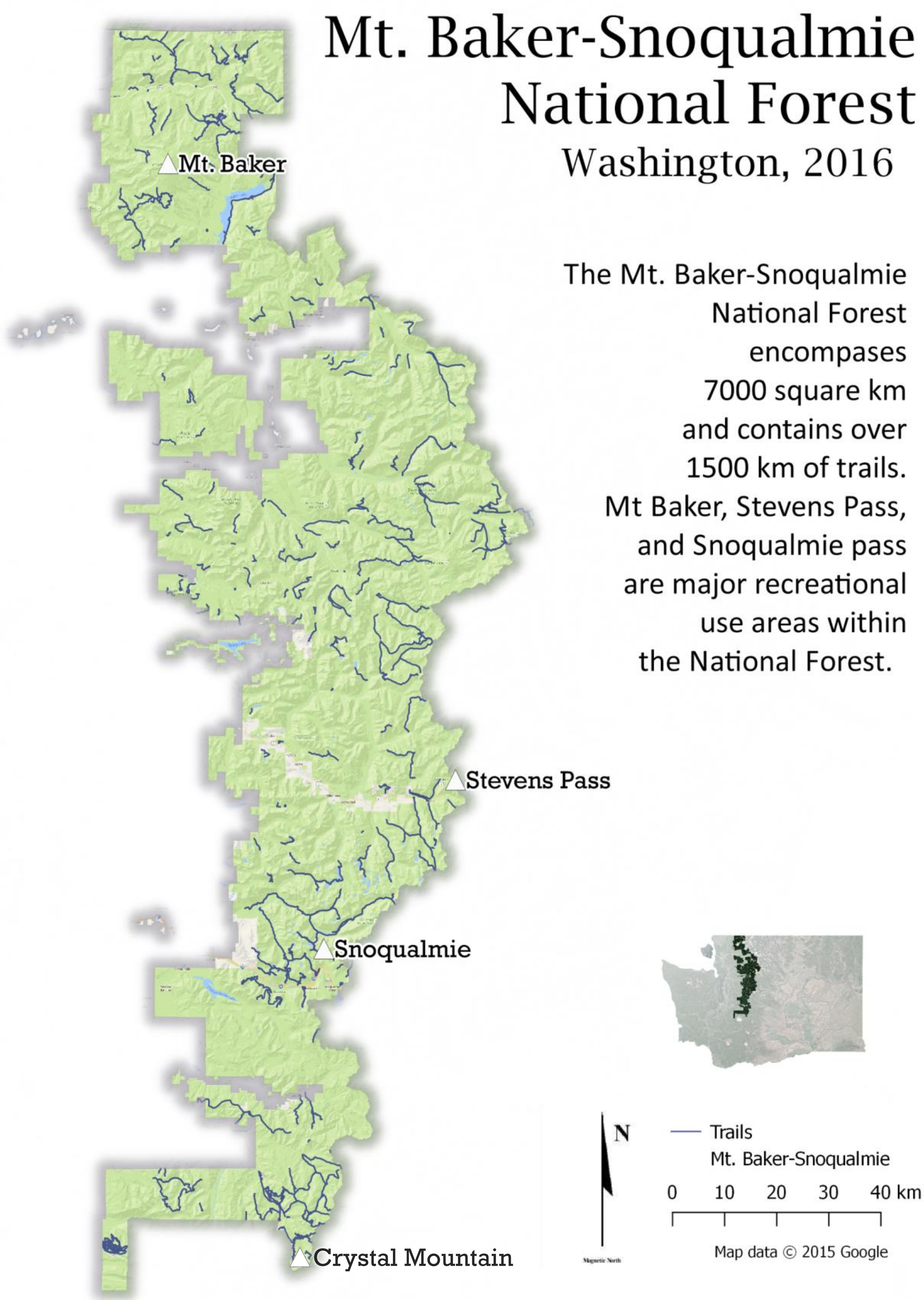


Figure 3. The area of interest, the Mt. Baker-Snoqualmie National Forest.

Recreation is an example of a cultural ecosystem service, a benefit provided by a natural environment. The benefits of cultural ecosystem services can be particularly difficult to map and quantify, as they include activities that are rather intangible, just like those mentioned above as activities documented by social media: appreciation of scenery, outdoor activities, tourism, exercise, sport, and spiritual fulfillment. Paracchini et al. (2014) conducted a study in the European Union attempting to model a recreation potential index and an assessment of potential demand for recreation in outdoor areas. This study also sought to assess accessibility of recreation sites, as well as characterize the people who use these sites. The team integrated visitor surveys from natural areas in Finland, the UK, and Denmark into their work. The findings indicated that factors including culture, age, type of living environment, latitude of location, and social status all affected the behavior of individual actors, while the landscape itself had varying levels of attraction for these people. For example, water and forests had a stronger attraction, while grasslands weren't particularly attractive. Overall, the study by Paracchini et al. indicates that it is necessary to assess the features of the landscape as well as to survey the actual visitors.

This study of MBSNF considered similar factors, including the visitor surveys and the location of hiking trails and ski resorts. A key difference in this study is the integration of social media data as a supplement visitor surveys. As seen in the description of different social media platforms and their demographics, social media offers access to a far broader group of recreationists, and opens the possibility of mapping specific locations of visitor activity within an area of interest. This study seeks to investigate visitation on the trails within MBSNF, and thus offer a greater degree of specificity. The accuracy of our social media data was checked against the available visitor surveys, in order to compare the efficacy of the method and measure correlations.

	Social	Economic	Biophysical
National Forest	Proximity to urban area, smaller towns nearby, recreation potential based on possible activites	Land use planning, infrastructure, land management	Forest management, wildlife management, environmental policy
Forest Regions	Designated parks and recreation areas, systems of trails, local communities with outdoor access	Regional economies, usage fees, highway access	Regional landscape type, access to natural areas
Trail Segments	Trail specific potential based on difficulty, location, reputation, and other social values	Maintenance and upkeep, parking, visitor amenties	Rivers, lakes,mountains, and other natural features as recreation attractions

Figure 4. Social-Ecological System table. National Forest was the primary focal scale with National forest above and trail segments below.

The study area is examined at three focal scales illustrated in Figure 3: the entire national forest, two specific regions (North Bend and Skykomish), and individual trail segments. Social media allowed examination of the social aspects of the study area on these three focal scales, including the recreational value. Further background information also revealed social importance of specific areas, such as designated parks or recreation areas, as well as the economic importance of such locations as ski resorts. Finally, the biophysical aspects of the study area were apparent by examining where the actual trails fell in relation to natural features. Similar to past studies, it is apparent that natural features such as lakes, rivers, and mountains all are attractive sites for recreation.

2. Methods

This study uses multiple sources of geotagged social media data to estimate numbers of recreational trail users. Overall 172 trails are represented with usage counts from Forest Service data and social media data. The trails wind through a variety of landscapes that see year round use, including hiking areas, mountain biking routes, ski areas, snowshoeing routes, high alpine traverses, and casual day use areas.

Flickr has been identified as a source of data on visitation rates (Wood *et al.* 2013). This project uses Flickr, Twitter, and Strava as sources of recreational use. Strava data represents the latest addition to the compiled social media. Strava is a platform for sharing running and biking efforts along geotagged trail segments. These social media platforms are compared to Forest Service permit data (FS). The FS permit data, although not without error, is thought to represent real visitation rates. FS data is collected from permit data for trail use. The data comes from surveys collected in spring to summer 2011 and fall 2013. During permit data collection, there was a minimum of one survey date per trail and a maximum of 20, with a median of two. The data is collected based on users entering areas that require a submission of a hard-copy permit at the trailhead. An ancillary goal of the project was to develop all methods in open source. Open source scripts for buffering, scraping, and merging data for analysis can be found in appendix A – E.

2.1 Data Collection and Processing

Flickr and Twitter data are processed using existing scripts to quantify ‘user days’ per area. For each photo from Flickr, metadata includes geographic coordinates, unique user ID, and date the photo was taken. One user-day in a location is defined as one unique user who took one or more photos on a unique day, in that location (Sharp et al 2016). For each trail, we count the total number of user-days across 10 years of photos from Flickr. Twitter user-days follow the same definition as Flickr.

Strava data is scraped from the Strava API using the Strava Scraper python script (Appendix B and C). Of the data that we collected, the important pieces are the creation date and retrieval date of the Strava segment, and the count of efforts per segment (total uses of that particular route). Daily use rate is calculated as efforts divided by age (years) divided by 365 (days), to provide average daily usage rates. Raw data is converted to shapefile with the JSON to shapefile script (Appendix D). From the shapefile any segment with a use rate equal to or less than 0 or a segment with an age less than 1 month (0.085 years) is removed to reduce variation in the use rate. The segments are then intersected with the trail buffer polygons and the use rate is summed by polygon, this is done with the intersect polygon script (Appendix E).

2.2 Buffer Distance

The goal of the buffer analysis is to develop a simple method to generalize trail segments to polygons which can capture photo and tweet locations. The buffers are also used to intersect

Strava data. To find an appropriate buffer distance the buffer must: minimize overlap of trail polygon segments to reduce the likelihood that use data is double counted, and minimize the number of photos/tweets that are falsely included as representing a trail. Varying distances were created and compared with the elbow method to find the point of diminishing returns where the number of trails with zero user days was minimized while the number of user days counted for multiple polygons was also minimized. Buffers were created with an open source python script (Appendix A)

2.3 Comparison

This study uses a total of 4 datasets, Forest Service permit data (FS), Strava Average use rate (Strava), Twitter user days (Twitter), and Flickr Photo user days (Flickr). Each data set is plotted on a histogram, transformed with log natural functions in R: log1p or log(Data+1), and plotted again. A correlation comparison matrix is generated for all of the variables. For each pair a simple regression was performed. Finally, a multiple regression is performed to find the most parsimonious model. A summary of analysis is illustrated in Table 1.

Table 1. Statistical models for each data set.

Histograms	All
Simple Linear Regression	FS ~ Flickr FS ~ Twitter FS ~ Strava
Multiple Regression	FS ~ Flickr+Twitter FS ~ Flickr+Strava FS ~ Strava+Twitter FS ~ Flickr+Twitter+Strava
Getis-Ord Gi*	FS ~ Flickr FS ~ Twitter FS ~ Strava
Geographically Weighted Regression	FS ~ Flickr FS ~ Twitter FS ~ Strava*

*ArcGIS was unable to complete GWR for Strava Data: "Results cannot be computed when there is either severe global or severe local multicollinearity (redundancy among model explanatory variables)."

In order to have a statistically significant comparison, a forest-wide comparison is also performed on a grid of 500x500m cells covering the MBSNF. Each cell is used to calculate user days for Flickr, Twitter, and Strava. Each cell is measured for distance to the nearest trail segment. A Getis-Ord Gi* hotspot analysis is performed to see if there is local variation within the data values. A geographic weighted regression is also performed with the distance to trail as the dependent variable and each of the social media platforms individually as the independent variable.

3. Results

3.1 Buffer Distance

Table 2. Summary statistics for the Buffer sensitivity analysis.

Buffer Delta	100-200m	200-300m	300-400m	400-500m	500-600m	600-700m	700-800m
PUD Delta	532.9-770	770-971.8	1144.8-971.8	1336.8-1144.8	1532.3-1336.8	1735.9-1532.3	1898.8-1735.9
Slope	0.4	0.5	0.6	0.5	0.5	0.5	0.6
Percent Change PUD	144.5%	126.2%	117.8%	116.8%	114.6%	113.3%	109.4%

For Mt. Baker-Snoqualmie National Forest, 500 meters was determined to be the best middle ground. In considering the performance of different buffer distances against the number of Flickr photo-user-days, for example, there was a linear relationship between buffer distance and total Flickr photo-user-days until the 500 meter buffer.

Measurement of change between different buffer values was expressed in the form of a buffer delta and PUD delta. These represent the raw change between values. Slope was also calculated based on the quotient of these deltas (Buffer Delta/PUD Delta = slope), as well as a percent change (PUD Delta high/PUD delta low = percentage change) in reference to the change of raw value in PUD. After 500 meters, the linear relationship began to level off subtly, and buffers up to 800 meters showed less drastic gains. As seen in Table 2, the slope decreases starting at the 400 to 500 meter increase, so anything beyond 500 meters sees a plateau in slope until 800 meters, suggesting a sort of “dead zone” between 500 meters and 800 meters. A buffer of 800 meters in some cases would begin to encompass data from other nearby trails. As a result, 500 meters was determined to be the threshold point at which the buffer stopped capturing the core data surrounding the trail of interest in the case of Flickr. This determination was applied to other social media data sources as well, in order to maintain a constant buffer while comparing different data sources as variables.

Compared to the expert digitized trail buffers the 500m buffer shares 259.83 square km and differs by 283 square km. Figure 5 shows the areas digitized by experts in green, and standard 500m buffers in purple. The areas are subjectively congruent, with some expert digitized polygons exceeding the bounds of the buffer and some buffers exceeding the bounds of the expert digitized polygons.

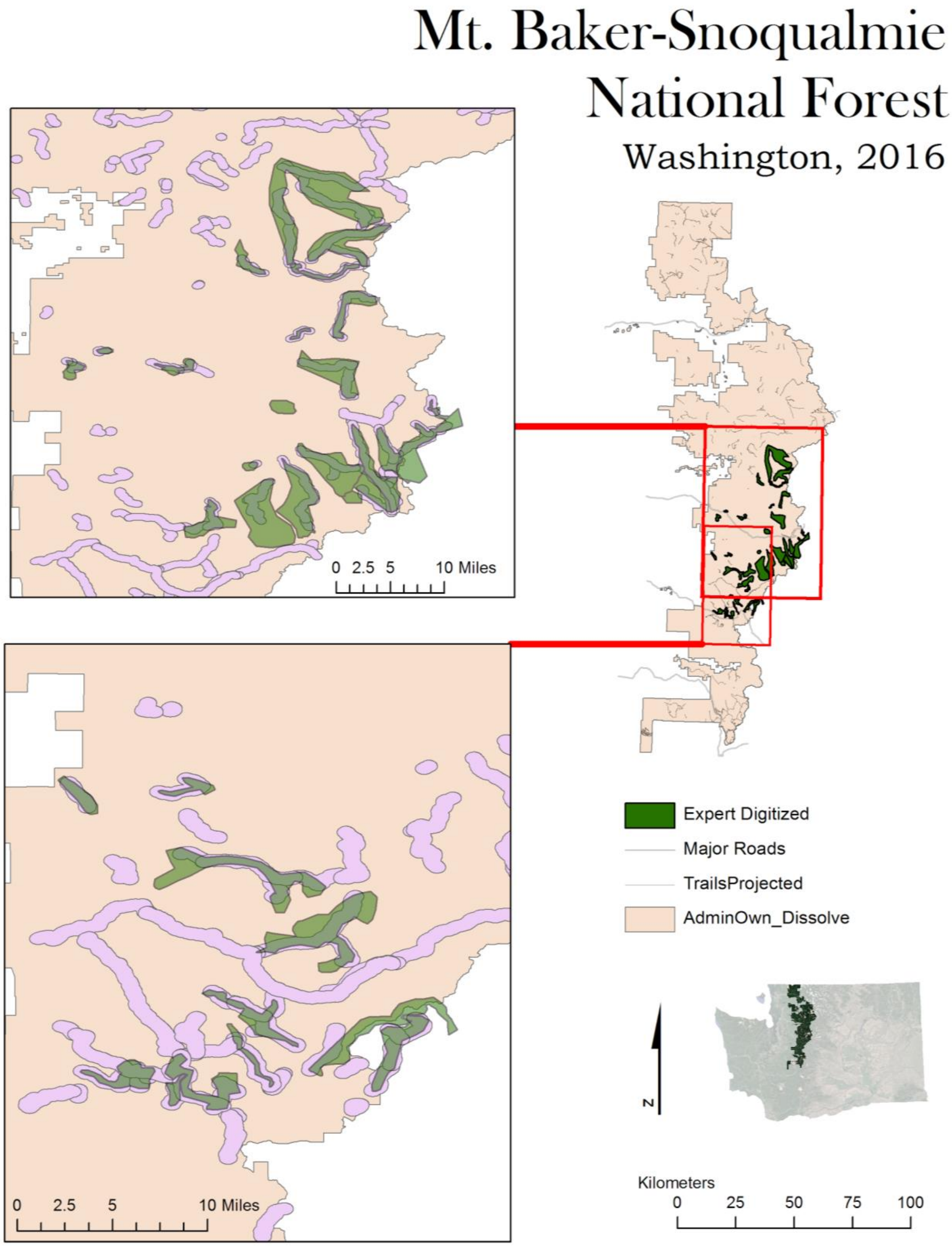


Figure 5. Standard 500m buffer compared to trail segments digitized by a local expert.

3.2 Strava Script Performance

Strava collection via the Strava API scraper is illustrated in Figure 6. The scraper starts with the specified minimum bounding box and uses the long axis of the area of interest (AOI) to determine the maximum size of the bounding box. The scraper samples each bounding box five times per row and five times per column. The scraper resets when the upper bounds exceeds half the size of the bounding box. This leads to sampling outside the AOI. Within the AOI 94.4 percent of the segments are collected with a bounding box with sides of 0.05 decimal degrees as illustrated in Figure 7. For the MBSNF the script took 3 days per run to collect the full dataset. Each increase in bounding box size shows a halving in run time, the bounding box of 1.6 degrees ran for approximately an hour, 0.8 for 2 hours, 0.4 for 4 hours, and so on.

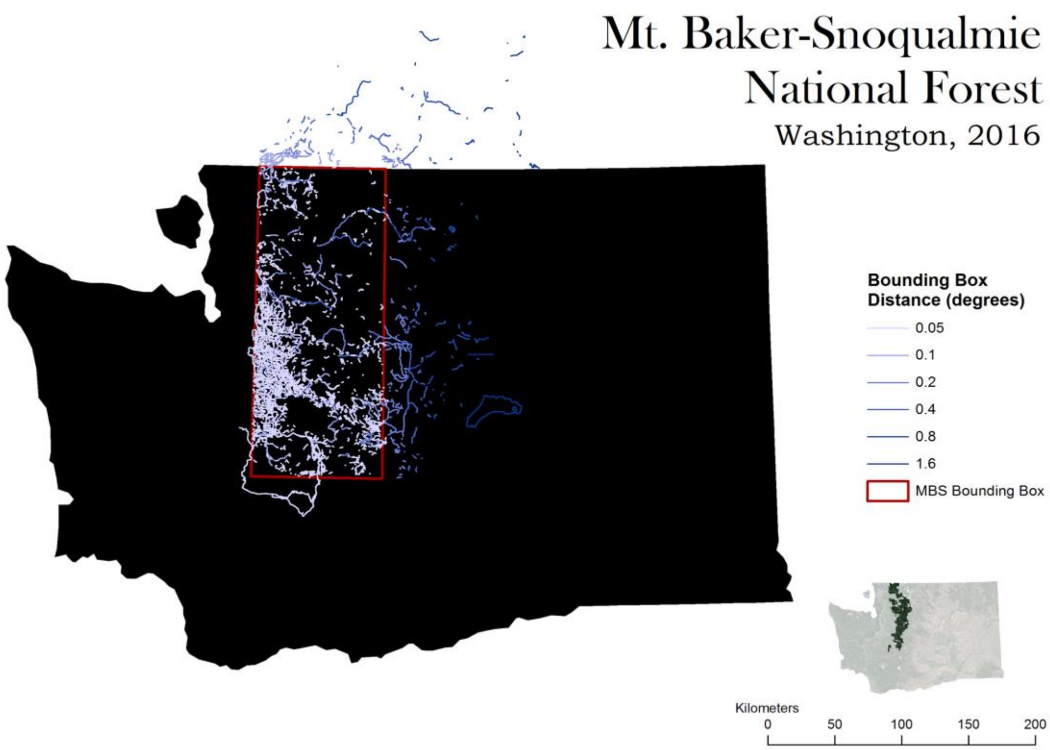


Figure 6. A map showing all collected Strava segments. Lighter colors were collected with smaller bounding boxes.

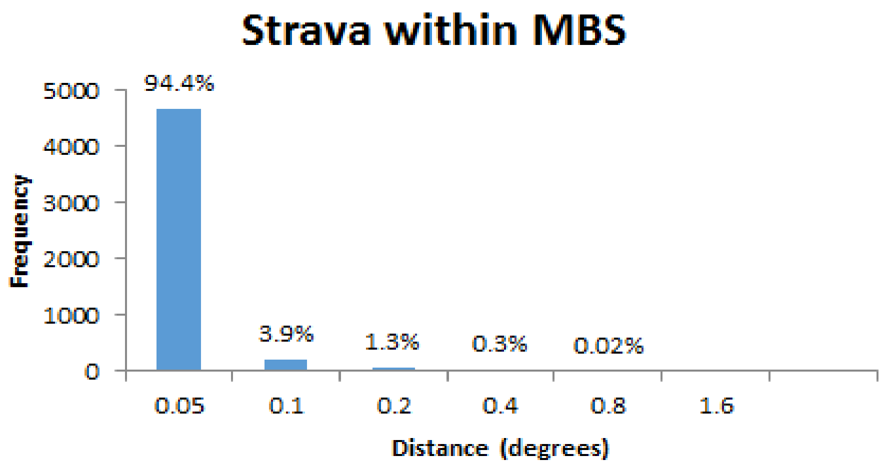


Figure 7. For all segments within the MBSNF, the proportion which were captured with each pass through the AOI.

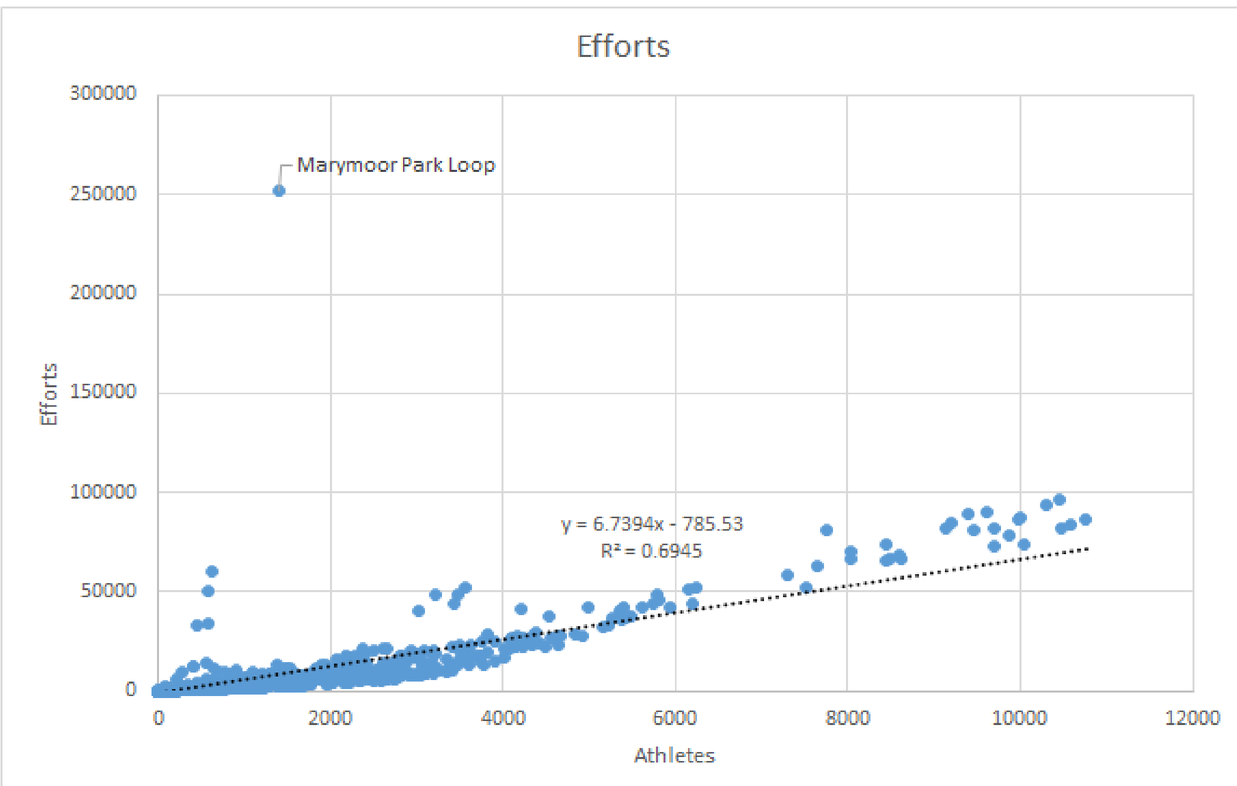


Figure 8. For each segment, the total number of efforts and athletes using the section.

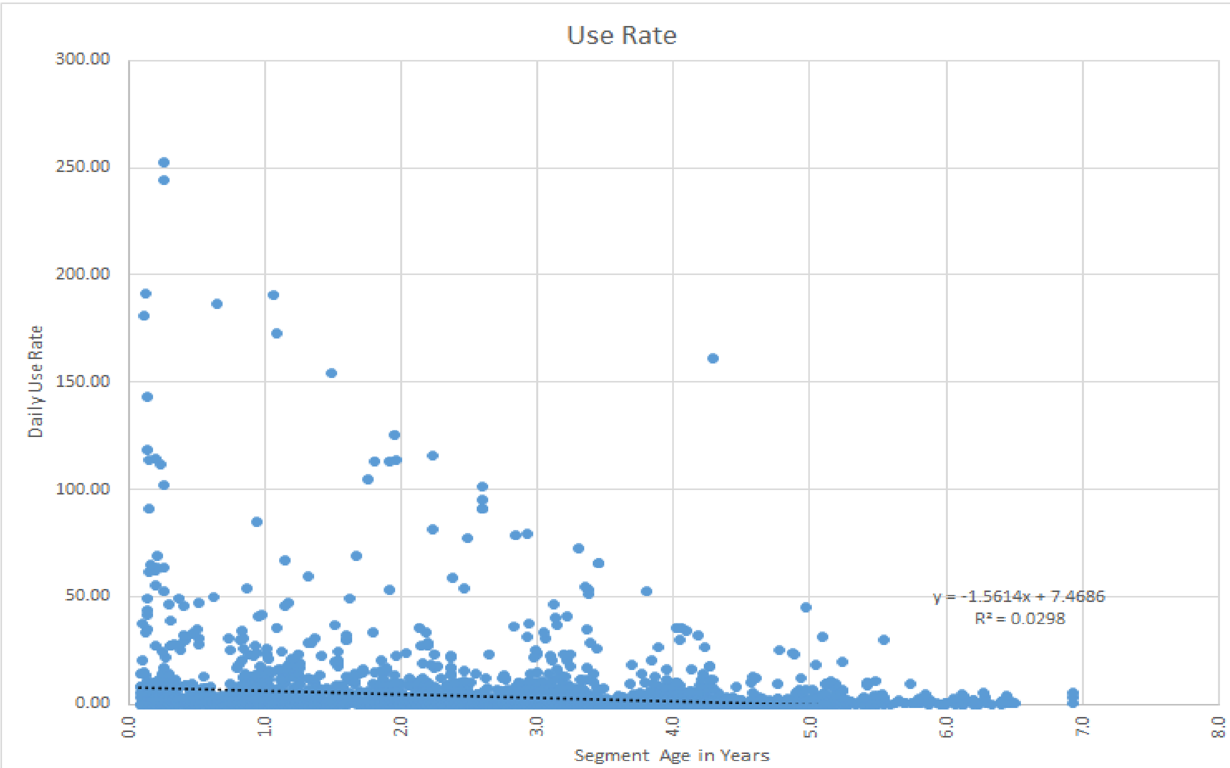


Figure 9. For each segment, the use rate per day relative to the age of the segment.

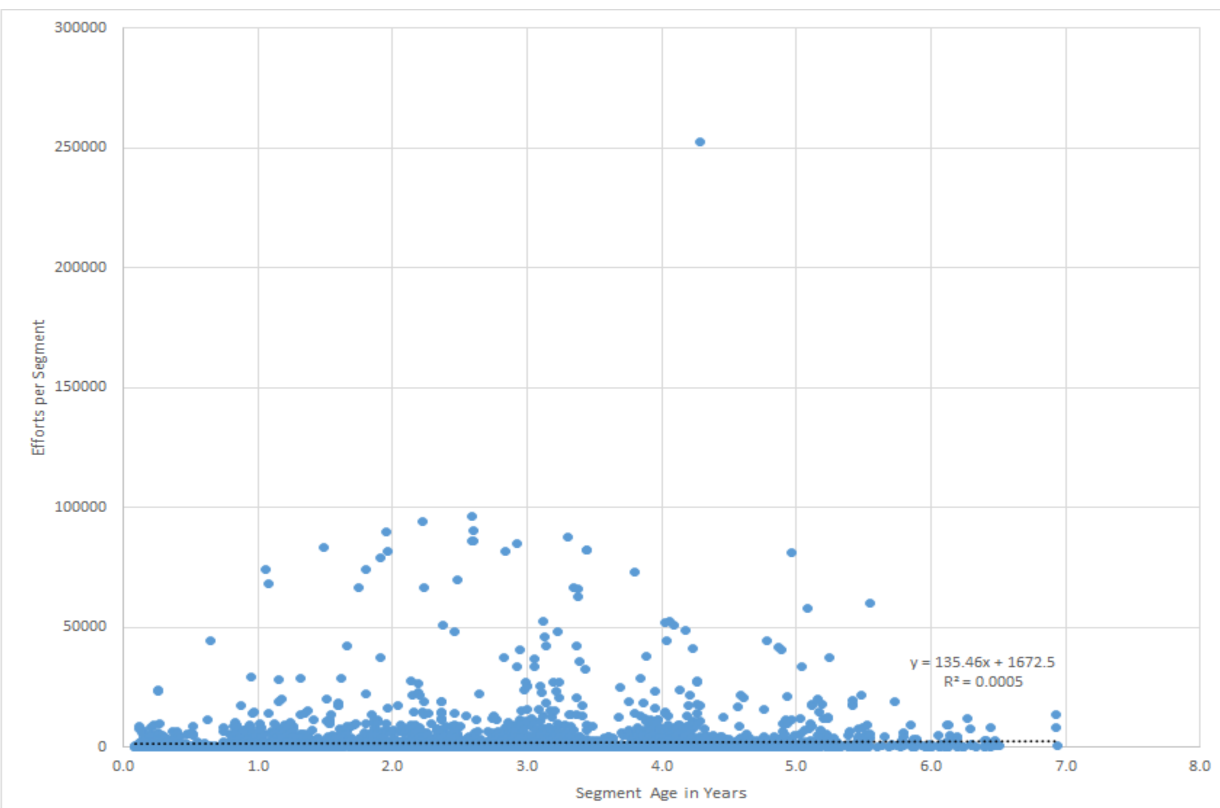


Figure 10. For each segment, the number of efforts relative to the age of the segment.

The number of athletes has a linear relationship with efforts (Figure 8). Figures 9 and 10 show that the age of segment is not a predictor of efforts. The use rate has a weak negative correlation with segment age. The daily use rate is strongly right skewed with many segments with few uses (Figure 11). The segment age shows a normal distribution (Figure 12), with new segment additions declining for the most recent years. Using trail segments as a unit of analysis we see an exponential relationship when the use rate for all segments that intersect the trail buffer are summed (Figure 13).

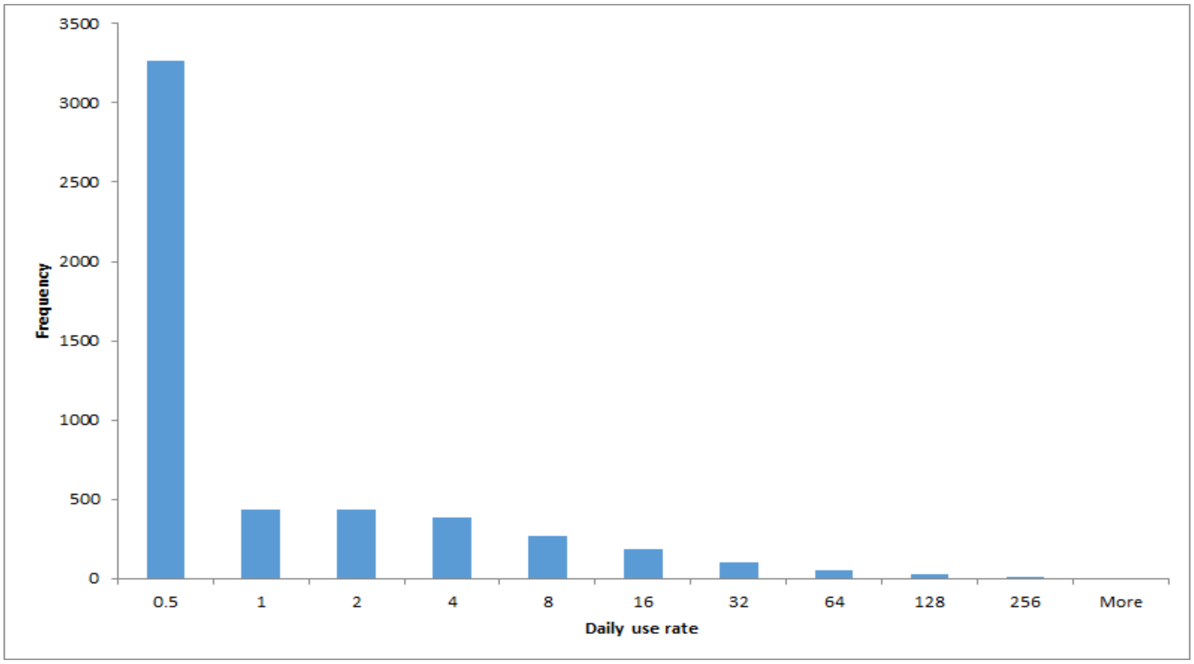


Figure 11. The daily use rate is strongly right skewed, with many segments having less than 1 use per day.

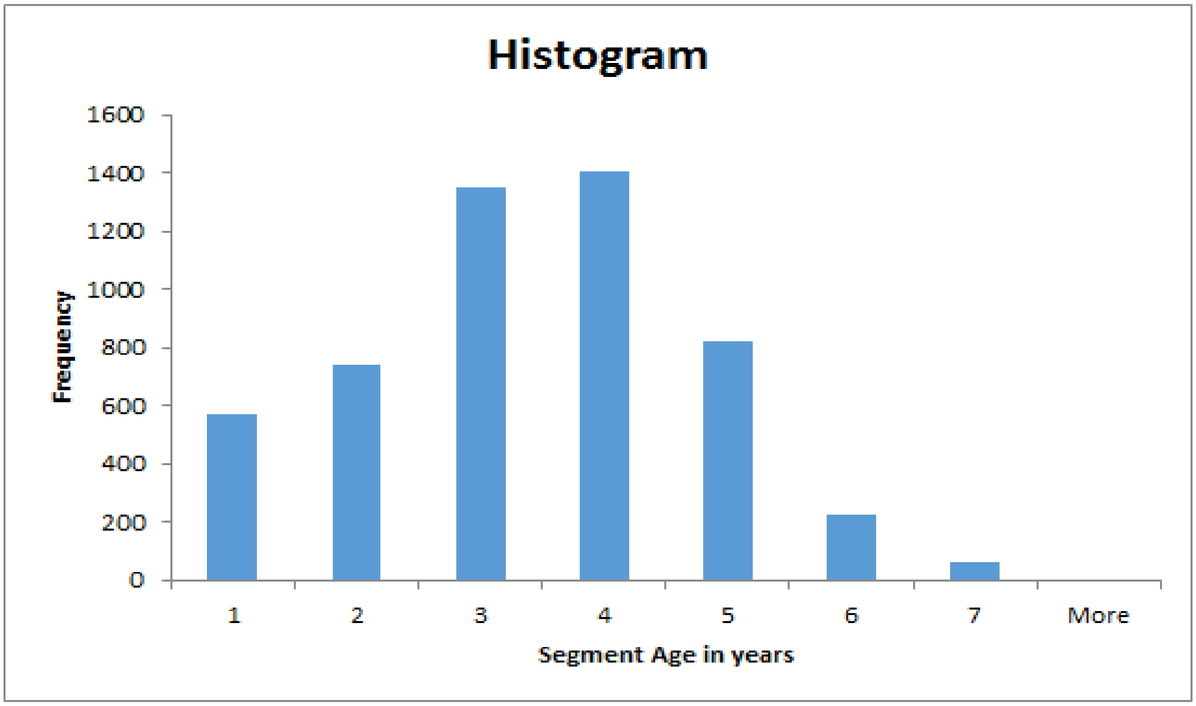


Figure 12. Segment age histogram. The data set is still relatively young.

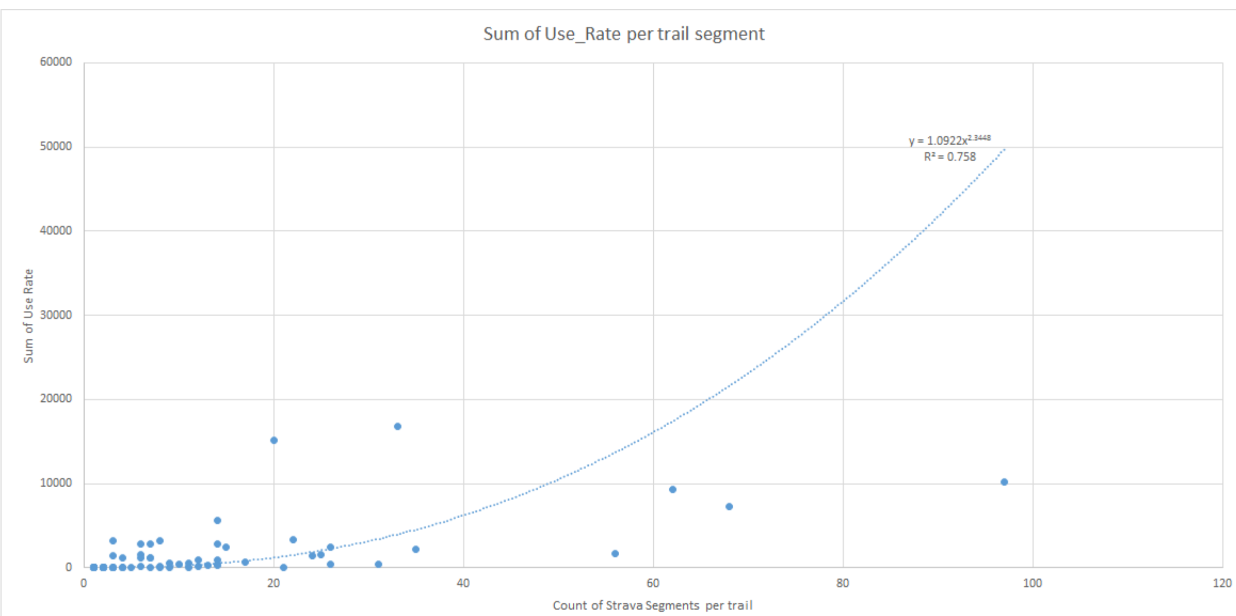


Figure 13. When compared against buffered trail segments we see as the number of segments which cross a trail buffer increases the sum of use rate grows exponentially.

3.2 Distribution of Data

Histograms of use rate shown in Figure 14 counted the frequency of each use rate corresponding to individual trails after log transformation (all values had 1 added to eliminate errors with log(0)). By limiting the data to the 22 trails with Forest Service permit data we see the data distributions shown in Figure 15. The Forest Service use rate estimation shows a rather equal distribution of use across all trails, with about half of trails with one daily permit and half with two. In the case of Flickr, There is a strong right skew, as most trails have a very low rate of use and a few have higher rates going up to 40 daily uses for the pair on the far extreme of the scale. For Strava the distribution is proportionately similar, with most trails having a use rate at or near zero while very few have a higher rate. In the case of Twitter, the majority of trails have a use rate distributed above zero, while the bulk of the remainders are at or near zero. Overall the distribution is skewed in the case of Strava and Flickr, while Twitter has a less extreme skew toward zero and the Forest Service use rates are rather evenly distributed. None of these datasets are normally distributed, but the use of natural log functions are intended to mitigate the skews in distribution.

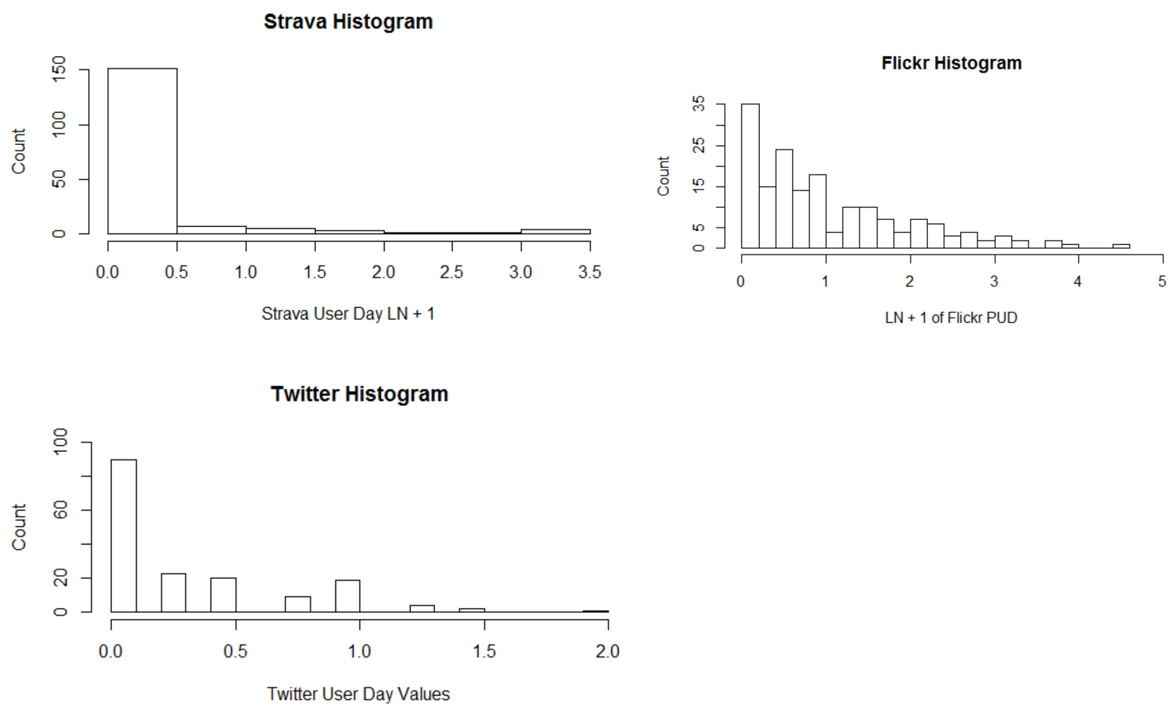


Figure 14. Log transformed histograms for data by trails.

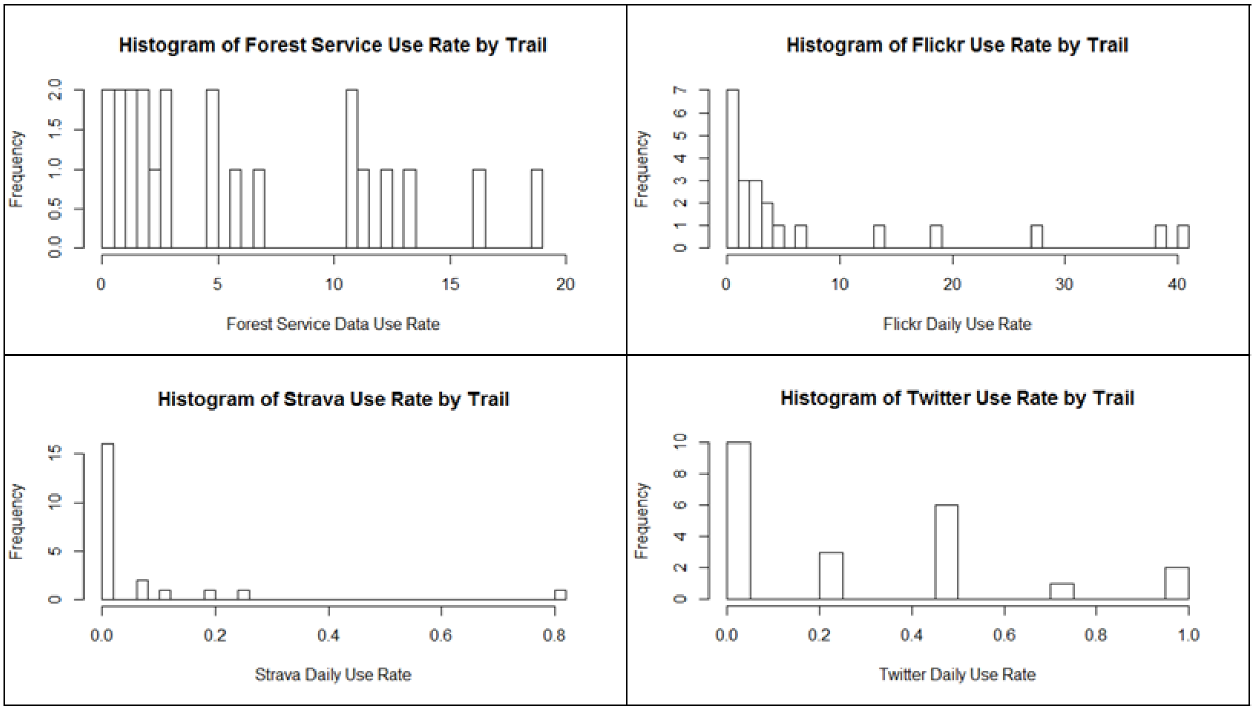


Figure 15. Histograms for trails with Forest Service permit data.

3.3 Correlations and Regression

Table 3. Linear regression summary statistics.

	Coefficients					Residuals					
Data	**term**	**estimate**	**std.error**	**F.stat**	**p.value**	**Df**	**Mult R Sq**	**Adj R Sq**	**std.error**	**F.stat**	**p.value**
FS ~ Flickr	(Intercept	3.268165	0.992565	3.292645	0.003637	20	0.5476	0.525	3.903	24.21	8.26E-05
	Flickr	0.341447	0.069395	4.92034	8.26E-05						
FS ~ Twitter	(Intercept	4.570075	1.608811	2.840654	0.010105	20	0.07326	0.02692	5.587	1.581	0.2231
	Twitter	4.602481	3.660344	1.25739	0.223097						
FS ~ Strava	(Intercept	5.017329	1.211907	4.140028	0.000507	20	0.1718	0.1303	5.282	4.147	0.05516
	Strava	13.12193	6.443281	2.036529	0.055155						
FS ~ Multiple	(Intercept	3.813385	1.167571	3.266085	0.004291	18	0.5699	0.4983	4.012	7.951	0.001388
	Strava	-0.95987	6.03298	-0.1591	0.875359						
	Twitter	-2.97148	3.109359	-0.95566	0.351905						
	Flickr	0.392691	0.099135	3.961161	0.000916						

Considering the variables in Table 3.3.1 showing single and multiple regression outputs, Flickr has the only coefficient that is statistically significant. Strava's coefficient is significant at the 0.90 confidence level. In the case of the multiple regression, Flickr again has the only statistically significant coefficient. The residuals show that Flickr also performs best at describing variation with an adjusted R squared value of .525 that is also statistically significant.

Table 4. Paired IV linear regression summary statistics.

	Coefficients					Residuals					
Data	**term**	**estimate**	**std.error**	**statistic**	**p.value**	**Df**	**Mult R Sq**	**Adj R Sq**	**std.error**	**F.stat**	**p.value**
FS ~ Strava + Flickr	(Intercept)	3.270905	1.01796	3.213196	0.004577	19	0.5481	0.5005	4.003	11.52	0.000528
	Strava	-0.87454	6.018537	-0.14531	0.885998						
	Flickr	0.348897	0.087707	3.97798	0.000806						
FS ~ Twitter + Strava	(Intercept)	4.319601	1.545414	2.795109	0.011546	19	0.195	0.1103	5.342	2.302	0.1273
	Strava	11.59078	6.83651	1.695424	0.106321						
	Twitter	2.721947	3.671552	0.741362	0.467539						
FS ~ Twitter + Flickr	(Intercept)	3.809042	1.136918	3.350323	0.003361	19	0.5693	0.524	3.907	12.56	0.000335
	Twitter	-2.96416	3.028223	-0.97884	0.339957						
	Flickr	0.384408	0.082171	4.678159	0.000164						

The paired regressions in Table 4 show that a the Flickr data has the highest adjusted R squared value at 0.525. The multiple regression featuring all three variables does not perform as well with an R squared value of .4983. Strava and Flickr together come close to showing statistical significance but fail to meet the requirement of a .95 confidence level when examining the p value.

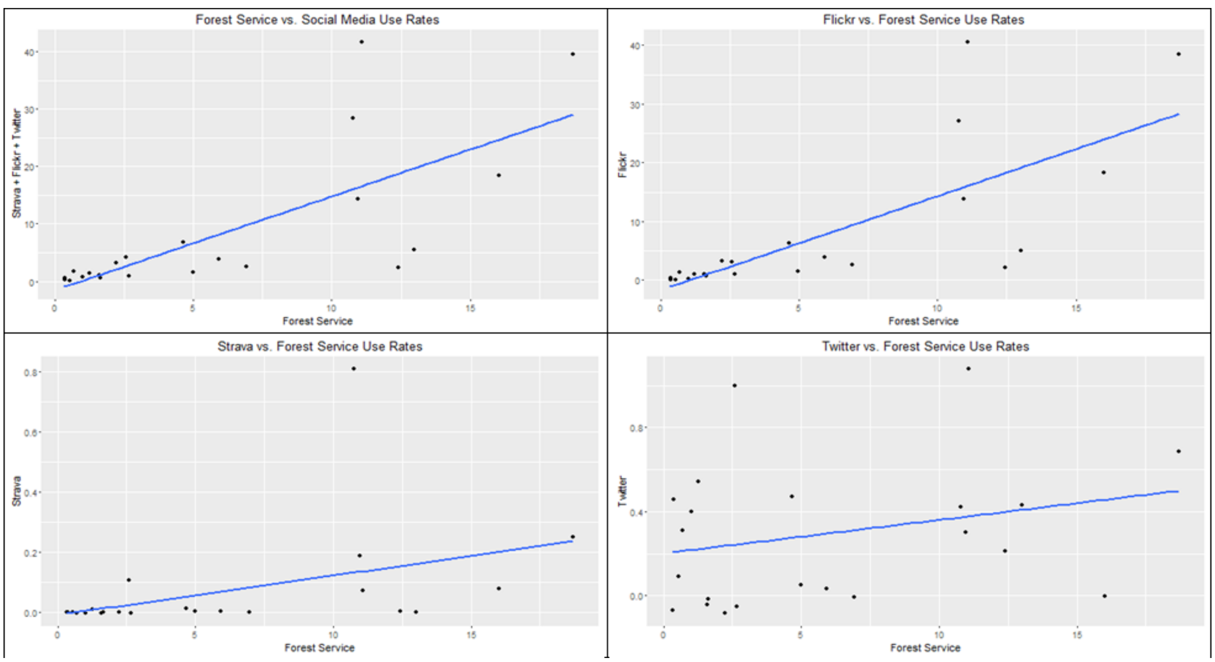


Figure 16. Linear regression plotted against xy scatter plot.

The linear models featured in Figure 16 show the general relationship between Forest Service permit data as a dependent variable and Flickr, Twitter, and Strava data as independent variables. The multiple regression and Flickr models in the top row are very similar, while the Strava and Twitter data show a lesser degree of clustering around the predictor line. Except for Twitter, all the models show more clustering closer to zero values.

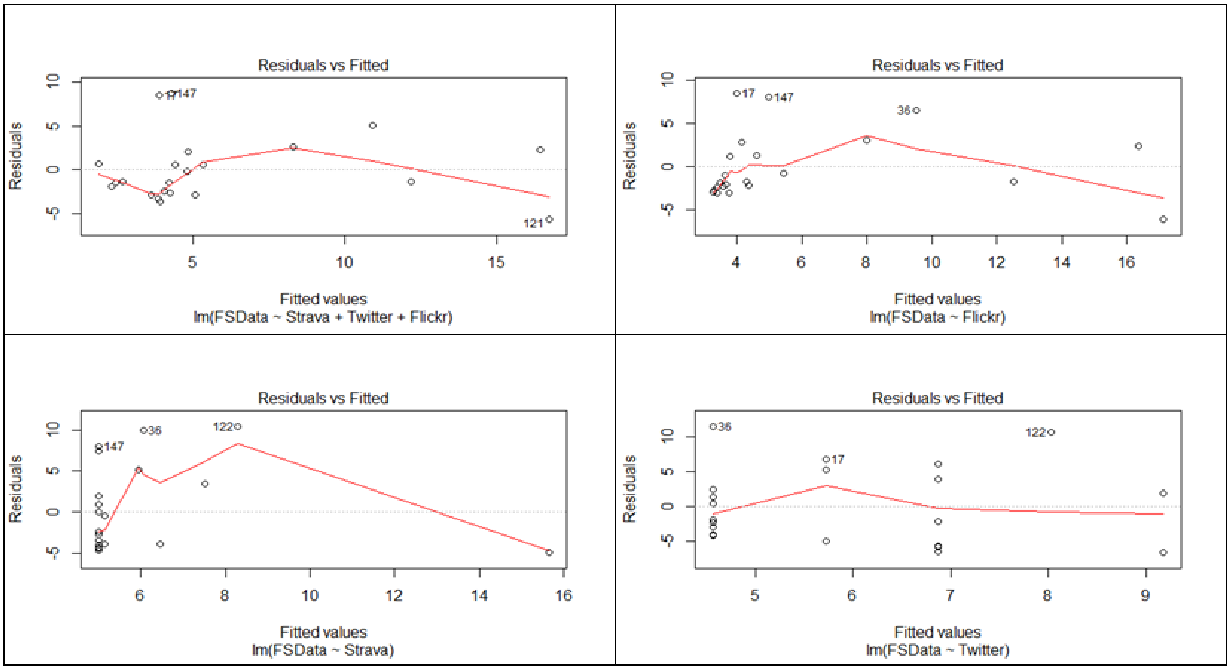


Figure 17. Residuals plotted against values.

Figure 17 demonstrates the relationship between the residuals and the fitted values. The multiple regression, Flickr, and Twitter models all appear to be homoscedastic, and mostly unbiased with their data clustering near zero. These models all have more variation with higher values, and less variation closer to zero. In the case of Strava, the model is heteroscedastic and biased.

Table 5. Spearman's correlation by trail segment.

	FS	Strava	Twitter	Flickr
FS	1	0.441	0.206	0.845
Strava	0.441	1	0.350	0.423
Twitter	0.206	0.350	1	0.584
Flickr	0.845	0.423	0.584	1

Table 6. Spearman's correlation by 500m x 500m cell.

	Strava	Twitter	Flickr
Strava	1	0.244	0.292
Twitter	0.244	1	0.287
Flickr	0.292	0.287	1

The Spearman's correlation between the dependent variable (FS) and the independent variables (Strava, Twitter, and Flickr) show the highest correlation between Flickr and FS data. None of the IVs have a strong correlation with each other in either the Spearman's correlation by trail or by cell (Tables 5 and 6).

3.4 Spatial Regression

An additional measure of analysis was to use a 500m x 500m cell grid across the AOI for which a Flickr photo user day, Twitter user day, and a Strava average use were calculated.

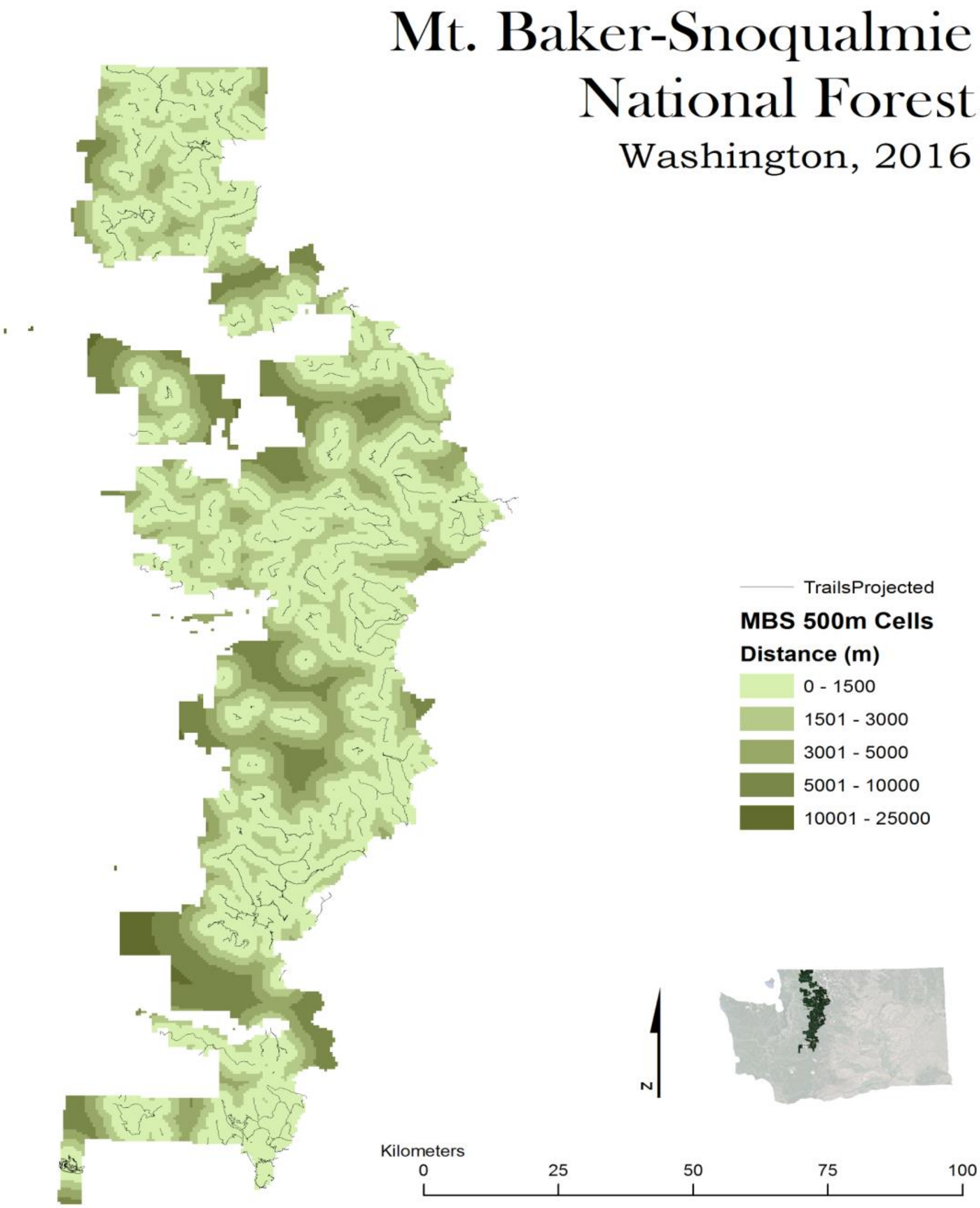


Figure 18. Distance to trail segment by cell.

To perform a regression, we used the continuous variable distance to the nearest trail shown in Figure 18 as the dependent variable. The raw data for Strava, Flickr, and Twitter and user days by cell can be seen in Figures 19-21. The raw data by trail can be seen in Appendix G.

3.4.1. Strava

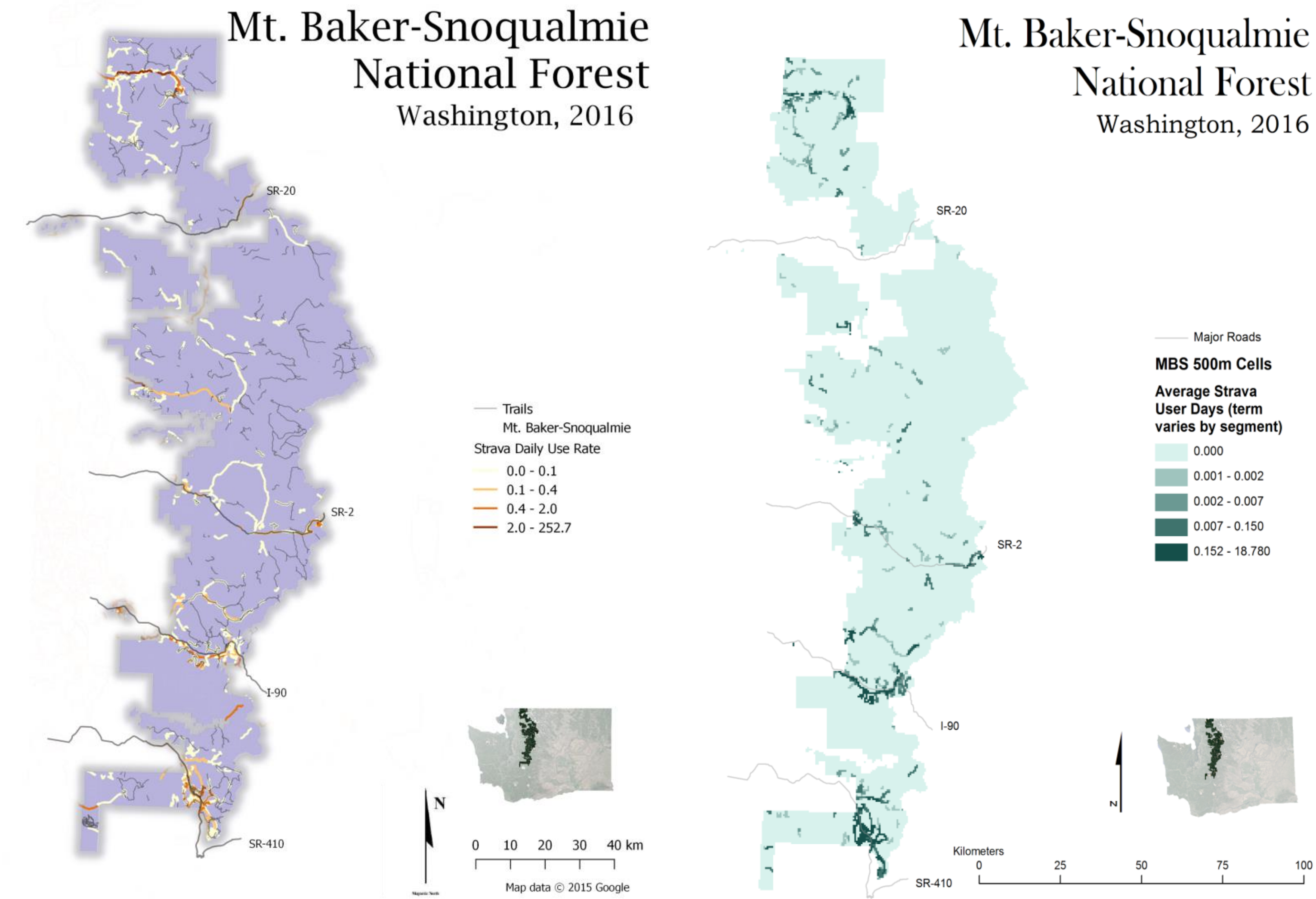


Figure 19. Left. Strava segments daily use rate. Right. Average daily use rate by cell.

3.4.2. Flickr

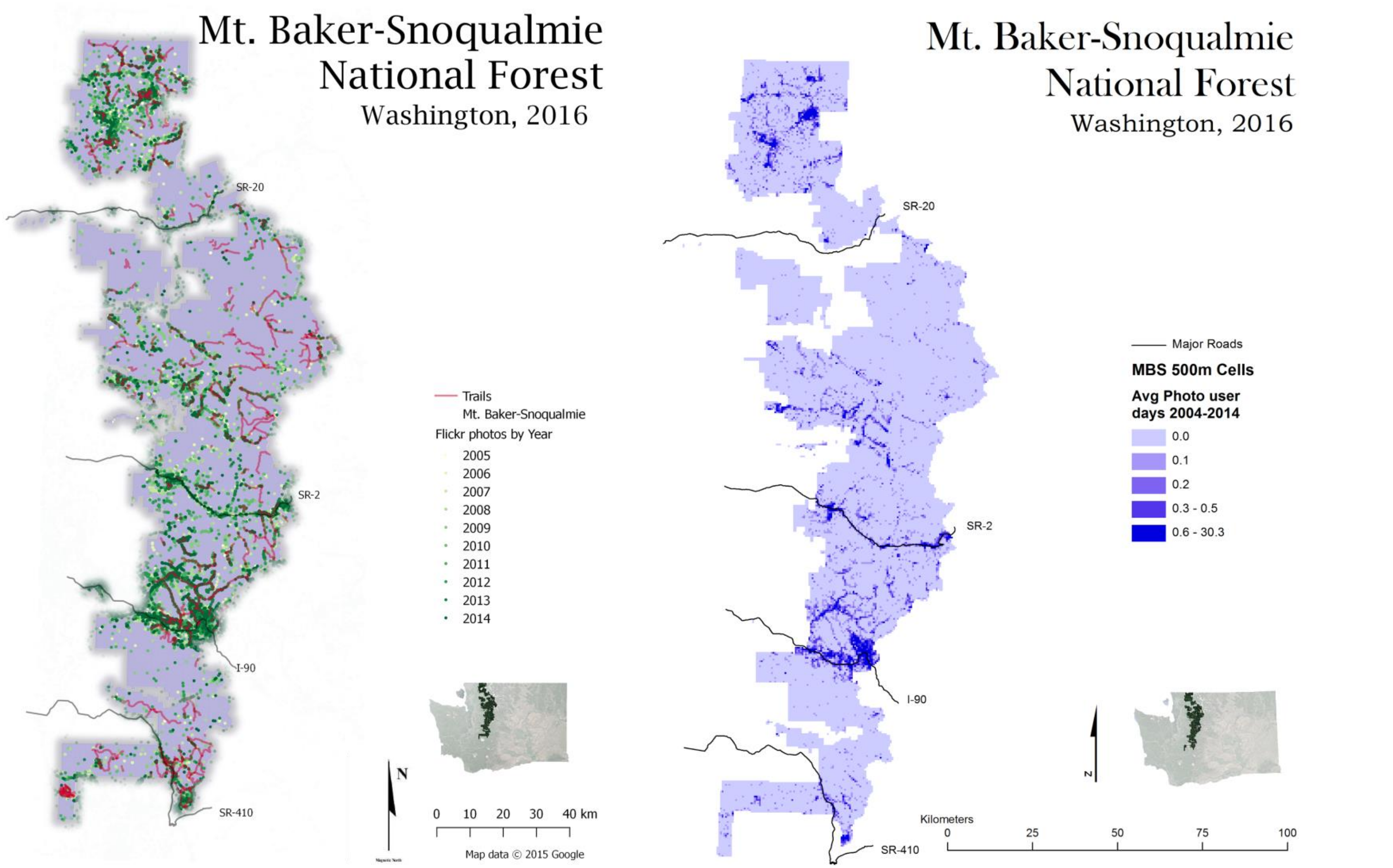


Figure 20. Left. Raw Flickr photos by year. Right. Average PUD per cell.

3.4.3. Twitter

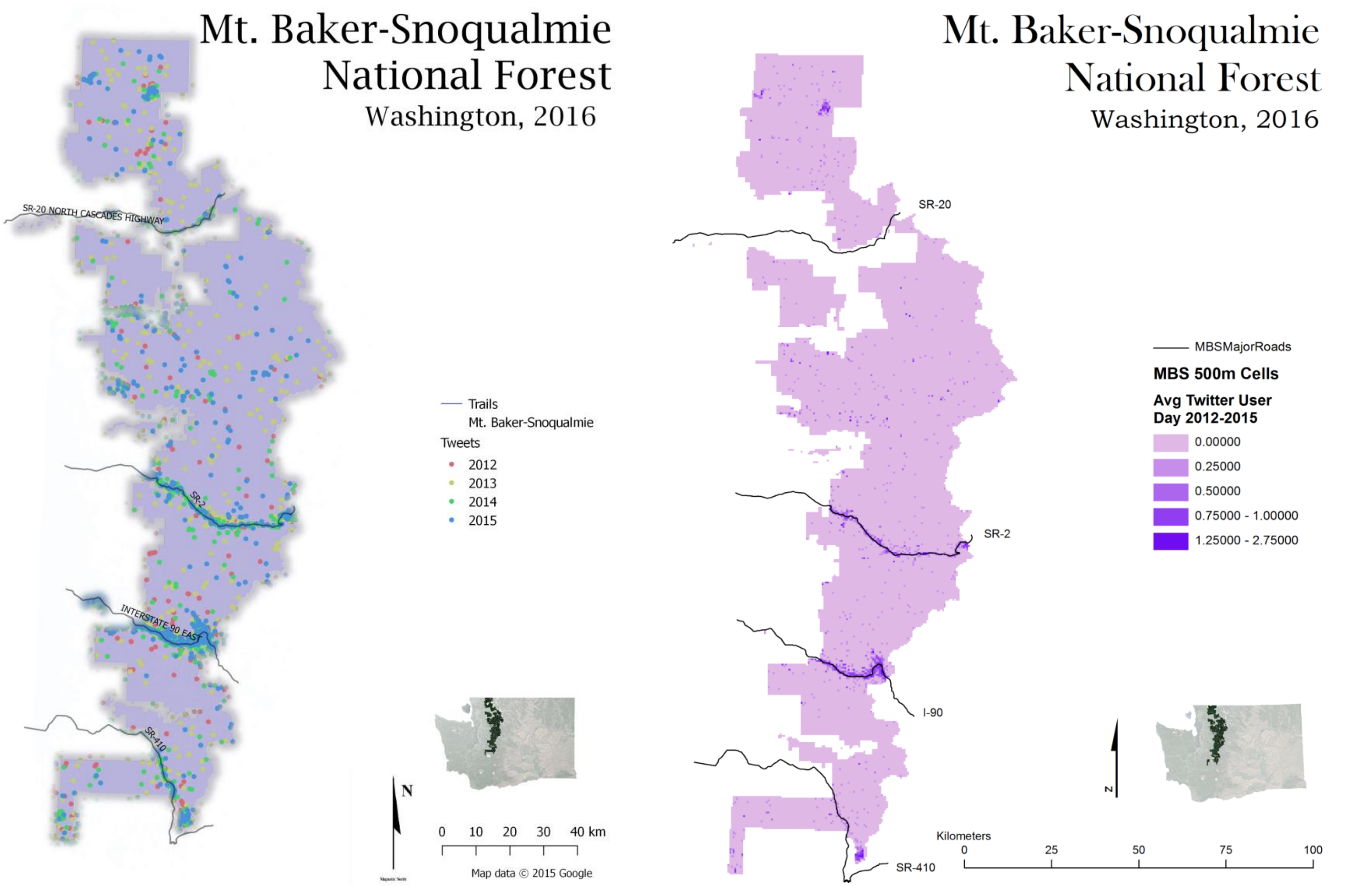


Figure 21. Left. Raw Twitter “Tweets” by year. Right. Average TUD per cell.

3.4.4 Getis-Ord Gi*

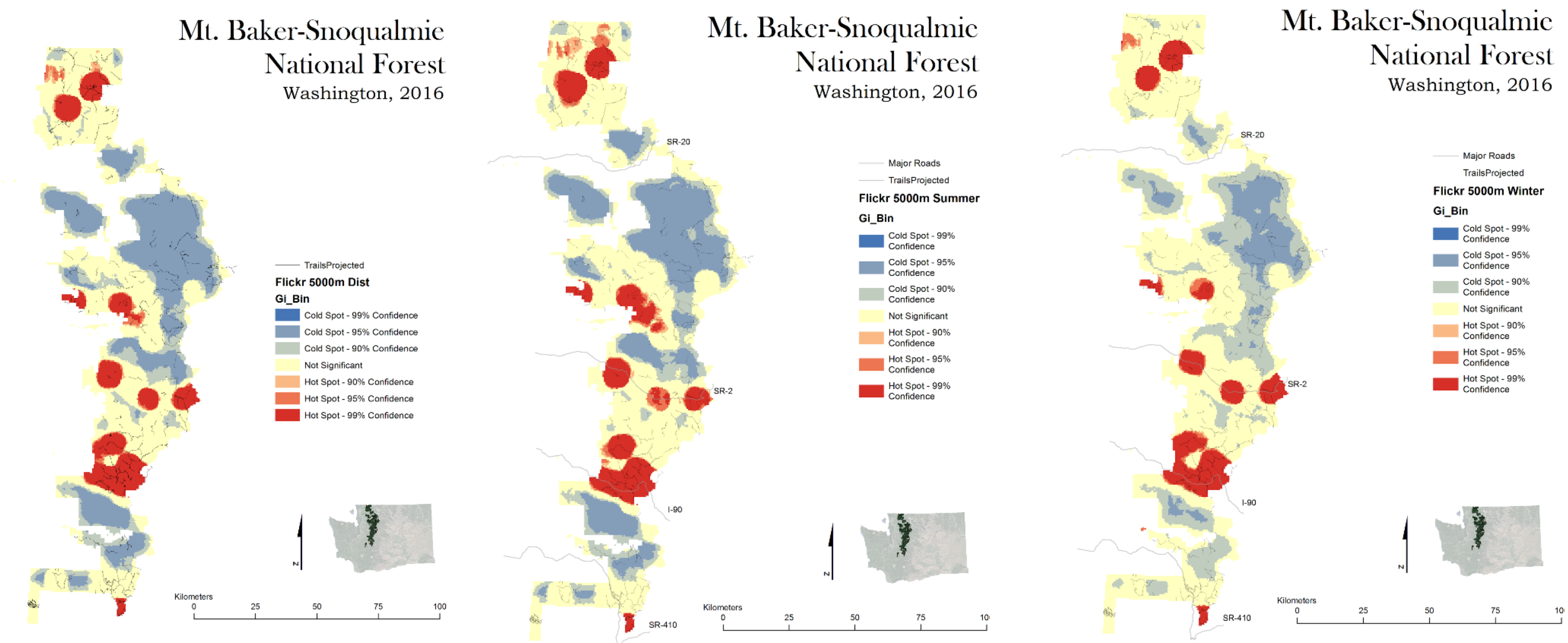


Figure 22. Getis-Ord Gi* hotspot analysis for a 5000m distance band. From left to right: Flickr average, Flickr Summer, Flickr Winter.

The hot spot analysis for Flickr for annual average data and for seasonal data is shown in Figure 22. The seasonal changes show a higher intensity in the summer months than the winter but no strong spatial variation. The hot spot analysis for Strava and Twitter are shown in Figure 23. Strava has a strong clustering of high values around recreational use areas. Twitter shows hot spots both around recreational use areas as well as major highways through MBSNF.

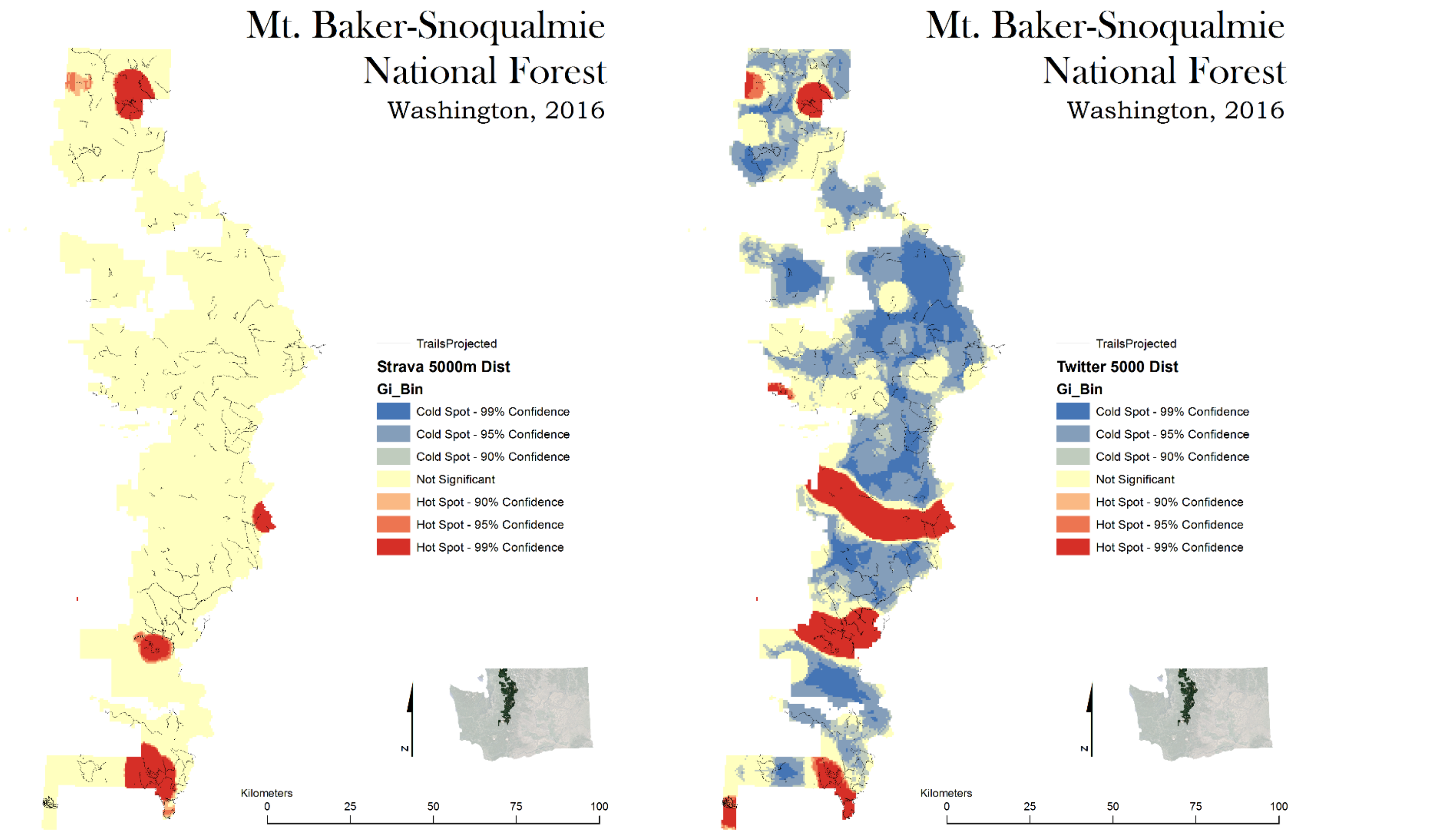


Figure 23. Getis-Ord Gi* hotspot analysis for a 5000m distance band. Left: Strava. Right: Twitter.

3.4.5 Geographically Weighted Regression

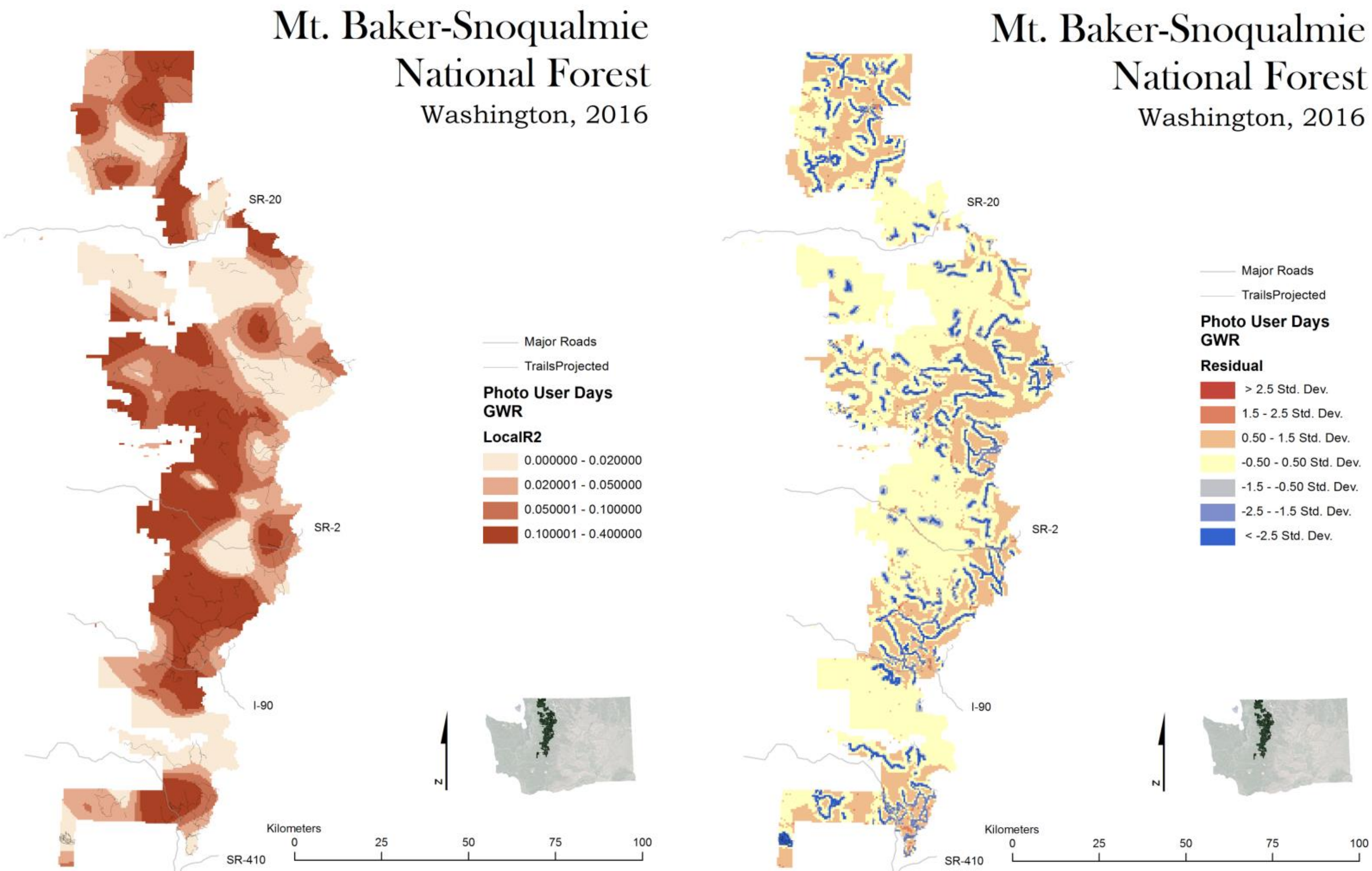


Figure 24. Geographically weighted regression for the Flicker dataset by cell. Left: Local R^2 values. Right: Residuals show underestimation near trails and overestimation away from trails.

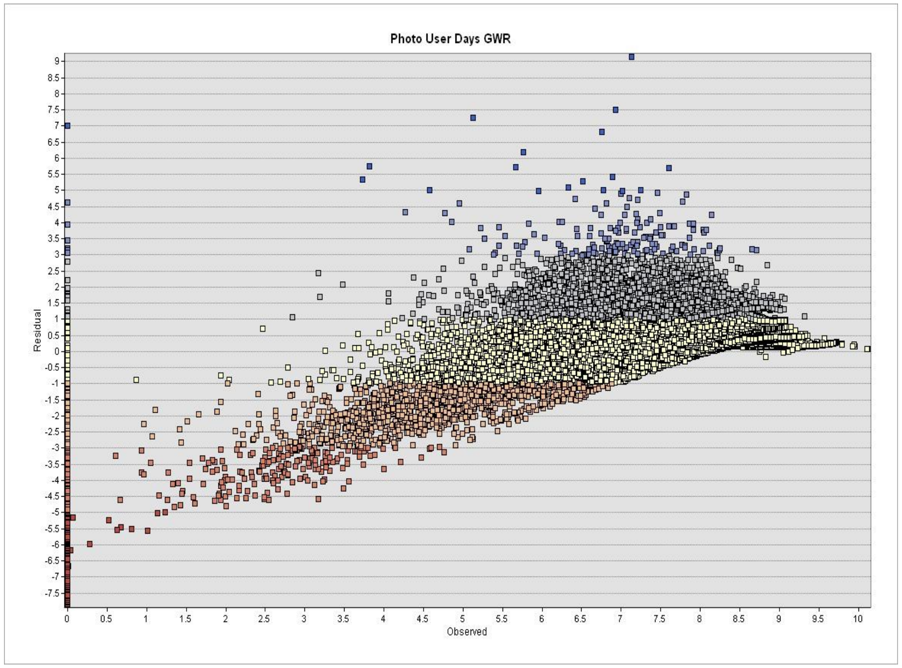


Figure 25. Geographically weighted regression residuals show significant linearity. Heteroscedastic results indicate modeling errors.

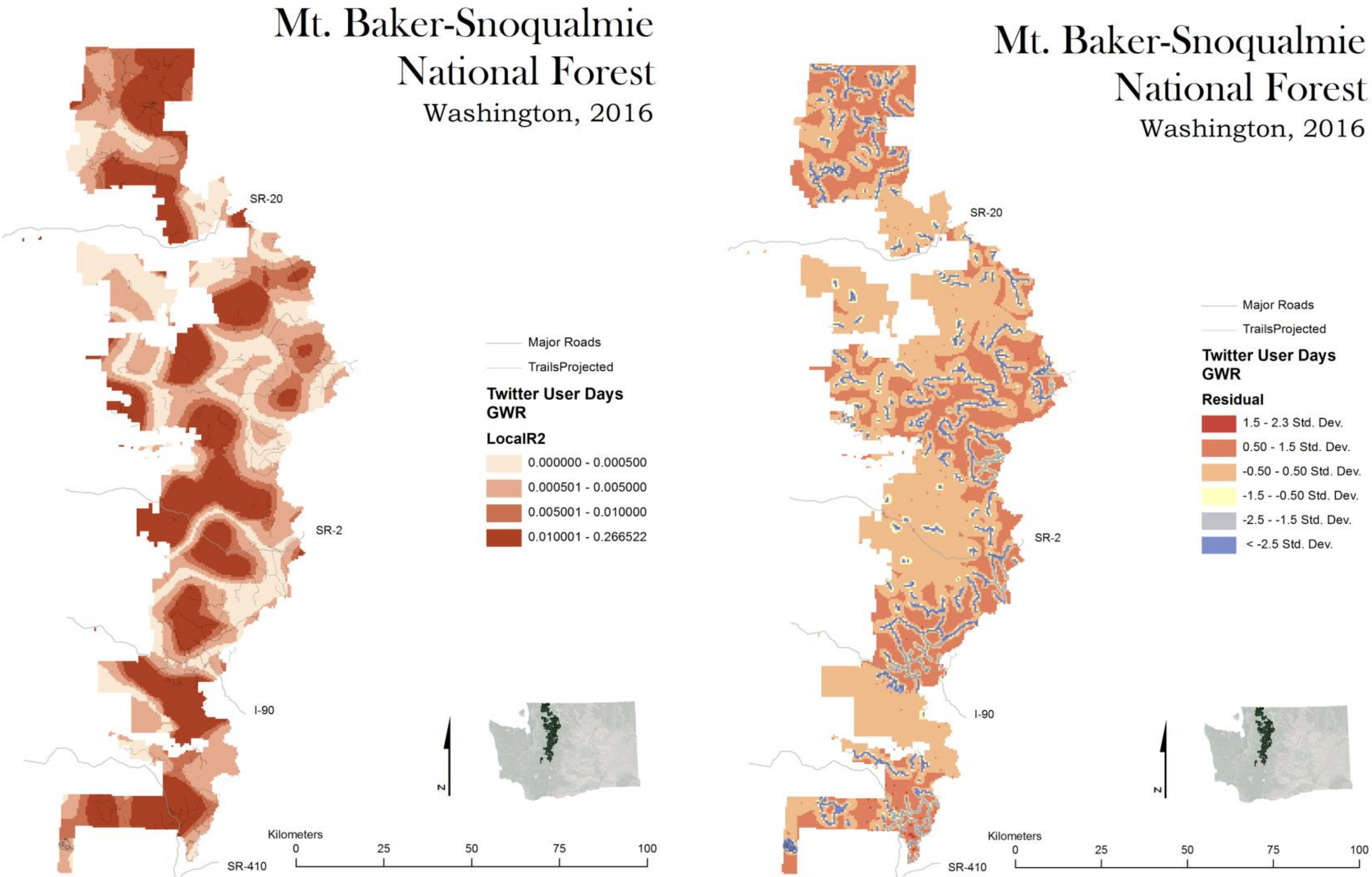


Figure 26. Geographically weighted regression. Left: Local R^2 values. Right: Residuals show underestimation near trails and overestimation away from trails.

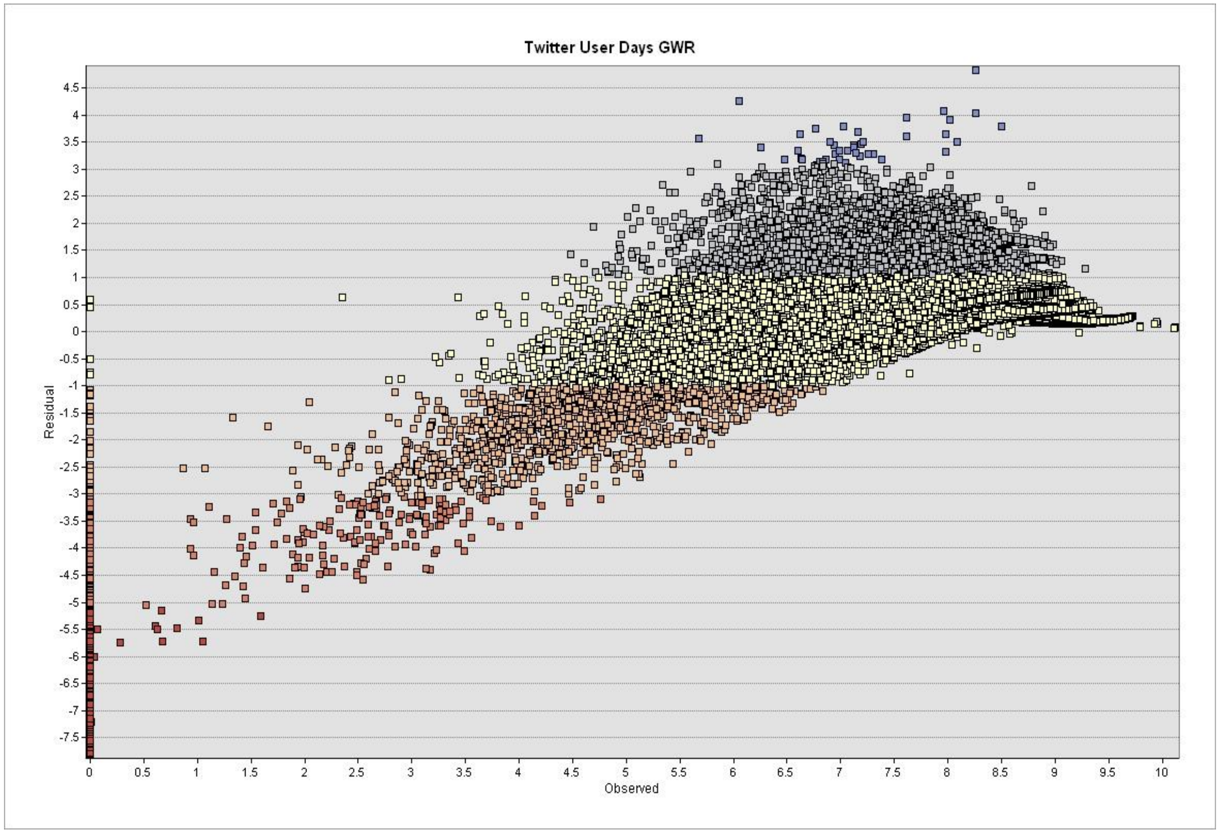


Figure 27. Geographically weighted regression residuals show significant linearity. Heteroscedastic results indicate modeling errors.

The geographically weighted regressions for Flickr (Figure 24) and Twitter (Figure 26) show no obvious pattern associated with trails or recreational use areas. The mapped residuals, show an underestimation of values near to trails and overestimation of values more distant from trails. For both the xy scatter plots of residuals show strong linearity (Figures 25 and 27) suggesting significant issues with a linear model.

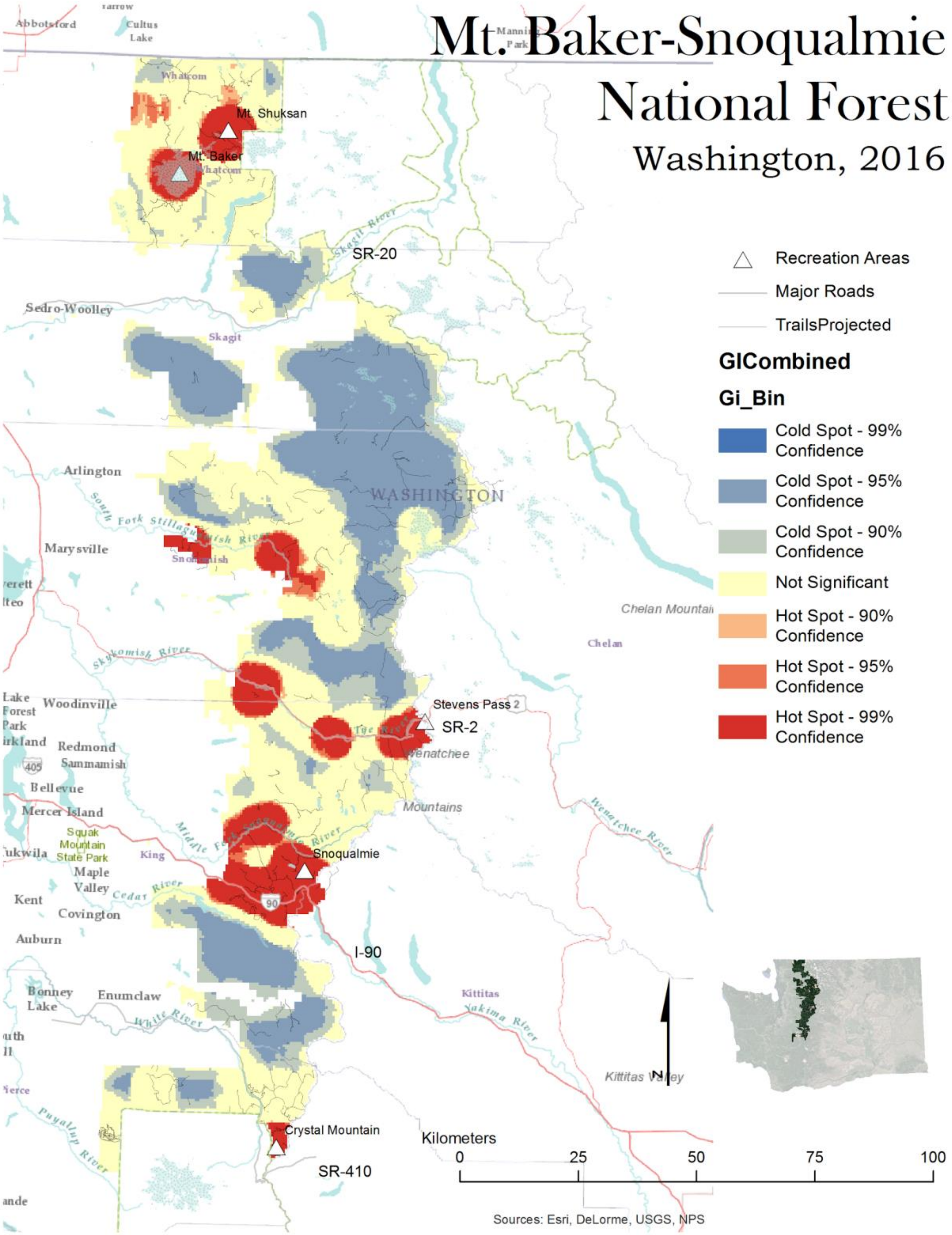


Figure 28. Summed daily use rates by cell. Getis-Ord Gi* analysis with a 5000m distance band. Hot spots correlate with major mountain recreation areas.

4. Discussion

4.1 Analysis of Results

4.1.1 Buffer Distance

The standard distance buffer performed adequately for the task at hand. The trail data matched reasonably and although the area in difference was approximately equivalent to the area in common the overlap was similar and the number of photos inappropriately captured by overlap was small. A standard buffer distance is an appropriate method for a non-expert to aggregate photos by trail segment for the InVEST tool kit.

4.1.2 Strava Data collection

API Limits

Overall, we are confident the script will collect the necessary data. The script in its current format does have several weaknesses worth noting. The script uses decimal degrees in order to comply with the needs of the Strava API. The segment explorer returns a maximum of ten segments per bounding box. The segment explorer also use decimal degrees as coordinate input in order to create a bounding box. Using decimal degrees means that for geographic areas which span more than two degrees of latitude, there will be a significant difference in the width of the segment explorer cells. The scraper uses a standard square cell, which expands by doubling in size until it is larger than the longest side of the bounding box for the AOI. Bounding boxes which have one side significantly longer than the other will find that the segment explorer will search significantly outside of the bounds of the AOI. Finally, non-square or non-rectangular AOI searches will still grid the area of the entire bounding box, which may add significantly to the time required to pull the entire dataset.

Usefulness of Data

The Strava data set shows some expected distributions. Efforts increase as the number of athletes increase, with the notable exception of the Marymoor race track. Use rate is highly right skewed with few high use segments and many segments with few uses. It is unexpected that efforts per segment are almost unrelated to the age of the segment. This suggests no linear relationship will be found for use from the existing data, because recorded segments are semi-random. The histogram of segment age (Figure 12) shows a semi-normal distribution. The interpretation of segment age would be that an initial burst of segments with the creation of the platform has not had time to develop into a highly left skewed data set as the segment collection approaches ‘completion’.

4.1.3 Regression analysis

All data are non-normally distributed and therefore not ideal candidates for regression. Even log transformed data are still right skewed. The regression analysis showed Flickr is the best predictor of the available Forest Service permit data. Flickr alone provides more explanatory power and has an adjusted r squared value of 0.525. All of the regressions are limited by the small sample size of Forest Service permit data and the model would benefit from additional permit data.

4.1.4 Getis Ord Gi*

The analysis suggests that the Strava, Flickr, and Twitter data sets have high values, highly clustered around the major mountain recreation areas. Twitter clustering corresponds to major highways as well. This suggests to us that Twitter use is correlated with driving, likely as passengers share their experience. Strava use shows the clustering limited tightly to Mountain recreation areas within MBS. Flickr shows additional hot spots surrounding the towns of Index and Skykomish.

4.1.5 Geographically Weighted Regression

All of the regression models had significant defects. All of the data sets were non-normal distributions. The residuals were heteroskedastic for all models. The geographic regressions for Twitter and Flickr shows values underpredicted near trails and overpredicted distant from trails. The relationship between the social media and distance to trails is likely non-linear, and an exponential decay model may fit more appropriately.

4.1.6 Potential sources of error

While the goal of this study was to compare visitation rates derived from geotagged social media to traditional usage data collected by the Forest Service, the limited availability of Forest Service data confers limited usefulness to the model. It was hoped that NVUM would provide an added data source, however the NVUM documentation clearly states it would not be appropriate for small scale analysis.

One of the analyses was buffer size in an effort to limit the area of overlap. An additional measure was to create a consolidated trail data set, provided by the Natural Capital Project. Grouping trails is an effective way to minimize this bias. In our sample the buffer intersections contain 16,103 of 302,881 Flickr photos (not PUD) and accounts for 237.76 square kilometers of a total 1693.6 square kilometers. Despite this there is still the opportunity for inaccurately geotagged photos to be outside of the buffered area. Figure 29 shows the distribution of raw Flickr photos for reference, multiple buffers capture the red points, single buffers capture the green points, and the purple points are not within the trail buffers. A balance must be struck when selecting buffer size. Red points show photos captured by multiple buffers which can lead to overestimates of use rates. Photo points shown in purple are not captured by the 500m buffer. The points shown in purple may either be appropriately outside the buffer area, or accidentally excluded through inappropriate geotag or inappropriate characterization of the trail through standard buffers.

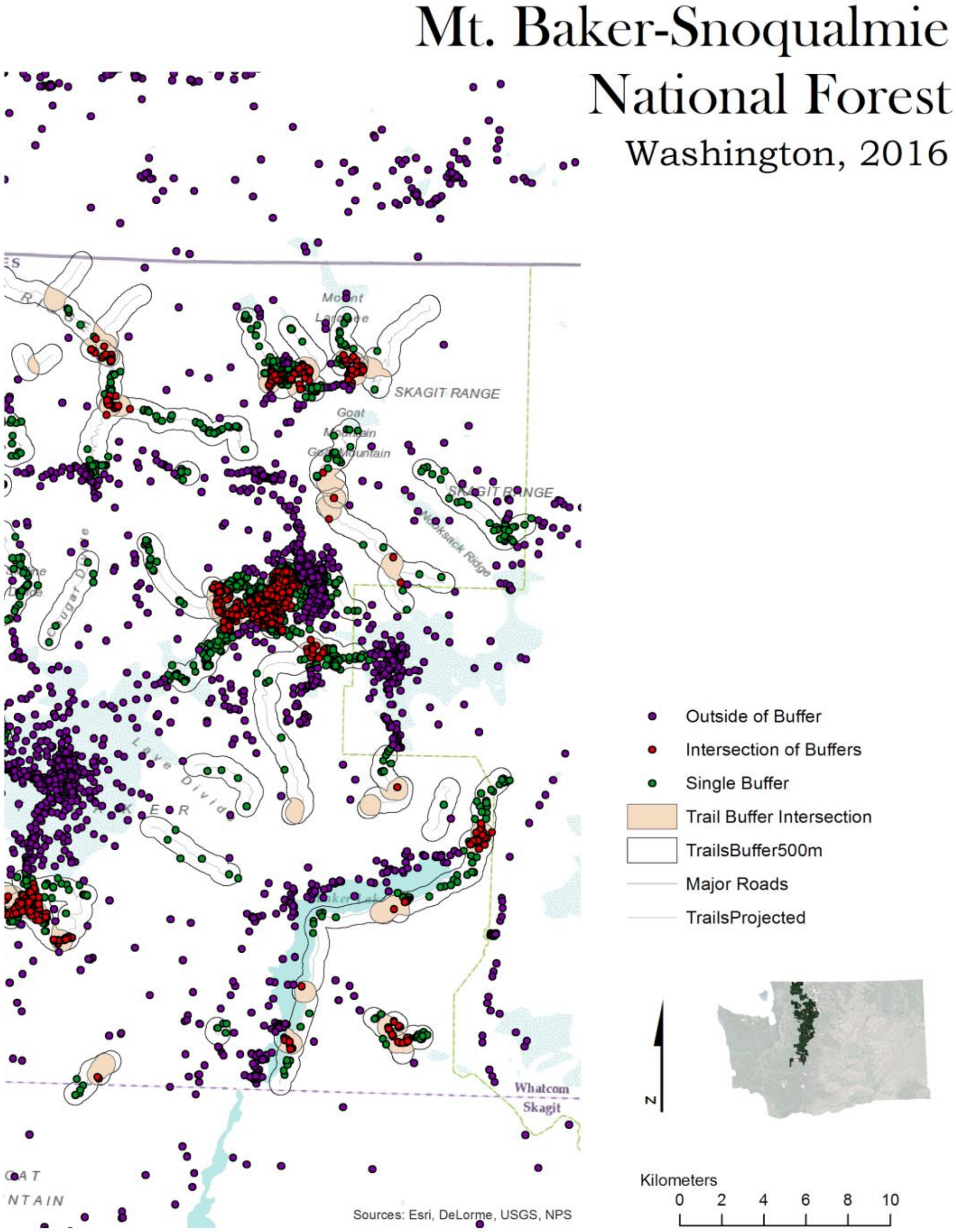


Figure 29. Buffer analysis sources of error. Repeated use counts resulting from overlapping buffers are shown in red. Photos outside of the trail buffers are shown in purple.

Finally, Strava segments may cross the buffer only at the trailhead, and not be appropriately excluded from the use rate sampling. This could be illustrated as a road segment, which passes a trailhead. The road would be tangent to the trail in question.

4.2 Summary

The weak correlations between explanatory variables suggests they have the opportunity to provide different contributions to a regression. The hot spot analysis shows unique distribution between the Flickr and the Twitter data. The multiple regression also suggests the Twitter and Flickr data provide the most parsimonious model. However, with the limited Forest Service data, the regressions predictive power is severely limited. If Strava continues to gain users and data on segment use it may develop into a valuable source of data. The data currently available from Strava does not provide a valuable addition to predicting global or local trail use relative to Flickr data and based on the available permit data for MBSNF.

This analysis found a significant geographic component to the data. Hotspot analysis showed strong clustering of values. The regression residuals showed the linear model underestimates use around trails and overestimates the surrounding areas. A non-linear model would be needed to predict trail usage where there is no data.

4.2.1 Social media demographics

The Pew research center lists demographics of social media users as illustrated in Figure 30. (http://www.pewinternet.org/2015/08/19/the-demographics-of-social-media-users/). Social media use has been shown to correlate with higher education, higher incomes, and age between 25 and 44 (Li et al. 2013). We can see wide variation from platform to platform. The age of social media users is clustered between 25 and 55 (Figure 31) while MBSNF users have higher proportions of very young and very old represented (Figure 32). In general we see more female users for the platforms of interest, Flickr and Twitter (Figure 33) than we see represented in the NVUM data for MBSNF (Figure 34). We see the white population dominating MBSNF use in NVUM data, while other populations would be over represented in social media (Figure 35). In MBS, 57.7% of users report participating in hiking/walking and 33.4% of users report it as their main activity (Figure 36). This use would be consistent with trail use, relevant to the research question. From these comparisons we can see men are more dominant in the MBS sample than the social media sample. The age distribution shows that relative to MBS usage, the youngest and oldest park users are probably underrepresented in social media, and teenagers are likely to be overrepresented.

Facebook Demographics

Among internet users, the % who use Facebook

	Internet users
Total	*72%*
Men	66
Women	77
White, Non-Hispanic	70
Black, Non-Hispanic (n=85)	67
Hispanic	75
18-29	82
30-49	79
50-64	64
65+	48
High school grad or less	71
Some college	72
College+	72
Less than $30,000/yr	73
$30,000-$49,999	72
$50,000-$74,999	66
$75,000+	78
Urban	74
Suburban	72
Rural	67

Source: Pew Research Center, March 17-April 12, 2015.

PEW RESEARCH CENTER

Instagram Demographics

Among internet users, the % who use Instagram

	Internet users
Total	*28%*
Men	24
Women	31
White, Non-Hispanic	21
Black, Non-Hispanic (n=85)	47
Hispanic	38
18-29	55
30-49	28
50-64	11
65+	4
High school grad or less	25
Some college	32
College+	26
Less than $30,000/yr	26
$30,000-$49,999	27
$50,000-$74,999	30
$75,000+	26
Urban	32
Suburban	28
Rural	18

Source: Pew Research Center, March 17-April 12, 2015.

PEW RESEARCH CENTER

Twitter Demographics

Among internet users, the % who use Twitter

	Internet users
Total	*23%*
Men	25
Women	21
White, Non-Hispanic	20
Black, Non-Hispanic (n=85)	28
Hispanic	28
18-29	32
30-49	29
50-64	13
65+	6
High school grad or less	19
Some college	23
College+	27
Less than $30,000/yr	21
$30,000-$49,999	19
$50,000-$74,999	25
$75,000+	26
Urban	30
Suburban	21
Rural	15

Source: Pew Research Center, March 17-April 12, 2015.

PEW RESEARCH CENTER

Figure 30. Social media demographics by gender, age, race, income, and residence.

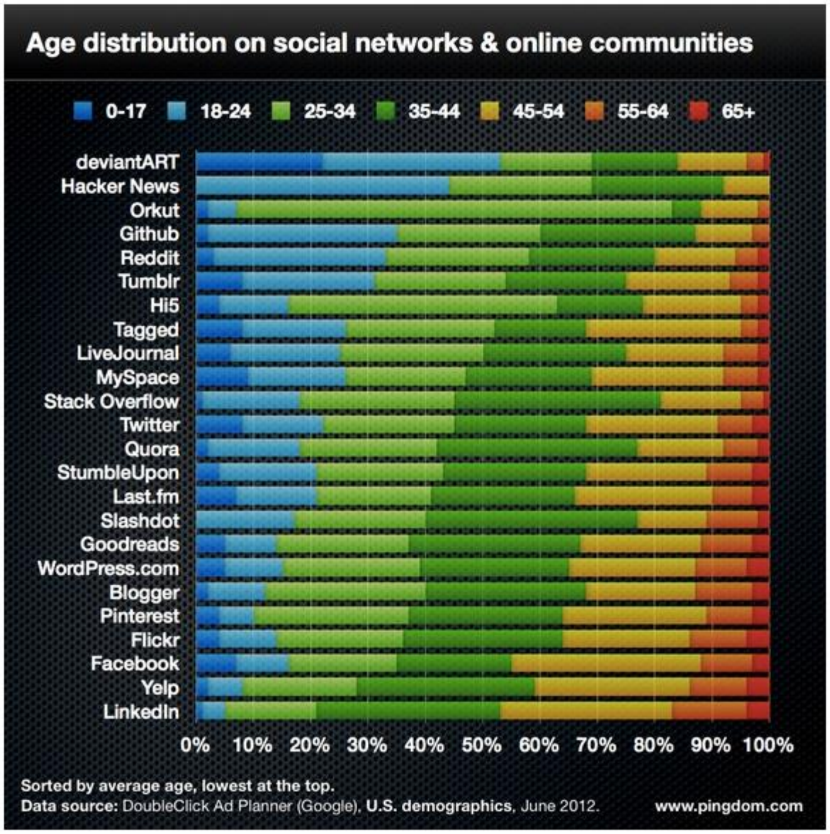


Figure 31. Social media users by age http://vator.tv/news/2012-08-22-age-gender-24-social-network-demographics

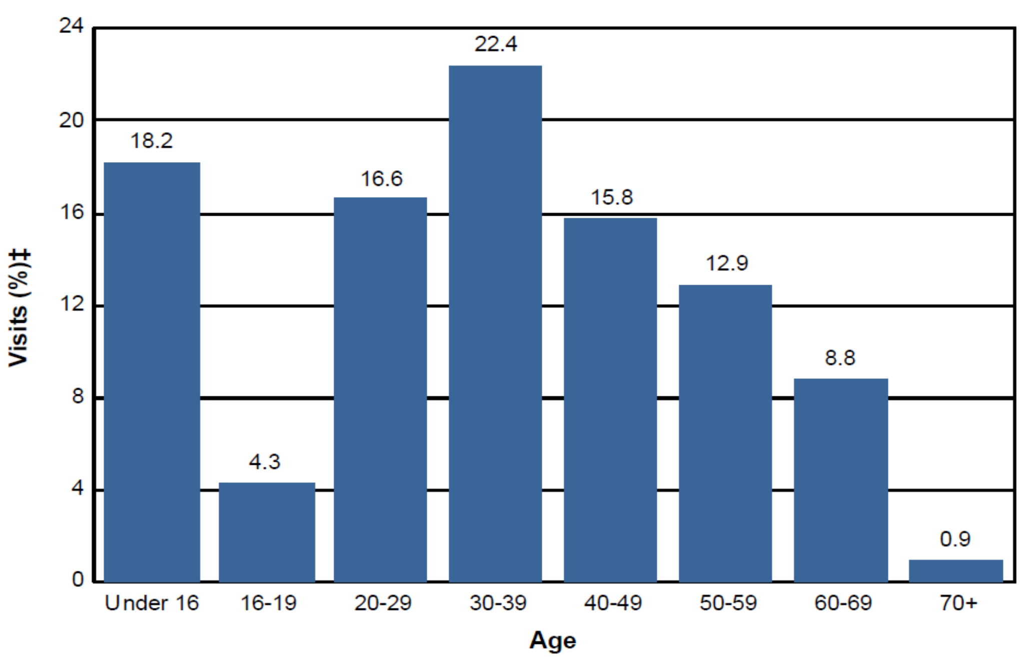


Figure 32. MBS users by age. (NVUM 2016)

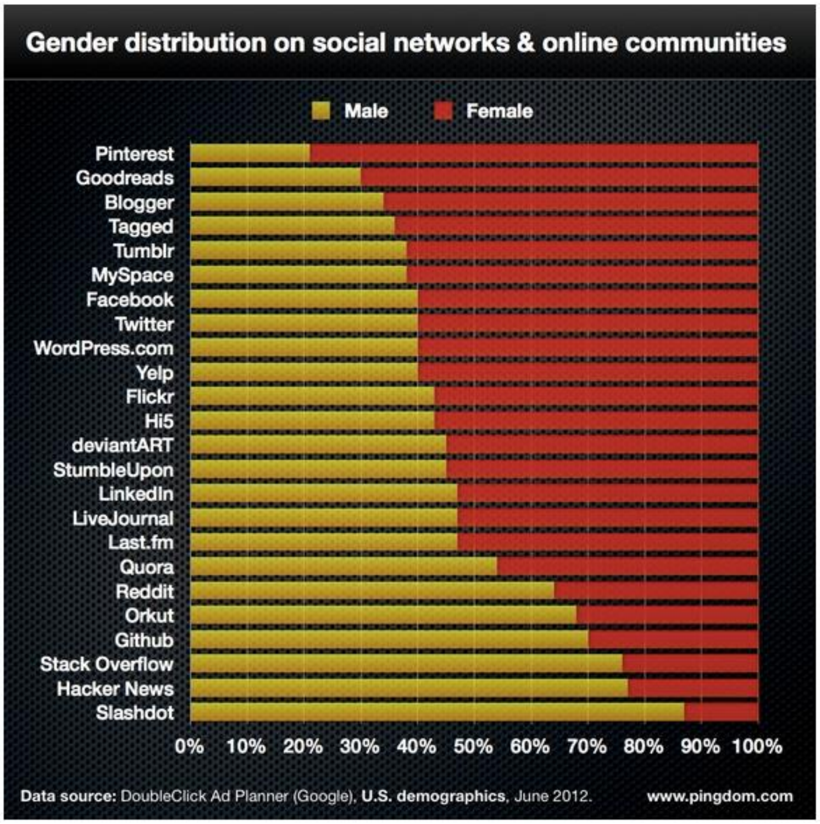


Figure 33. Social media users by gender. http://vator.tv/news/2012-08-22-age-gender-24-social-network-demographics

Gender	Survey Respondents†	National Forest Visits (%)‡
Female	1,055	40.5
Male	1,256	59.5
Total	**2,311**	**100.0**

Figure 34. MBSNF Visitors by Gender. (NVUM 2016)

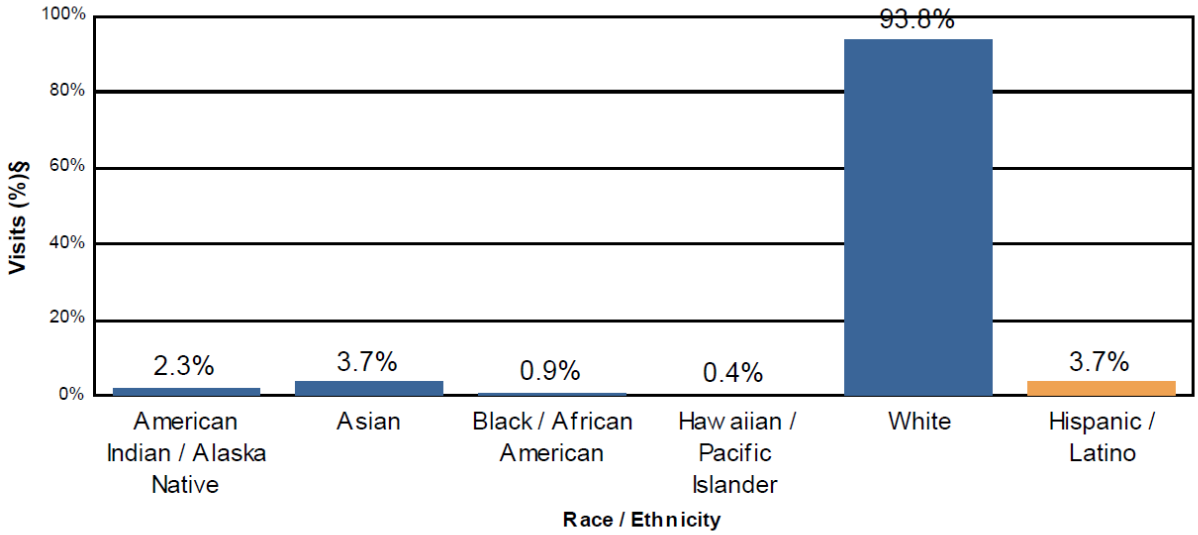


Figure 35. MBSNF users by race.

Activity	% Participation*	% Main Activity‡	Avg Hours Doing Main Activity
Hiking / Walking	57.7	33.4	4.0
Viewing Natural Features	48.8	5.7	3.4
Relaxing	37.5	6.4	24.2
Viewing Wildlife	36.7	0.9	10.9
Driving for Pleasure	24.4	3.8	3.2
Downhill Skiing	23.2	22.3	4.7
Developed Camping	11.8	5.1	35.2
Picnicking	11.5	1.2	3.3
No Activity Reported	9.8	9.8	
Nature Center Activities	7.8	0.1	1.3
Nature Study	6.3	0.1	6.0
Other Non-motorized	5.1	0.3	2.7
Gathering Forest Products	4.6	1.2	2.2
Cross-country Skiing	4.3	2.5	3.3
Fishing	4.2	0.9	8.2
Backpacking	3.1	1.7	28.1
Visiting Historic Sites	3.1	0.0	8.0
Hunting	2.8	2.5	7.3
Bicycling	2.4	0.1	3.8
Primitive Camping	1.9	0.9	29.0
OHV Use	1.8	1.3	4.4
Some Other Activity	1.8	0.4	1.9
Motorized Trail Activity	1.7	0.3	6.0
Resort Use	1.3	0.0	43.0
Non-motorized Water	1.2	0.3	4.0
Other Motorized Activity	1.1	0.0	0.0
Snowmobiling	0.3	0.0	4.0
Horseback Riding	0.0	0.0	7.0
Motorized Water Activities	0.0	0.0	0.0

Figure 36. MBSNF user activity preferences and participation.

5. Business Case & Implementation Plan

5.1 Comparison with traditional survey methods

Both traditional survey methods and social media studies will have a high cost of development. The US Forest Service uses the National Visitor Use Monitoring Process to monitor visitation (English et. al. 2001). The framework was developed with credit to senior scientists ($101,000 average annual salary), project coordinator ($71,000 average annual salary), statistician ($76,000 average annual salary) and economist ($103,000), in addition to seven other collaborators and 11 contributors. If we estimate six months of development using averages only for the primary contributors plus an estimated 33% additional for contributors (23%) and overhead (10%), we see an estimated cost of $165,000 for the development of sample methods.

Development of the InVEST recreation model required a GIS specialist ($56,000 average annual salary), an ecologist ($57,000 average annual salary) both for two months, while additionally a software engineer ($95,000 average annual salary) was required for six months. For salaries ($18,333.33 and $47,500 respectively) an estimated total of $65,833, and additional contributors such as students and staff time (23%) and overhead (10%) for an added 33% or $21,725.This suggests a development cost of approximately $87,000.

A comparison of the implementation of each survey method strongly favors the use of social media where data is available. The InVEST recreation model is publicly available and may be applied to any project. A data scientist or GIS specialist would be capable of pulling the data required in two weeks or less, for an estimated cost of under $2000. The NVUM methods in contrast, require national training, contracting workforce, training workforce, job hazard analysis and law enforcement plan, press release, permissions from lessees, procurement, pre-inspection and gathering quarterly reports. The survey would take approximately 200 site days plus preparation and training (USDA NVUM handbook) for an estimated 225 site days. Additional quality assurance and control of data collection add necessary days to the data processing. Using these proxies for time, if we estimate one forester and two technicians for one month and 200 person days (contract or staff), we find ($9843 forester, $12666 tech 2x, $32,633 survey 225 days) an estimated survey cost of $55,142 in person hours.

Each survey method provides a unique benefit. Comparison of social media against traditional survey methods is important for validation of alternative sampling methods. In that respect NVUM data could be considered a valuable compliment to social media data. NVUM also provides exact counts of use, survey of motivation and spending, as well as other proxy data such as ticket sales, permit information, and length of stay. However, NVUM data is intended to characterize a region and is not appropriate for small scale sampling such as individual trails or subsets of large districts. Social media provides insight into exactly what areas users are utilizing. Photos, tweets and recreational use records can also provide insight into what an individual values about a site through tags, captions, and subject matter. Social media also provides the benefit of fine scale time sampling, by date, season, or year. Finally, social media is capable of sampling areas not reached by traditional survey methods.

Both sampling methods present the potential for bias. Social media is heavily influenced by bias because it is a self-selecting subset of the population (Li et al 2013). Not all media is shared on one platform, therefore using multiple platforms may prove necessary to get an accurate cross section of the population. Additionally, those who don't use social media may utilize the recreation areas in different ways than those who do. Social media demographics may change over time and regular standardization should be done to assure accuracy of assessments. NVUM makes explicit efforts to control for bias in sampling, however the sampling methods do limit what uses can effectively be sampled. For example, exits to public parks can be effectively sampled, but recreational driving on highways through National Forests may not be effectively sample. NVUM samples are a small portion of the time and a subset of the population. (NVUM 2001)

5.2 Potential for Partnership

A key to implementing this methodology for a successful business endeavor is to ensure that there is adequate data in the case of both traditional visitation surveys and social media data. A partnership between social media providers and recreational use areas could provide a benefit to both parties. Land managers could advertise and incentivize social media use. The US Fish and Wildlife Service already provides opportunities for media sharing through photo competitions (https://www.fws.gov/refuges/photography/photographyContests.html). Trail and park signs could link park users to social media and provide information regarding incentives to use. Social media sites benefit from the added users and advertising and the park is provided with detailed data on park use.

Reaching an agreement with both the recreational use areas and social media concerning data collection authorization and sharing could enable land managers to move forward with a far more powerful application. Collecting social media data for recreational use areas, then comparing survey data can establish a predictive model. Once this is established, land managers could provide estimated site specific use data.

Sessions (2015) looked at Flickr PUD as a predictor of National Park Service (NPS) visitation. Her study found a correlation between seasonal usage and Flickr photo data. The correlation with Flickr data was site specific, however, and did not translate into a generalized park model. This means each area would require an independently ground validated model in order to predict usage. The NPS may be capable of collecting site specific data in addition to its existing park-wide data.

A complete business model based on this project will overall call for repeating our methodology using the best possible data resources. Forging partnerships with organizations who can provide access to this data in the form of both ground-level surveys and social media metadata will be key to successful results. We have demonstrated that such an endeavor is possible using social media scraping methods, and also indicated that ground-level survey data has the potential to validate social media data if it is collected in a greater volume and across a broad scale of time. With these resources in place, implementation of our business model is promising.

5.3 Future Work

The current work only uses metadata, location, and time. Social media is a rich source of information and using the content could provide a valuable source of information. Tags can provide insight into content and photo content can also express user values. Machine learning algorithms can be used for processing natural language and photo content (Misra et al 2016). These algorithms have advanced significantly in recent years and may provide a valuable addition to metadata alone.

One aspect not analyzed here which could have a significant impact is how distance from a trailhead and cell coverage influence the social media post density. Strava segment users have a number of methods for uploading data, including GPS. However, mobile device users may put the cameras away when they lose cell coverage or turn off their phones. Likewise, they would be more likely to run out of battery further from coverage and further from the trailhead.

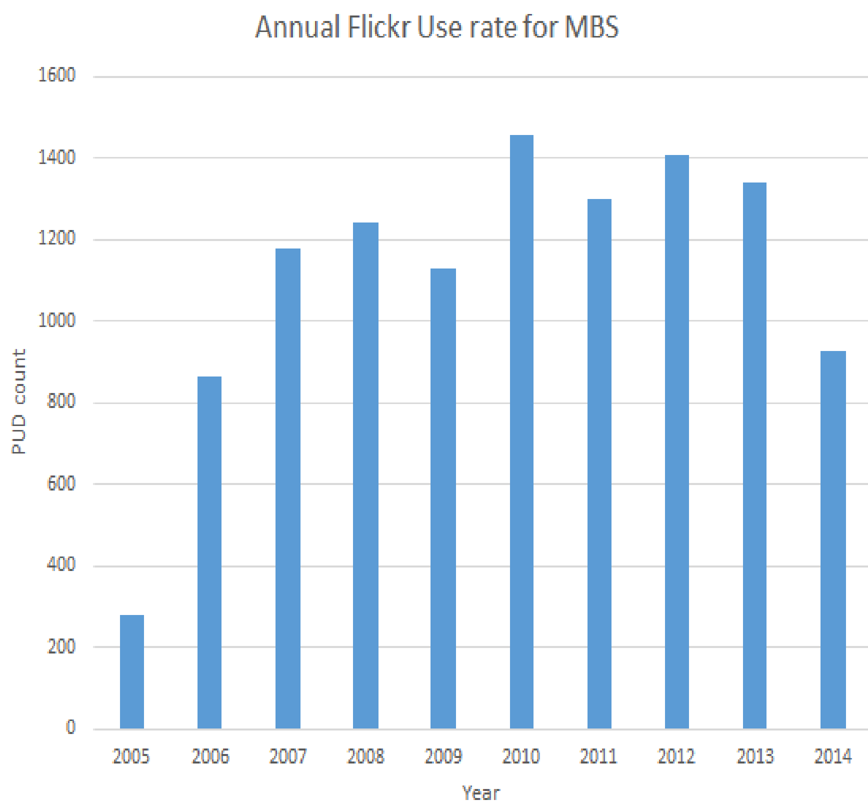


Figure 37. Flickr use rate by year for MBSNF.

The social media landscape is in constant flux. Use rates change based on user preferences, new apps, and changes in technology. Flickr use rates are shown by year in Figure 37. Additionally, APIs and available data change. Potentially valuable sources such as Facebook and Instagram may open their APIs allowing for increased analysis. Instagram offers a wide user base that is growing, and collecting geographic metadata that makes it possible to map visitation on a larger scale than Flickr. However, Instagram currently has a closed API and restricts data collection to a sandbox mode, unless applications such as a social media scraper are approved for official use by a review process (Swanner 2015). Use of this data rests on acquiring special permission, or changes in the API. The general methods described here are a useful framework, but validating data with survey information will be essential for accurate methodology. Finally, development of the analysis methods must keep pace with the changes in the social media landscape.

6. References

Allton, Mike. "Social Media Active Users by Network [INFOGRAPH]." The Social Media Hat. May 30, 2014. Accessed August 18, 2016. https://www.thesocialmediahat.com/active-users.

American Statistical Association. 1997. “More about Mail surveys.” ASA Series What is a Survey? Section on Survey Research Methods. http://158.132.155.107/posh97/private/research/survey/Mail.pdf/.

"Cellular Geographic Service Areas." Federal Communications Commission. September 09, 2013. Accessed August 18, 2016. https://www.fcc.gov/general/cellular-geographic-service-areas.

“Company Salaries and Reviews.” Glassdoor. Accessed August 18, 2016. http://www.glassdoor.com.

"Data Library." Mt Baker-Snoqualmie National Forest. January 10, 2012. Accessed August 18, 2016. http://www.fs.fed.us/r6/data-library/gis/mtbaker-snoqualmie/.

Dotan, Amir, and Panayiotis Zaphiris. "A cross-cultural analysis of Flickr users from Peru, Israel, Iran, Taiwan and the UK." *International Journal of Web Based Communities* 6, no. 3 (2010): 284-302.

Duggan, Maeve. "The Demographics of Social Media Users." Internet Science Tech RSS. August 19, 2015. Accessed August 18, 2016. http://www.pewinternet.org/2015/08/19/the-demographics-of-social-media-users/.

English, Donald. Kocis, Susan. Zarnoch, Stanley. Arnold, Ross. 2001 “Forest Service National Visitor Use Monitoring Process” Research Method Documentation

Laney, Leroy O. 2009. *Assessing Tourism's Contribution To The Hawaii Economy*. Economic Forecast. First Hawaiian Bank. http://www.4mauirealestate.com/sites/default/files/FHB_Tourism_Study_09325.pdf.

Li, Linna, Michael F. Goodchild, and Bo Xu. "Spatial, temporal, and socioeconomic patterns in the use of Twitter and Flickr." *Cartography and Geographic Information Science* 40, no. 2 (2013): 61-77.

Misra, Ishan. Zitnick, C. Lawrence. Mitchell, Margaret. Girshick, Ross. 2016 “Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels” Cornell University Library. arXiv:1512.06974 [cs.CV]. http://arxiv.org/abs/1512.06974/.

Outdoor Industry Association. "The Outdoor Recreation Economy." *Boulder, CO* (2012).

Paracchini, Maria Luisa, Grazia Zulian, Leena Kopperoinen, Joachim Maes, Jan Philipp Schägner, Mette Termansen, Marianne Zandersen, Marta Perez-Soba, Paul A. Scholefield, and Giovanni Bidoglio. "Mapping Cultural Ecosystem Services: A Framework to Assess the Potential for Outdoor Recreation across the EU." *Ecological Indicators* 45 (2014): 371-85. doi:10.1016/j.ecolind.2014.04.018.

Pergams, Oliver RW, and Patricia A. Zaradic. "Evidence for a fundamental and pervasive shift away from nature-based recreation." Proceedings of the National Academy of Sciences 105, no. 7 (2008): 2295-2300.

"Photography Contests." National Wildlife Refuge System. Accessed August 18, 2016. https://www.fws.gov/refuges/photography/photographyContests.html.

Sharp, R., Tallis, H.T., Ricketts, T., Guerry, A.D., Wood, S.A., Chaplin-Kramer, R., Nelson, E., Ennaanay, D., Wolny, S., Olwero, N., Vigerstol, K., Pennington, D., Mendoza, G., Aukema, J., Foster, J., Forrest, J., Cameron, D., Arkema, K., Lonsdorf, E., Kennedy, C., Verutes, G., Kim, C.K., Guannel, G., Papenfus, M., Toft, J., Marsik, M., Bernhardt, J., Griffin, R., Glowinski, K., Chaumont, N., Perelman, A., Lacayo, M. Mandle, L., Hamel, P., Vogl, A.L., Rogers, L., and Bierbower, W. 2016. InVEST 3.3.0 User's Guide. The Natural Capital Project, Stanford University, University of Minnesota, The Nature Conservancy, and World Wildlife Fund.

Slavonia, Mark. "Sampling Strava." March 12, 2015. Accessed August 18, 2016. http://markslavonia.com/sampling-strava/.

Smith, Craig. "14 Interesting Flickr Stats." DMR. July 13, 2015. Accessed August 18, 2016. http://expandedramblings.com/index.php/flickr-stats/.

Spasojevic, Mirjana, Tim Kindberg, Rowanne Fleck, and Abigail Sellen. "The Ubiquitous Camera: An In-Depth Study of Cameraphone Use." (2005). http://research.microsoft.com/apps/pubs/default.aspx?id=69232/.

Strava. “Run and Cycling Tracking on the Social Network for Athletes” Strava.com. http://www.strava.com/.

Swanner, Nate. "Instagram limits developer API access with new app review process." The Next Web. http://thenextweb.com/dd/2015/11/17/instagram-limits-developer-api-access-with-new-app-review-process (accessed August 4, 2016).

Turner, Rochelle. Travel & Tourism Economic Impact. Report. 2015. Accessed August 18, 2016. https://www.wttc.org/-/media/files/reports/economic impact research/regional 2015/world2015.pdf/.

US Department of Agriculture. “National Visitor Use Monitoring Results.” 2013. USDA Forest Service National Summary Report.

Appendices

Appendix A: Buffer Scripts

```
# -*- coding: utf-8 -*-
"""
Creating a buffer for polyline trail data.
Created on Thu Jun 30 17:46:56 2016
Draws from:
https://pcjericks.github.io/py-gdalogr-cookbook/vector_layers.html#create-buffer
https://github.com/OSGeo/gdal/blob/trunk/gdal/swig/python/samples/gdalcopyproj.py

@author: MStelmach
"""

import ogr, os, shutil



daShapefile = r" ‘’’input shapefile path’’’ "
output = r" ‘’’output shapefile location’’’ "

# Input Feature name, Output or Buffered Feature name, Buffer Distance
def createBuffer(inputfn, outputBufferfn, bufferDist):
    #Set data source - input data
    inputds = ogr.Open(inputfn)
    inputlyr = inputds.GetLayer()
    inputlyrdef = inputlyr.GetLayerDefn()

    #Determine if the file name already exists.  If it exists it will be deleted
    shpdriver = ogr.GetDriverByName('ESRI Shapefile')
    if os.path.exists(outputBufferfn):
        shpdriver.DeleteDataSource(outputBufferfn)

    #Create a new polygon shapefile
    outputBufferds = shpdriver.CreateDataSource(outputBufferfn)
    bufferlyr = outputBufferds.CreateLayer(outputBufferfn, geom_type=ogr.wkbPolygon)

     #read fields in input file
    fieldlist = []
    for i in range(inputlyrdef.GetFieldCount()):
        fieldname =  inputlyrdef.GetFieldDefn(i).GetName()
        fieldTypeCode = inputlyrdef.GetFieldDefn(i).GetType()
        #fieldType = inputlyrdef.GetFieldDefn(i).GetFieldTypeName(fieldTypeCode)
        fieldWidth = inputlyrdef.GetFieldDefn(i).GetWidth()
        Getprecision = inputlyrdef.GetFieldDefn(i).GetPrecision()

        fieldlist.append(fieldname)
```

```
        newfield = ogr.FieldDefn(fieldname, fieldTypeCode)
        newfield.SetWidth(fieldWidth)
        newfield.SetPrecision(Getprecision)
        bufferlyr.CreateField(newfield)

    featureDefn = bufferlyr.GetLayerDefn()

        # For each feature in the input file create a buffer polygon in the new shapefile
    for feature in inputlyr:

        ingeom = feature.GetGeometryRef()
        # if the feature has a geometry, buffer it
        if ingeom:
            geomBuffer = ingeom.Buffer(bufferDist)
            outFeature = ogr.Feature(featureDefn)
            outFeature.SetGeometry(geomBuffer)


            #copy fields
            for field in fieldlist:
                inputvalue= feature.GetField(field)
                outFeature.SetField(field, inputvalue)

            bufferlyr.CreateFeature(outFeature)

def main(inputfn, outputBufferfn, bufferDist):
    createBuffer(inputfn, outputBufferfn, bufferDist)

if __name__ == "__main__":
    inputfn = daShapefile
    outputBufferfn = output
    bufferDist = 500.0

    #run buffer
    main(inputfn, outputBufferfn, bufferDist)
    # copy projection
    proj = os.path.splitext(daShapefile)[0]+'.prj'
    outproj = os.path.splitext(output)[0]+'.prj'
    shutil.copy(proj, outproj)
```

Appendix B: Strava API Scraper

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 11 19:42:05 2016

@author: MStelmach
"""

import stravalib
import requests
import datetime
import polyline
import time
import types
from osgeo import ogr
from math import ceil
import sys, os
import json

#Get current time
starttime = datetime.datetime.now()
scriptstart = starttime

#Define the bounding box for strava search
#Example: Bounds = r"C:\MyDocuments\adminownprojected.shp"
#Example 2: Bounds = [-122.19401695785098, 46.874837987936324, -120.90679054890244,
48.99982719694641]
def BoundingBox(Bounds):
    global bbox
    #input either SW and NE corner as a length 4 list or a shapefile in WGS84
    if type(Bounds) == types.StringType:
        try:
            driver = ogr.GetDriverByName('ESRI Shapefile')
            ds = driver.Open(Bounds)
            if ds:
                layer = ds.GetLayerByIndex(0)
                extent  = layer.GetExtent()
                #print extent
                bbox = {'LonMin':extent[0], 'LatMin':extent[2], 'LonMax':extent[1], 'LatMax':extent[3]} #SW and
NE corners
                ds.Destroy()
            else:
                sys.exit("Input not appropriate Shapefile: operation canceled")
        except:
            sys.exit("Input not Shapefile: operation canceled")
    elif type(Bounds) == types.ListType and len(Bounds)==4:
```

```
    bbox = {'LonMin':Bounds[0], 'LatMin':Bounds[1], 'LonMax':Bounds[2], 'LatMax':Bounds[3]}
  for item in bbox:
    if bbox[item]>180 or bbox[item]<-180:
      print 'Warning Bounds must be in WGS84 decimal degrees'

  #bounding box
  global LatMin, LatMax, LonMin, LonMax
  LatMin = bbox['LatMin']
  LatMax = bbox['LatMax']
  LonMin = bbox['LonMin']
  LonMax = bbox['LonMax']

  latcount = (abs(LatMax-LatMin))/0.01
  loncount = (abs(LonMax-LonMin))/0.01
  totalCells = ceil(latcount)*ceil(loncount)
  processtime = totalCells/30000


  if totalCells > 20000:
    print '{0} cells in area, processing time {1} or more days (depending on trail density). '.format
(totalCells, processtime)
    print '30,000 API Calls per day maximum'

  return [LonMin, LatMin, LonMax, LatMax]

AOI =
BoundingBox(r"H:\MyDocuments\UW\CapstoneProject\GIS_Background\adminown\adminownprojected.s
hp")
print bbox

workingdir = “”” Define output location”””
if not os.path.exists(workingdir):
  os.mkdir(workingdir)

tempfile = os.path.join(workingdir + ' “”” Output filename, include .txt extension “”” ')

#Stravalib client identification
AccessToken = " ‘’’SPECIFY ACCESS TOKEN FROM STRAVA.COM’’’ "
client = stravalib.client.Client()
client.access_token = AccessToken

#Bounding box tracking file
def boundfiletracker(starttime, workingdir, bounds, bbox, dist):

  tracker = {}
  boundfiletracker = open(os.path.join(workingdir + 'LastBoundingBox.txt'), 'w')
  tracker['StartTime'] = starttime.isoformat()
  tracker['CurrentTime'] = datetime.datetime.now().isoformat()
```

```
    tracker['currentbox'] = {'boxLatMin':bounds[0], 'boxLonMin':bounds[1], 'boxLatMax':bounds[2],
'boxLonMax':bounds[3]}
    tracker['extent'] = bbox
    tracker['dist'] = dist
    boundfiletracker.write(json.dumps(tracker))
    boundfiletracker.close()


# create an empty set for segment ID numbers, and counters for row, column, and count of empty lists
segmentlist = set()
emptylistcount =0
row =1
count = 0
calls = 0
dist = 0.05
LatMinOriginal = LatMin
LonMinOriginal = LonMin
latmax = abs(LatMax-LatMin)
lonmax = abs(LonMax-LonMin)
if latmax>lonmax:
    maxdist = latmax
else:
    maxdist = lonmax


MBSStrava = {}
#start searching from the southwest and work north by 0.01 decimal degree
# when the latitude exceeds the max, restart at lat min and shift east 0.01 deg
# and expand by double each time through the area of interest
while dist<maxdist:

    while (LatMin+(dist/2))<LatMax or (LonMin+(dist/2))<LonMax:#if (LatMin + dist)< LatMax all results will
be inside bounds

        if (LatMin+dist/2)<LatMax:
            #search an area just large enough to contain the largest running segments
            bounds = [LatMin, LonMin, LatMin+dist, LonMin+dist]
            boundfiletracker(starttime, workingdir, bounds, bbox, dist)

            time.sleep(4) # trying to avoid the 30000/day limit
            try:
                calls +=1
                for item in client.explore_segments(bounds):

                    #if the item exists and has not already been scraped, scrape data
                    if item and item.id not in segmentlist:
                        SegmentID = item.id #unique ID number
                        print 'New segment ID: ', SegmentID
```

```
            segmentlist.add(SegmentID)
            SegmentName = item.name #segment name
            calls +=1

            try:
                SegDetails = client.get_segment(SegmentID)
                usetype = SegDetails.activity_type

                creationdate = SegDetails.created_at #Date the segment was uploaded to strava
                creationdate = creationdate.replace(tzinfo=None)

                effortcount = SegDetails.effort_count #efforts between creation date and collection date
                #print 'effort count' , effortcount
                athletecount = SegDetails.athlete_count #number of athletes who have tried the
segment
                timedelt = starttime-creationdate
                TimeDelt = timedelt.total_seconds()
                yrs = TimeDelt/(60*60*24*365) #convert seconds to years

                if yrs:
                    userate = float(effortcount)/yrs #Efforts per year as 'userate'

                    #print userate
                else:
                    userate = 0.0

                line  = SegDetails.map.polyline

                shape = polyline.decode(line) #google polyline encoding as a string decode with
polyline lib

                if shape:
                    wkt = ogr.Geometry(ogr.wkbLineString)
                    for item in shape:
                        wkt.AddPoint(item[1],item[0])
                    #Create a list with all attributes.

                    MBSStrava[str(SegmentID)]={'name': SegmentName,
                                    'segmentID': int(SegmentID),
                                    'usetype': usetype,
                                    'creationdate': int(creationdate.strftime('%Y%m%d')),
                                    'effortcount':int(effortcount),
                                    'athletecount':int(athletecount),
                                    'years':float(yrs),
                                    'userate':float(userate),
                                    'wkt':str(wkt),
                                    'bounds':bounds,
                                    'dist': dist}
```

```
                writefile = open(tempfile, 'w')
                s = json.dumps(MBSStrava)
                writefile.write(s)
                writefile.close()
                print len(MBSStrava), 'segments read'


            except requests.exceptions.HTTPError:
                print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
                print bounds
                time.sleep(60)
                continue
            except stravalib.exc.RateLimitExceeded:
                print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
                print bounds
                time.sleep(60)
                continue
            except KeyboardInterrupt:
                break
            except:
                print 'Other exception occurred:', sys.exc_info()
                break

        else:
            if item:
                try:
                    print item.id , 'already contained in segmentlist'
                except ValueError:
                    print 'Value Error Exception on a duplicate segment'
                except:
                    print 'Other exception on a duplicate segment', sys.exc_info()[0]
            else:
                print 'empty list:', emptylistcount
    LatMin += (dist/5)
    count +=1
    print 'row:',row, ' , count:', count, 'calls:', calls, 'dist:', dist, 'bounds:', bounds
except requests.exceptions.HTTPError:
    print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
    time.sleep(60)
    LastBox = bounds
    print 'last sample area: ', str(bounds)
    continue

except stravalib.exc.RateLimitExceeded:
    print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
    LastBox = bounds
    print bounds
    time.sleep(60)
```

```
            continue
        except KeyboardInterrupt:
            #data_source.Destroy()
            break
        except:
            print 'Other exception occurred:', sys.exc_info()
            #data_source.Destroy()
            break

    else: #when all latitudes have been searched restart at the south and shift east
        print 'sleep'
        time.sleep(4) # trying to avoid the 30000/day limit, but 3 was not long enough.
        LatMin = LatMinOriginal
        LonMin += (dist/5)
        row +=1
        print 'New Row: ' , row
        print 'lat, long : ' , LatMin, LonMin
        print 'count : ' , count
        print bounds
        print datetime.datetime.now()
        boundfiletracker(starttime, workingdir, bounds, bbox, dist)
  dist = dist*2
  LatMin = LatMinOriginal
  LonMin = LonMinOriginal
#writefile.close()
#data_source.Destroy()
```

Appendix C: Restart Scraper

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jul 19 20:53:01 2016

@author: MStelmach
"""

import polyline
import simplejson
import datetime
from osgeo import ogr
import stravalib
import time
import requests
import sys

starttime = datetime.datetime.now()

workingdir = ' “””Set your working directory””” '
#segments = workingdir + '\.txt'#ADD YOUR PREVIOUS OUTPUT
previous_JSON = workingdir + '\lastboundingbox.txt'#ADD YOUR PREVIOUS TRACKING OUTPUT

tempfile = workingdir + '\stravasegments20160721.txt'  #ADD YOUR PREVIOUS OUTPUT

jsonreader = open(previous_JSON, 'r')

reader = jsonreader.read()

print reader, type(reader)

stats = simplejson.loads(reader)
#lastboundingbox = boundreader.readline().replace('\n','')
lastrunstart = stats['StartTime'] #boundreader.readline().replace('\n','')
lastrunend = stats['CurrentTime']#boundreader.readline().replace('\n','')
currentbox = stats['currentbox']#boundreader.readline().replace('\n','')
extent = stats['extent']#boundreader.readline().replace('\n','')
dist = stats['dist']
#print lastboundingbox , type (lastboundingbox)
#print 'last run start:', lastrunstart, type (lastrunstart)
#print 'last run end:',lastrunend, type (lastrunend)
#print 'current box: ', currentbox, type (currentbox)
#print 'extent:', extent, type (extent)
jsonreader.close()
```

```
lastend = datetime.datetime.strptime(lastrunend,'%Y-%m-%dT%H:%M:%S.%f')
laststart = datetime.datetime.strptime(lastrunstart,'%Y-%m-%dT%H:%M:%S.%f')
lastruntime = lastend-laststart
print 'last run time (hours) :', lastruntime.total_seconds()/(60*60)
deltadate = starttime.date() - lastend.date()
deltadate = deltadate.total_seconds()/(60*60*24)
print deltadate, 'days since last run'

LatMin = currentbox['boxLatMin']
LatMax = currentbox['boxLatMax']
LonMin = currentbox['boxLonMin']
LonMax = currentbox['boxLonMax']
LatMinOriginal = extent['LatMin']
LatMaxOriginal = extent['LatMax']
LonMinOriginal = extent['LonMin']
LonMaxOriginal = extent['LonMax']

#print LatMin,LatMax,LonMin,LonMax, dist

#previoussegments = set()
#StravaSegments = open(segments, 'r')

#MBSStrava = simplejson.loads(StravaSegments.read())
#for line in MBSStrava:
#    previoussegments.add(MBSStrava[line]['segmentID'])

#StravaSegments.close()
#print previoussegments


#Stravalib client identification
AccessToken = "" #ADD YOUR ACCESS TOKEN
client = stravalib.client.Client()
client.access_token = AccessToken

#Bounding box tracking file
def boundfiletracker(starttime, workingdir, bounds, bbox, dist):

    tracker = {}
    boundfiletracker = open(workingdir + '\LastBoundingBox.txt', 'w')
    tracker['StartTime'] = starttime.isoformat()
    tracker['CurrentTime'] = datetime.datetime.now().isoformat()
    tracker['currentbox'] = {'boxLatMin':bounds[0], 'boxLonMin':bounds[1], 'boxLatMax':bounds[2],
'boxLonMax':bounds[3]}
    tracker['extent'] = bbox
    tracker['dist'] = dist
    boundfiletracker.write(simplejson.dumps(tracker))
    boundfiletracker.close()
```

```
# create an empty set for segment ID numbers, and counters for row, column, and count of empty lists
segmentlist = set()
emptylistcount =0
row =1
count = 0
calls = 0

latmax = abs(LatMaxOriginal-LatMinOriginal)
lonmax = abs(LonMaxOriginal-LonMinOriginal)
if latmax>lonmax:
    maxdist = latmax
else:
    maxdist = lonmax

#start searching from the southwest and work north by (dist) decimal degree
# when the latitude exceeds the max, restart at lat min and shift east 0.01 deg
while dist<maxdist:

    while LatMin<LatMaxOriginal or LonMin<LonMaxOriginal:
        if LatMin<LatMaxOriginal:
            #search an area just large enough to contain the largest running segments
            bounds = [LatMin, LonMin, LatMin+dist, LonMin+dist]
            boundfiletracker(starttime, workingdir, bounds, extent, dist)
            if calls >= 30000:
                timedif = datetime.datetime(starttime.year, starttime.month, starttime.day+1)-starttime
                if timedif.total_seconds() < 0:
                    calls = 0
                    starttime = datetime.datetime.now()
                elif timedif.total_seconds()>0:
                    time.sleep(timedif.total_seconds())
                    calls = 0
                    starttime = datetime.datetime.now()

            time.sleep(4) # trying to avoid the 30000/day limit, but 3 was not long enough.
            try:
                calls +=1
                for item in client.explore_segments(bounds):

                    #if the item exists and has not already been scraped, scrape data
                    if item and str(item.id) not in MBSStrava:
                        SegmentID = item.id #unique ID number
                        print 'New segment ID: ', SegmentID
                        SegmentName = item.name #segment name
                        calls +=1
                        #print SegmentName
                        try:
                            SegDetails = client.get_segment(SegmentID)
```

```
                usetype = SegDetails.activity_type
                #print calls
                creationdate = SegDetails.created_at #Date the segment was uploaded to strava
                creationdate = creationdate.replace(tzinfo=None)
                effortcount = SegDetails.effort_count #efforts between creation date and collection date
                athletecount = SegDetails.athlete_count #number of athletes who have tried the
segment
                timedelt = starttime-creationdate
                TimeDelt = timedelt.total_seconds()
                yrs = TimeDelt/(60*60*24*365) #convert seconds to years

                if yrs:
                    userate = float(effortcount)/yrs #Efforts per year as 'userate'

                else:
                    userate = 0.0
                line  = SegDetails.map.polyline
                shape = polyline.decode(line) #google polyline encoding as a string decode with
polyline lib

                if shape:
                    wkt = ogr.Geometry(ogr.wkbLineString)
                    for item in shape:
                        wkt.AddPoint(item[1],item[0])

                    #Create a list with all attributes.
                    #info = [, SegmentID, creationdate.strftime('%Y%m%d'), int(effortcount),
int(athletecount), yrs, userate, wkt]

                    MBSStrava[str(SegmentID)]={'name': SegmentName,
                                    'segmentID': int(SegmentID),
                                    'usetype': usetype,
                                    'creationdate': int(creationdate.strftime('%Y%m%d')),
                                    'effortcount':int(effortcount),
                                    'athletecount':int(athletecount),
                                    'years':float(yrs),
                                    'userate':float(userate),
                                    'wkt':str(wkt),
                                    'bounds':bounds,
                                    'dist': dist}
                    writefile = open(tempfile, 'w')
                    s = simplejson.dumps(MBSStrava)
                    writefile.write(s)
                    writefile.close()
                    print len(MBSStrava), 'segments read'


            except requests.exceptions.HTTPError:
```

```
                print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
                print bounds
                time.sleep(60)
                continue
            except stravalib.exc.RateLimitExceeded:
                print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
                print bounds
                time.sleep(60)
                continue
            except KeyboardInterrupt:
                break
            except:
                print 'Other exception occurred:', sys.exc_info()
                break

        else:
            if item:
                try:
                    print item.id , 'already contained in segmentlist'
                except ValueError:
                    print 'Value Error Exception on a duplicate segment'
                except:
                    print 'Other exception on a duplicate segment', sys.exc_info()[0]
            else:
                print 'empty list:', emptylistcount
    LatMin += (dist/5)
    count +=1
    print 'row:',row, ' , count:', count, 'calls:', calls, 'dist:', dist, 'bounds:', bounds
except requests.exceptions.HTTPError:
    print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
    time.sleep(60)
    LastBox = bounds
    print 'last sample area: ', str(bounds)
    continue

except stravalib.exc.RateLimitExceeded:
    print 'Rate Exceeded, waiting 1 minute', datetime.datetime.now()
    LastBox = bounds
    print bounds
    time.sleep(60)
    continue
except KeyboardInterrupt:
    #data_source.Destroy()
    break
except:
    print 'Other exception occurred:', sys.exc_info()
    #data_source.Destroy()
    break
```

```
    else: #when all latitudes have been searched restart at the south and shift east
        time.sleep(4) # trying to avoid the 30000/day limit, but 3 was not long enough.
        LatMin = LatMinOriginal
        LonMin += (dist/5)
        row +=1
        print 'New Row: ' , row
        print 'lat, long : ' , LatMin, LonMin
        print 'count : ' , count
        print bounds
        print datetime.datetime.now()
        boundfiletracker(starttime, workingdir, bounds, extent, dist)
dist = dist*2
LatMin = LatMinOriginal
LonMin = LonMinOriginal
```

Appendix D: JSON to Shapefile

```
# -*- coding: utf-8 -*-
"""
Created on Sat Jul 16 06:33:00 2016

@author: MStelmach
"""

from osgeo import ogr,osr
import simplejson

input_file = 'H:\MyDocuments\UW\CapstoneProject\scripts\workingstravasegments20160724.txt'
output_shapefile =
'H:\MyDocuments\UW\CapstoneProject\NatCapData\Strava\MBSStravaSegmentsFROMJSON20160726.
shp'

reader = open(input_file, 'r')

segmentdict = simplejson.loads(reader.read())

fmt = '%Y-%m-%dT%H:%M:%S.%f' #iso date format input

driver = ogr.GetDriverByName("ESRI Shapefile")
srs = osr.SpatialReference()
srs.SetWellKnownGeogCS("WGS84") #WGS 84 decimal degrees
http://spatialreference.org/ref/epsg/wgs-84/

data_source = driver.CreateDataSource(output_shapefile)
layer = data_source.CreateLayer("StravaSegs", srs, ogr.wkbMultiLineString)

field_name = ogr.FieldDefn("Name", ogr.OFTString)
field_name.SetWidth(50)
layer.CreateField(field_name)
field_StravaID = ogr.FieldDefn( "StravaID", ogr.OFTInteger)
field_StravaID.SetWidth(12)
layer.CreateField(field_StravaID)
field_useType = ogr.FieldDefn( "UseType", ogr.OFTString)
field_useType.SetWidth(12)
layer.CreateField(field_useType)
field_CreateDate = ogr.FieldDefn("C_Date", ogr.OFTInteger)
layer.CreateField(field_CreateDate)
field_TimeDelt = ogr.FieldDefn("TimeDelt", ogr.OFTReal)
layer.CreateField(field_TimeDelt)
layer.CreateField(ogr.FieldDefn("Efforts", ogr.OFTInteger))
layer.CreateField(ogr.FieldDefn("Use_Rate", ogr.OFTReal))
layer.CreateField(ogr.FieldDefn("Ath_Cnt", ogr.OFTInteger))
```

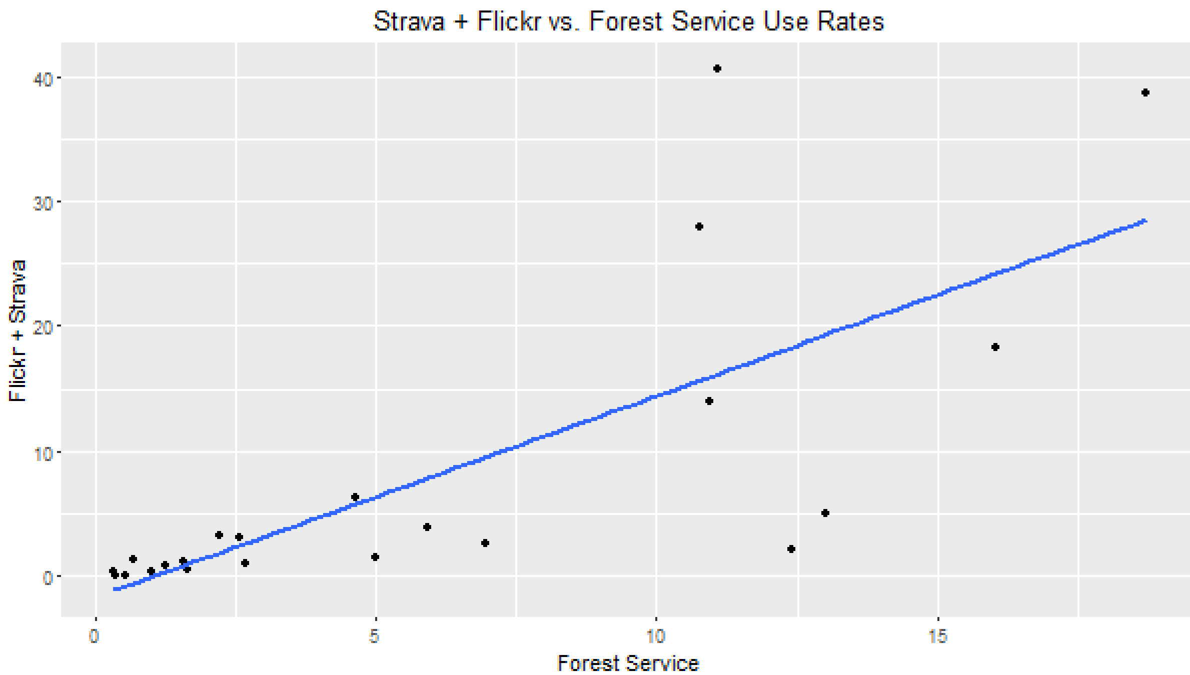
```
field_bounds = ogr.FieldDefn("Bounds", ogr.OFTString)
field_bounds.SetWidth(100)
layer.CreateField(field_bounds)
layer.CreateField(ogr.FieldDefn("Dist", ogr.OFTReal))

for line in segmentdict:
    this = segmentdict[line]
    wkt = this['wkt'] #return the coordinates within the list
    if wkt:
        athletes = this['athletecount']
        usetype = this['usetype']
        efforts = this['effortcount']
        creationdate = this['creationdate']
        yrs = this['years'] #convert seconds to years
        userate = this['userate']
        SegID = this['segmentID']
        segmentName = this['name']
        bounds = str(this['bounds'])
        dist = this['dist']

        #Set feature attributes
        feature = ogr.Feature(layer.GetLayerDefn())
        feature.SetField("Name", segmentName)
        feature.SetField("StravaID", SegID)
        feature.SetField("UseType", usetype)
        feature.SetField("C_Date", creationdate)
        feature.SetField("TimeDelt", yrs)
        feature.SetField("Efforts", efforts)
        feature.SetField("Use_Rate", userate)
        feature.SetField("Ath_Cnt", athletes)
        feature.SetField("Bounds", bounds)
        feature.SetField("Dist", dist)

        polyline = ogr.CreateGeometryFromWkt(wkt)
        feature.SetGeometry(polyline) #Set geometry to of line feature
        layer.CreateFeature(feature) #Create in shapefile
        feature.Destroy() #remove to free resources
        layer.SyncToDisk()
        data_source.SyncToDisk()

reader.close()
data_source.Destroy()
```

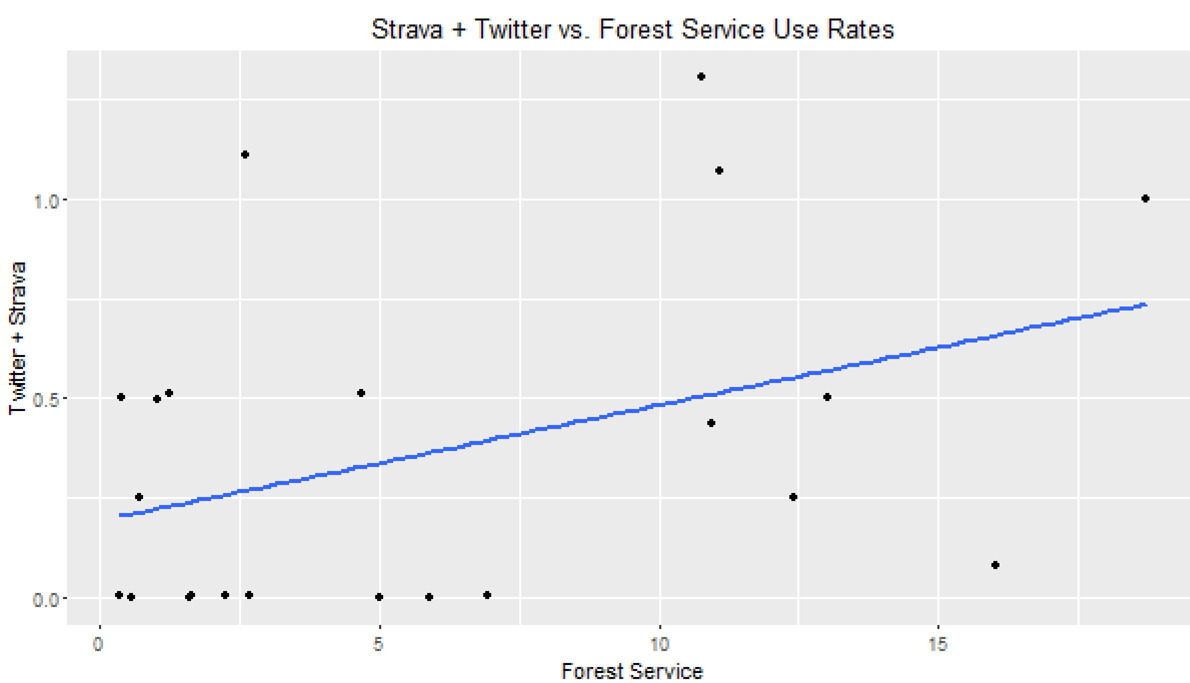
Appendix E: Intersect Strava Segments with Trail buffers

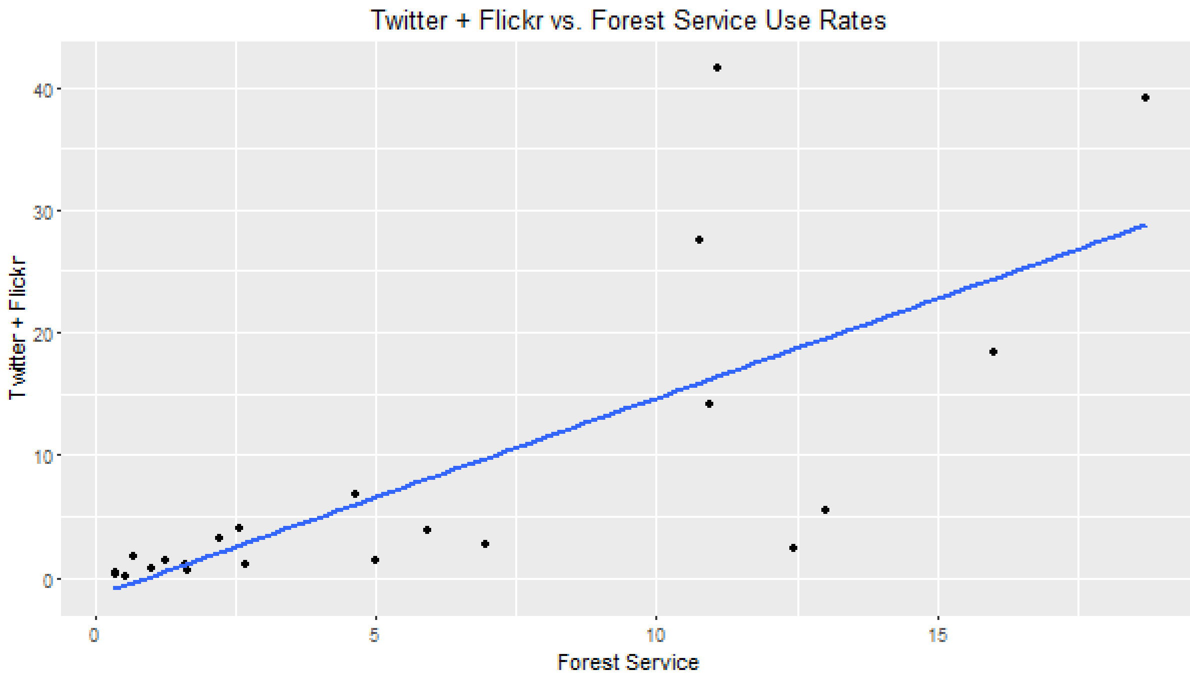
```
# -*- coding: utf-8 -*-
"""
Created on Sun Jul 24 10:44:27 2016
@author: MStelmach
"""

import os
from osgeo import ogr, osr
#from collections import Counter

input_shapefile = r “””Strava segments output from JSON to Shapefile””” '
intersectWith = r' ’’’Trail buffer shapefile””” '
output = r' ”””Output Shapefile””” '

driver = ogr.GetDriverByName("ESRI Shapefile")
srs = osr.SpatialReference()

print 'input shapefile', os.path.exists(input_shapefile)
print 'intersect shapefile', os.path.exists(intersectWith)

#Open the trails buffer shapefile in read only mode
dsTrailBuffer = driver.Open(intersectWith, 0)
lyrTrailBuffer = dsTrailBuffer.GetLayer()
lyrdefTrailBuffer = lyrTrailBuffer.GetLayerDefn()
trailfeat = lyrTrailBuffer.GetNextFeature()
trailgeom = trailfeat.geometry().Clone()
#for i in range(lyrdefTrailBuffer.GetFieldCount()):
TBproj = lyrTrailBuffer.GetSpatialRef()
schema  = lyrTrailBuffer.schema


#Open the strava data in read only mode
dsStrava = driver.Open(input_shapefile, 0)
lyrStrava = dsStrava.GetLayer()
stravafeat = lyrStrava.GetNextFeature()
stravageom = stravafeat.geometry().Clone()
lyrdefStrava = lyrStrava.GetLayerDefn()
Sproj = lyrStrava.GetSpatialRef()
srs = Sproj.ExportToWkt()

#create the output shapefile
if os.path.exists(output):
    driver.DeleteDataSource(output)
```

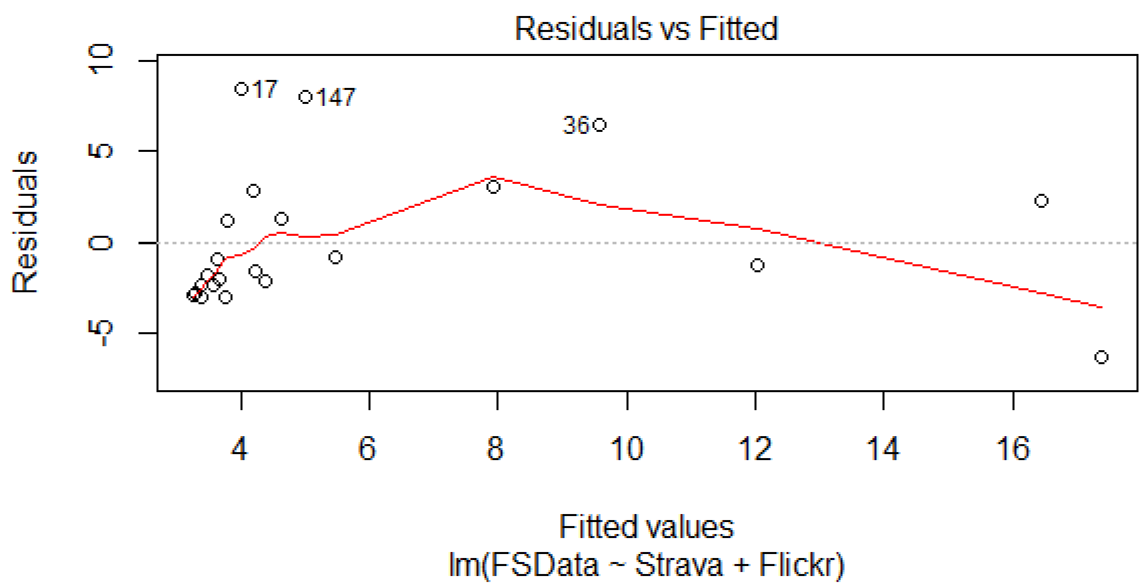
```
dsOutput = driver.CreateDataSource(output)
insrs = osr.SpatialReference()
insrs.ImportFromWkt(srs)
#data_source = driver.CreateDataSource(output)
outlayer = dsOutput.CreateLayer(output, insrs, ogr.wkbMultiPolygon)
outlayer.CreateFields(schema)

outlayer.CreateField(ogr.FieldDefn("Use_Rate", ogr.OFTReal))
outlayer.CreateField(ogr.FieldDefn("SSegCount", ogr.OFTInteger))
outdef = outlayer.GetLayerDefn()
outfeature = ogr.Feature(outdef)


for feature in lyrTrailBuffer:
    for i in range(feature.GetFieldCount()):
        value = feature.GetField(i)
        outfeature.SetField(i, value)
    trailgeom = feature.geometry()
    outfeature.SetGeometry(trailgeom)
    ssegcount = 0
    userate = 0
    for item in lyrStrava:
        stravageom = item.geometry()
        if trailgeom.Intersect(stravageom):
            userate += item.GetField('Use_Rate')
            ssegcount +=1


    outfeature.SetField("Use_Rate", userate)
    outfeature.SetField("SSegCount", ssegcount)
    outlayer.CreateFeature(outfeature)
    lyrStrava.ResetReading()

#Close all open data sets/layers
dsTrailBuffer.Destroy()
dsStrava.Destroy()
dsOutput.Destroy()
```

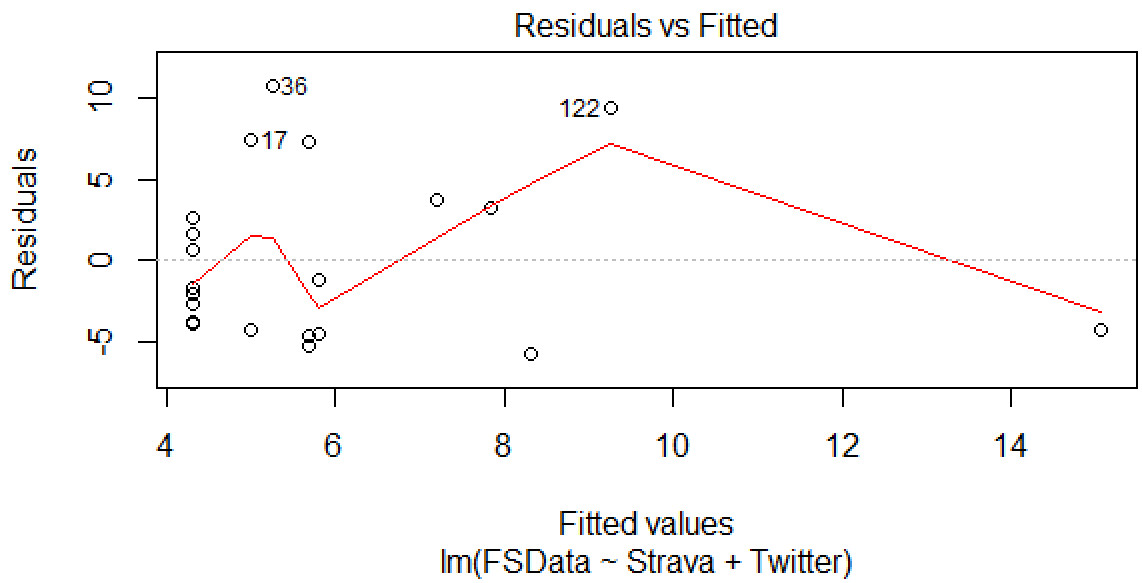
Appendix F: Supplementary Graphs

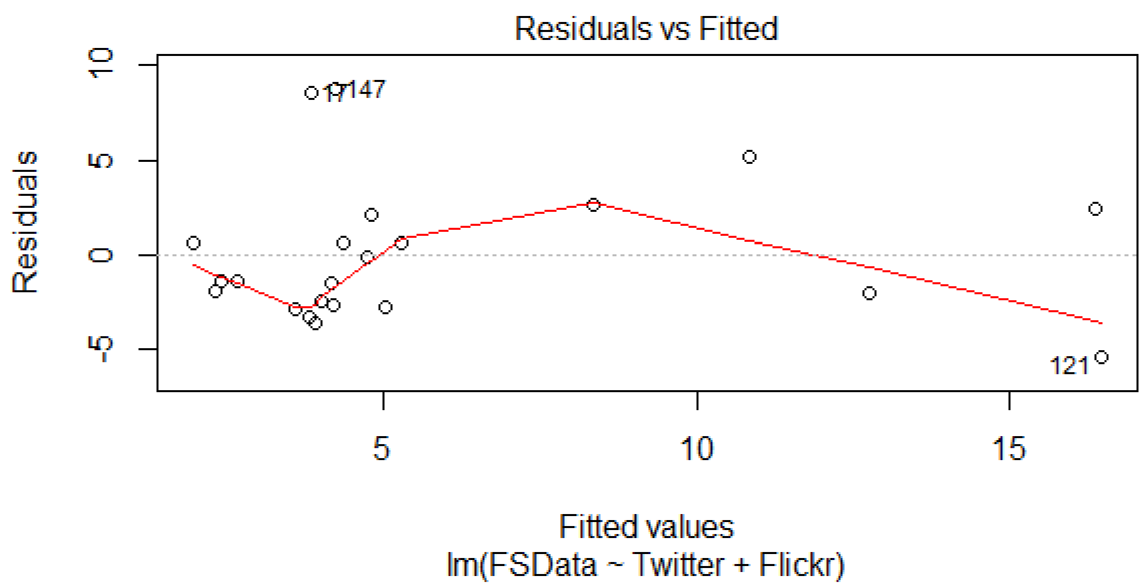




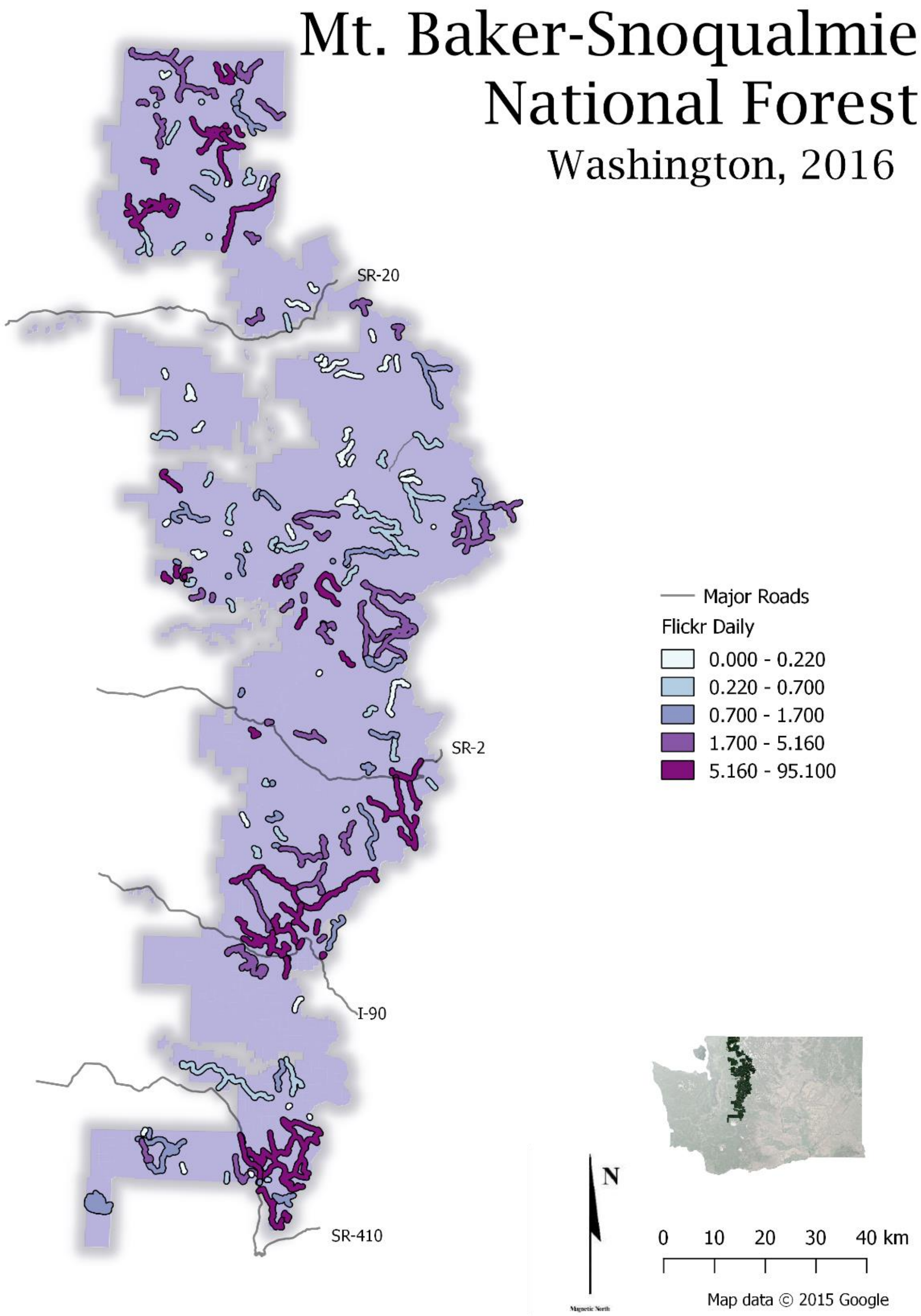

Twitter + Flickr vs. Forest Service Use Rates
Twitter + Flickr
Forest Service
0
10
20
30
40
0
5
10
15

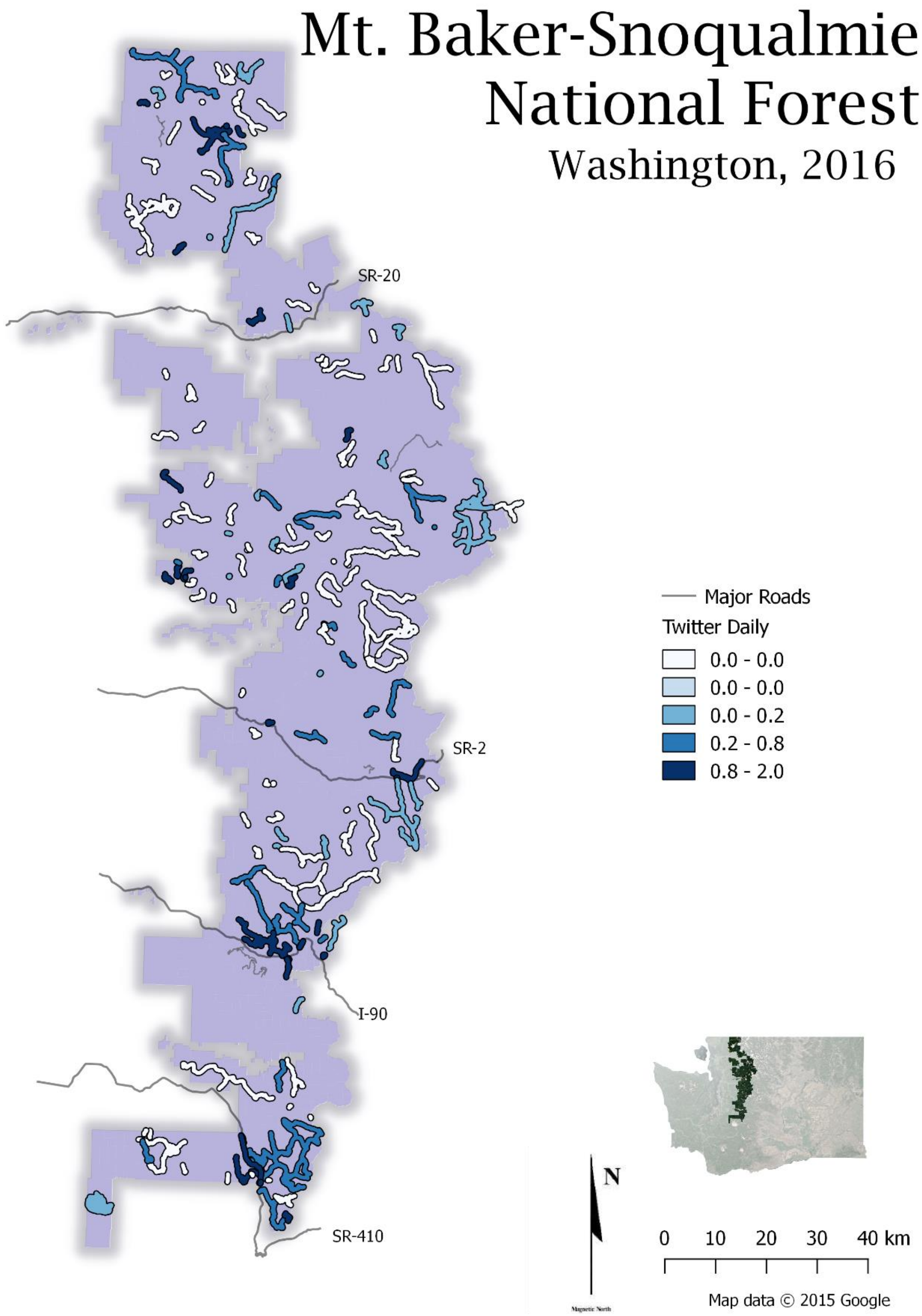

Residuals vs Fitted
Residuals
Fitted values
lm(FSData ~ Strava + Flickr)
17
147
36
-5
0
5
10
4
6
8
10
12
14
16

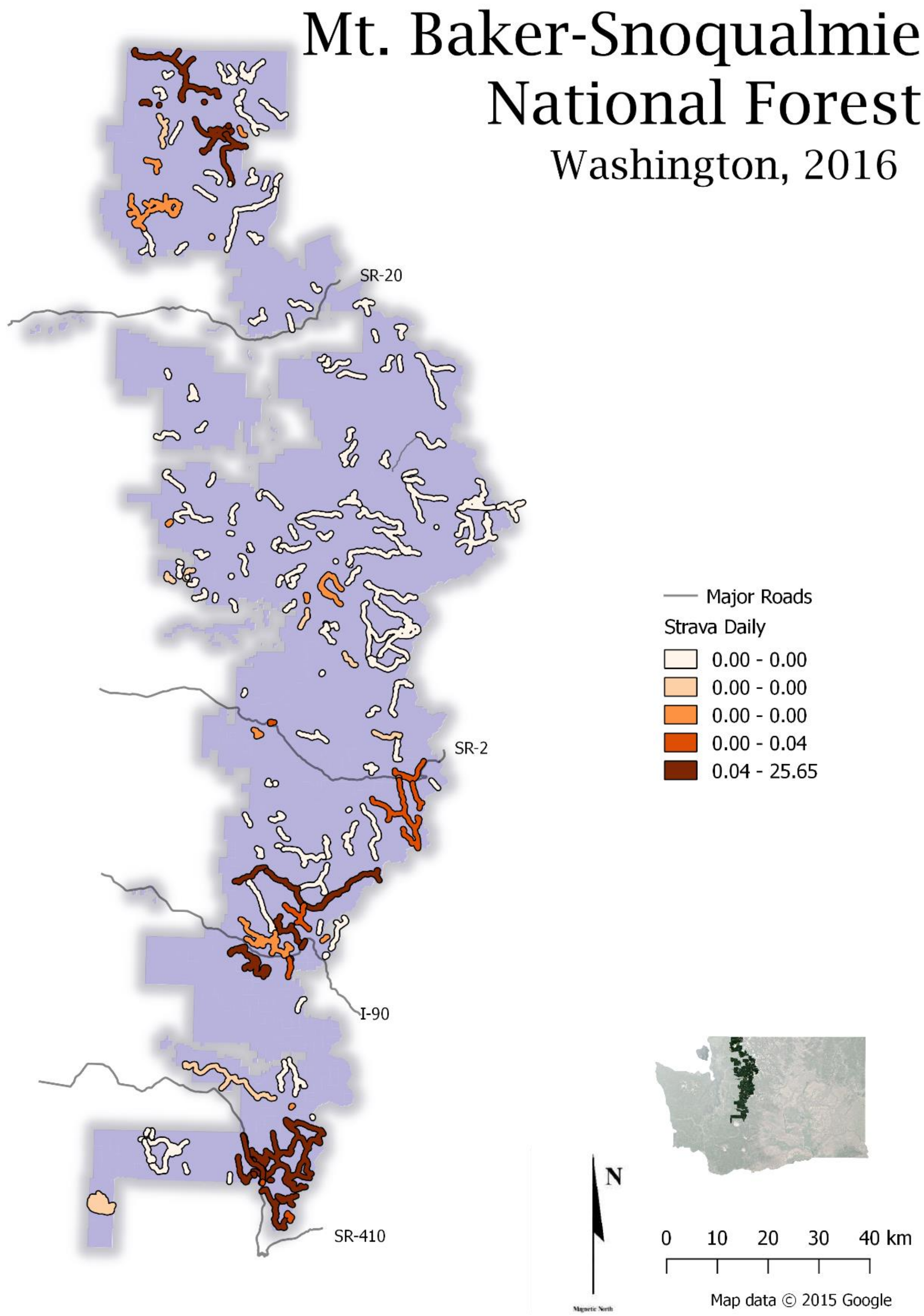

Residuals vs Fitted
Residuals
10
5
0
-5
36
17
122
4
6
8
10
12
14
Fitted values
lm(FSData ~ Strava + Twitter)

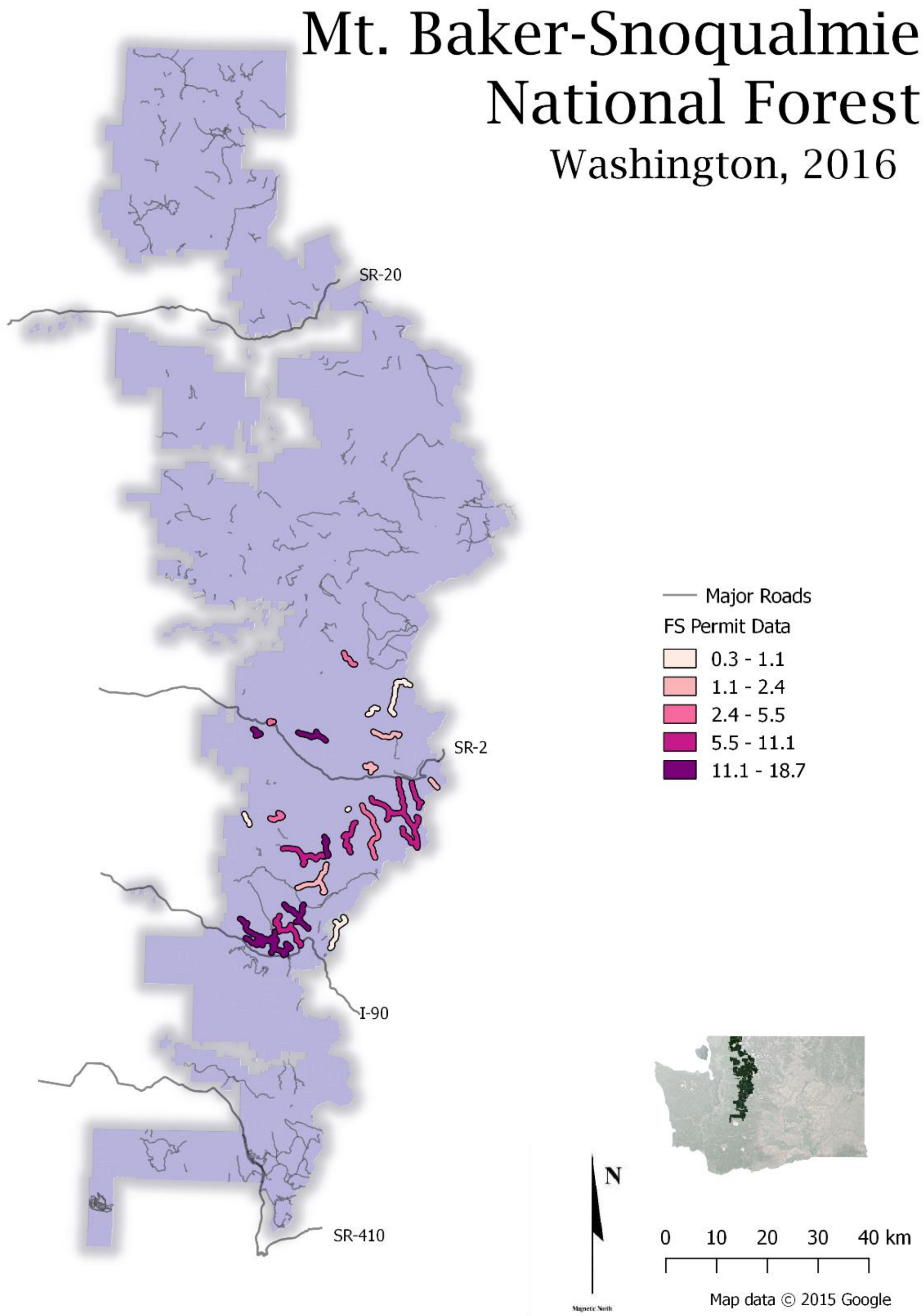

Residuals vs Fitted
Residuals
10
5
0
-5
147
121
5
10
15
Fitted values
lm(FSData ~ Twitter + Flickr)

Appendix G: Trail Data User Days




Mt. Baker-Snoqualmie
National Forest
Washington, 2016
SR-20
SR-2
I-90
SR-410
Major Roads
Twitter Daily
0.0 - 0.0
0.0 - 0.0
0.0 - 0.2
0.2 - 0.8
0.8 - 2.0
N
0 10 20 30 40 km
Map data © 2015 Google


Mt. Baker-Snoqualmie
National Forest
Washington, 2016
SR-20
SR-2
I-90
SR-410
Major Roads
Strava Daily
0.00 - 0.00
0.00 - 0.00
0.00 - 0.00
0.00 - 0.04
0.04 - 25.65
N
0 10 20 30 40 km
Map data © 2015 Google


Mt. Baker-Snoqualmie
National Forest
Washington, 2016
SR-20
SR-2
I-90
SR-410
Major Roads
FS Permit Data
0.3 - 1.1
1.1 - 2.4
2.4 - 5.5
5.5 - 11.1
11.1 - 18.7
N
0 10 20 30 40 km
Map data © 2015 Google

Mt. Baker-Snoqualmie National Forest

Washington, 2016