Paper SAS4098-2020

Solving the VAST Challenge: Three Case Studies Using SAS® Visual Analytics

Riley Benson, Lisa Everdyke, and Karl Prewo, SAS Institute Inc.

ABSTRACT

The VAST Challenge is an annual contest that provides a synthetic data set and a series of questions for teams from around the world to test their existing tools against and to push boundaries by developing novel tools and methods. Teams within SAS have dedicated their spare time to create entries to the VAST Challenge for several years. This paper explores three of those entries showing how SAS® Visual Analytics, other SAS tools, and open-source software were combined to tackle the challenges and show how the challenge has helped uncover areas for growth.

INTRODUCTION

VIS (IEEE Visualization Conference, http://ieeevis.org) is the premier academic conference on the topic of data visualization. Visual Analytics Science and Technology (VAST) is the sub-conference of VIS specific to visualization and interactive techniques applied to any domain requiring data analysis. The VAST Challenge is a workshop event, hosted at the VIS conference, where contestants present their solutions.

Teams from SAS have entered the VAST Challenge several times because it provides an excellent resource for testing our products against a realistic problem and often shows us opportunities for growth that may be beyond current customer expectations.

This paper shows the value of using test data with established ground truth in your own work and will detail how SAS Visual Analytics and other tools were used to create three VAST Challenge solutions. Throughout the paper we will highlight techniques and improvements used to tackle the challenges that you can use in your data exploration.

THE VAST BENCHMARK REPOSITORY

All of the data sets used in the VAST Challenges remain available as part of the VAST Benchmark Repository along with all solutions submitted each year. The data variety spans from time series and text stream data to audio recordings and theme park maps. The domains are also varied but often involve trying to uncover nefarious activity or explain a complex situation. The challenges are all inspired by proposals submitted from sponsors, so the data always has its roots in a real-world problem that an organization needs solved.

The most valuable aspect of the challenge data is its embedded ground truths. This is created by the challenge organizers using simulation software to orchestrate a scenario from which the data sets are then collected. Important events and distractions are woven into the simulation along with errors and uncertainty in the collection of the data.

2015 MINI-CHALLENGE 1 – DINOFUN WORLD

The 2015 challenge scenario introduced the fictional DinoFun World theme park and required tracking a crime committed during a weekend when a local sports celebrity was going to make a special appearance. The first mini-challenge provided location data on each visitor from devices either provided by the park or through an app installed on visitors' phones.

From this data the goals were the following:

1. Find types of park visitors who had similar behavior and explain both how they tended to experience the park and what could be done to improve their experience.
2. Describe the typical behavior and the changes over three days' worth of data.
3. Identify any specific anomalous events along with potential explanations.

For a quick video overview of the SAS team's entry, see: https://youtu.be/BIPscav27Js.

PREPARING PARK VISITOR DATA

The data came in the form of a large table with time-stamped position data for each visitor and a map showing the boundaries of the attractions in the park.

Timestamp	Id	Type	X	Y
...	...	...	...	...
6/8/14 10:13	436	movement	83	45
6/8/14 18:36	436	movement	56	31
6/8/14 9:48	436	check-in	0	67
...	...	...	...	...
6/7/14 10:01	941	movement	65	47
6/8/14 8:57	941	movement	26	16
6/6/14 13:49	941	check-in	47	11
...	...	...	...	...

Table 1. Sample of Data for Movement in the Dino Fun Park

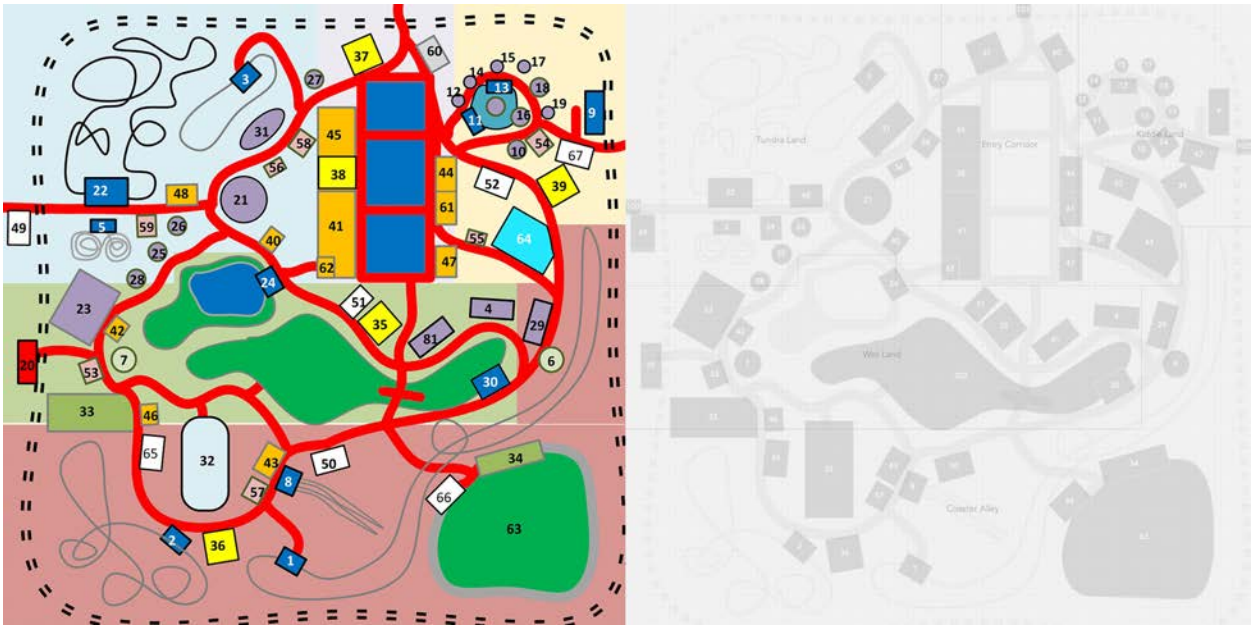

Figure 1. Provided Map of Park Attractions and Simplified Version of Map

To explore this data, it first had to be structured in a conventional tabular form. Python was starting its rise in popularity as a language for data science, so we decided to tackle the problem using both Python and SAS Visual Analytics.

The map was converted into an SVG file, which allowed the code to assign the most likely attraction to each position where a check-in was indicated. There were also places in the

park where users might spend a lot of time, but they didn't explicitly check-in. Stops at places like the restaurants, bathrooms, gift shops, and first aid locations had to be intuited by looking for long pauses without movement from the visitor.

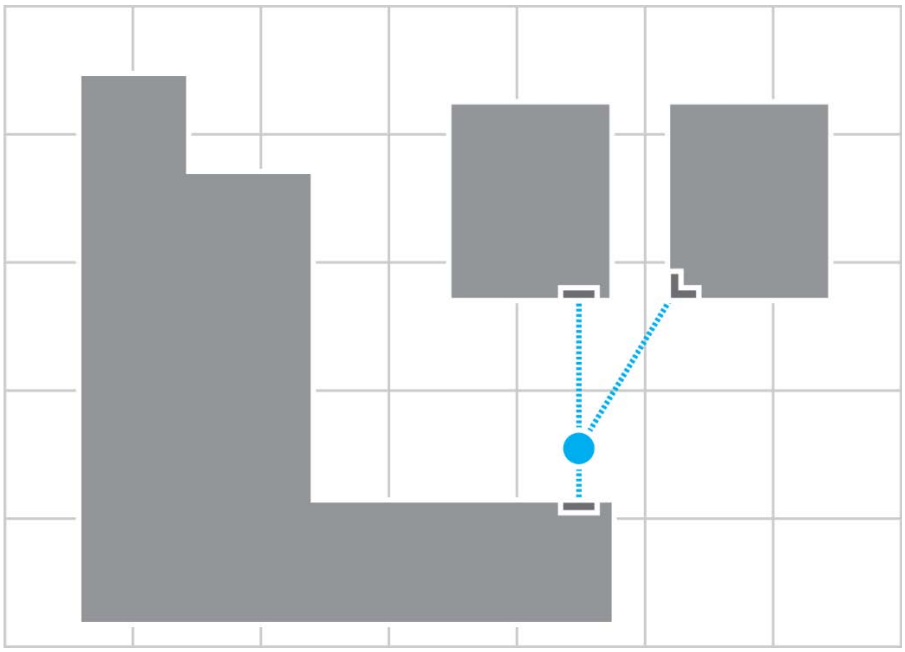

Figure 2. Illustration of Attraction Mapping by Finding the Nearest Polygon Edge

Some of the required data manipulations, like measuring the distance traveled for a park visitor or the time spent waiting at each attraction, were done using calculated items in SAS Visual Analytics. Other calculations were performed in Python with the goal of simulating a team where a Python expert and SAS Visual Analytics expert were collaborating.

> TIP: Always prepare data with as few tables as possible. This not only helps for a more efficient experience in SAS Visual Analytics, but also aids collaboration between tools.

Two tables were created. One table tracked detailed movement data for a visitor and t remained very similar to the source data. that the other table focused only on identified check-ins to attractions and other buildings.

Timestamp	Id	Attraction	Region	Wait (s)	...
6/7/14 10:01	941	Main Entrance	Entry	170	...
6/8/14 8:57	2672	Wrightraptor Mountain	Coaster Alley	649	...
6/6/14 13:49	4343	Kauf's Lost Canyon Escape	Tundra Land	1428	...
...	...	...	...	...	...

Table 2. Sample of Data Summarized to Only Attraction Check-ins

For this challenge, we kept data synchronized between Python and SAS Visual Analytics using manual re-imports of the data files.

EXPLORING BEHAVIORS IN THE PARK

Once the data and additional metrics were prepared, we used SAS Visual Analytics to examine the distributions of the metrics, find visitors that were outliers, find temporal patterns, and identify high frequency paths.

The output from this investigation was a set of time ranges, visitor IDs, and attractions of interest. Many times, a visitor or group of visitors was identified as an outlier in some metric, perhaps due to how long they spent waiting at a particular attraction, and we would then plot their path using the map view or path analysis to get further details.

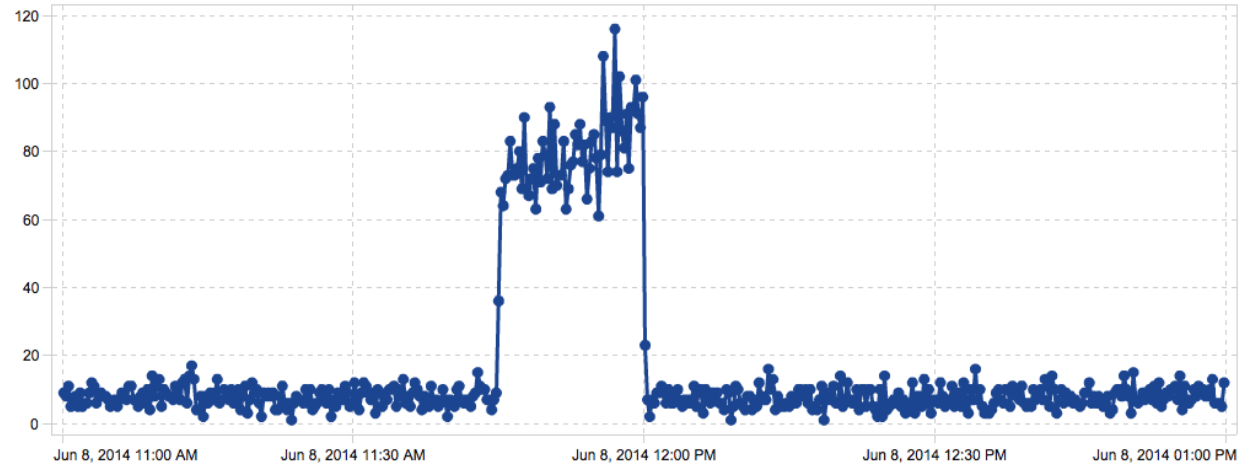

Figure 3. Time Series of Activity Frequency That Highlights a Sudden Increase on June 8th, 2014

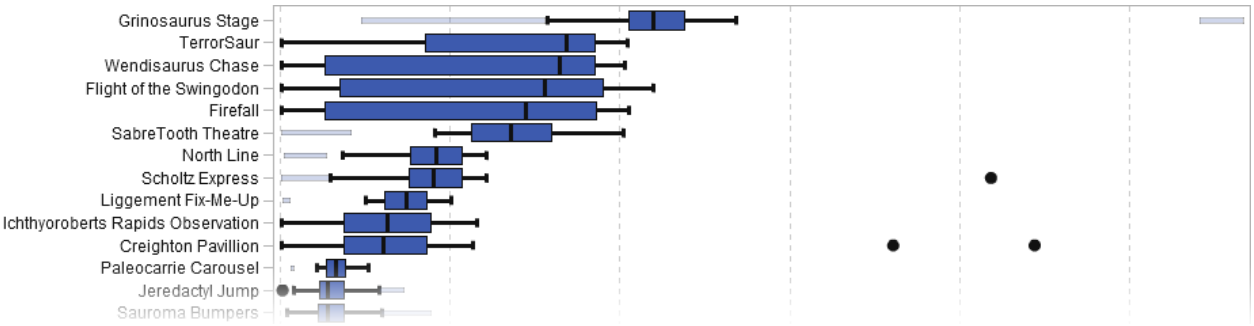

Figure 4. Sample of the Box Plot of Check-in Durations for Attractions

The path analysis object in SAS Visual Analytics was primarily intended to show how customers move through a company's process. Call center flows or e-commerce website traffic, for example, use this type of analysis often. We repurposed it to find groups of visitors who traveled together throughout the day.

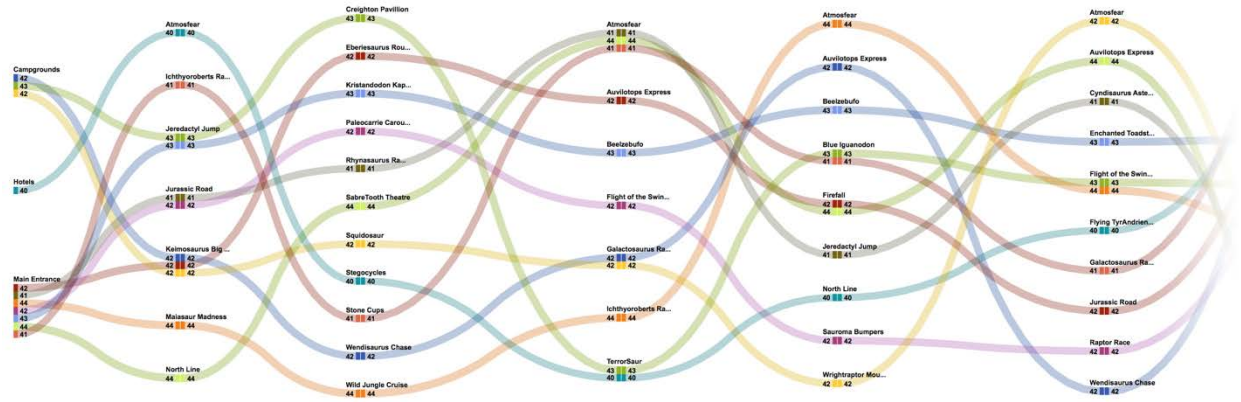

Figure 5. Path Analysis Output by Attraction for Large Groups

The output from the path analysis is visualized as a Sankey diagram, but this method does not scale well to the many thousands of visitor paths since it can only aggregate paths between attractions that are identical. To reach interpretable results, we had filter to a subset of visitors or only track paths through a more highly aggregated set of groups, such as the type of attraction or region of the park.

> TIP: When using Path Analysis start by visualizing paths through the highest levels of aggregation before breaking the paths down into more detail.

At the time of this challenge, SAS Visual Analytics did not have support for custom map providers. Therefore, all of the geospatial work was also done using Python scripts. Scatterplots and animated heat maps provided overviews of visitor activity to accompany the more analytical visualizations.

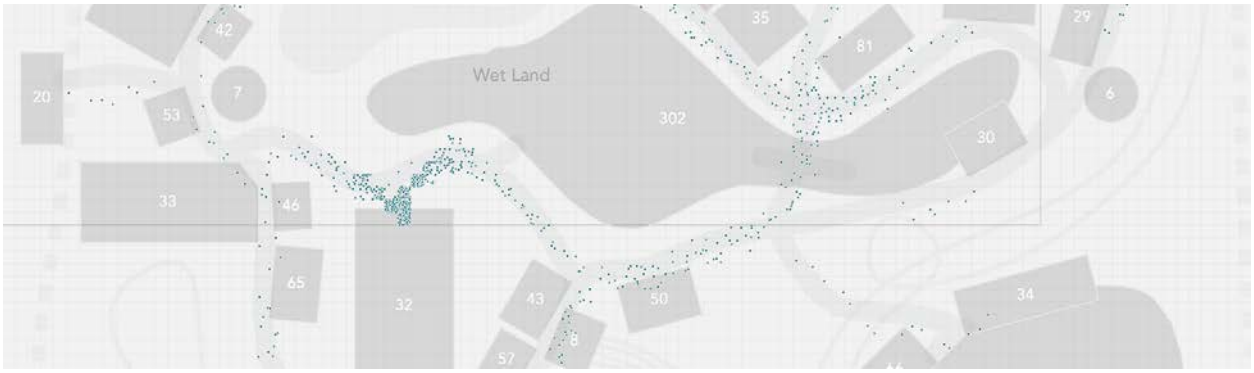

Figure 6. Scatter Plot Showing Visitor Location During Anomaly

Rendering a single individual's path throughout the park was an unexpected challenge. Visitors tended to walk across the same path many times and that generated a lot of over-plotting. To help counter this we introduced jitter to each of the movement points for a visitor within a five-meter square to reduce the chance of overlap and as a bonus this helped represent some of the positional uncertainty present in the data.

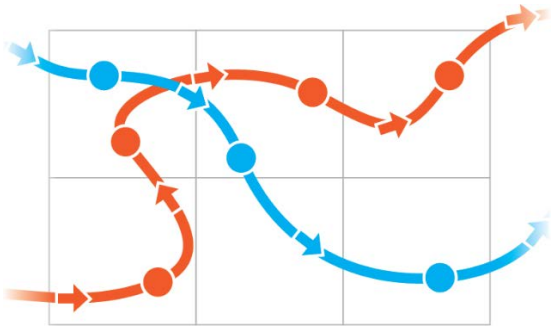

Figure 7. Visitor Path Through Park Shown Using Directed LineSets

The jittered points were connected using Bezier curves that remained continuous throughout the reported movement points in the path being visualized. The intent was not to draw the path as the visitor might have taken it, but instead to provide a curve that was easy for the eye to follow along a leg of the journey. This relied on the same concept used in the LineSets visualization. Arrows were added to show directionality and segments were divided based on check-ins and color-coded for easier identification.

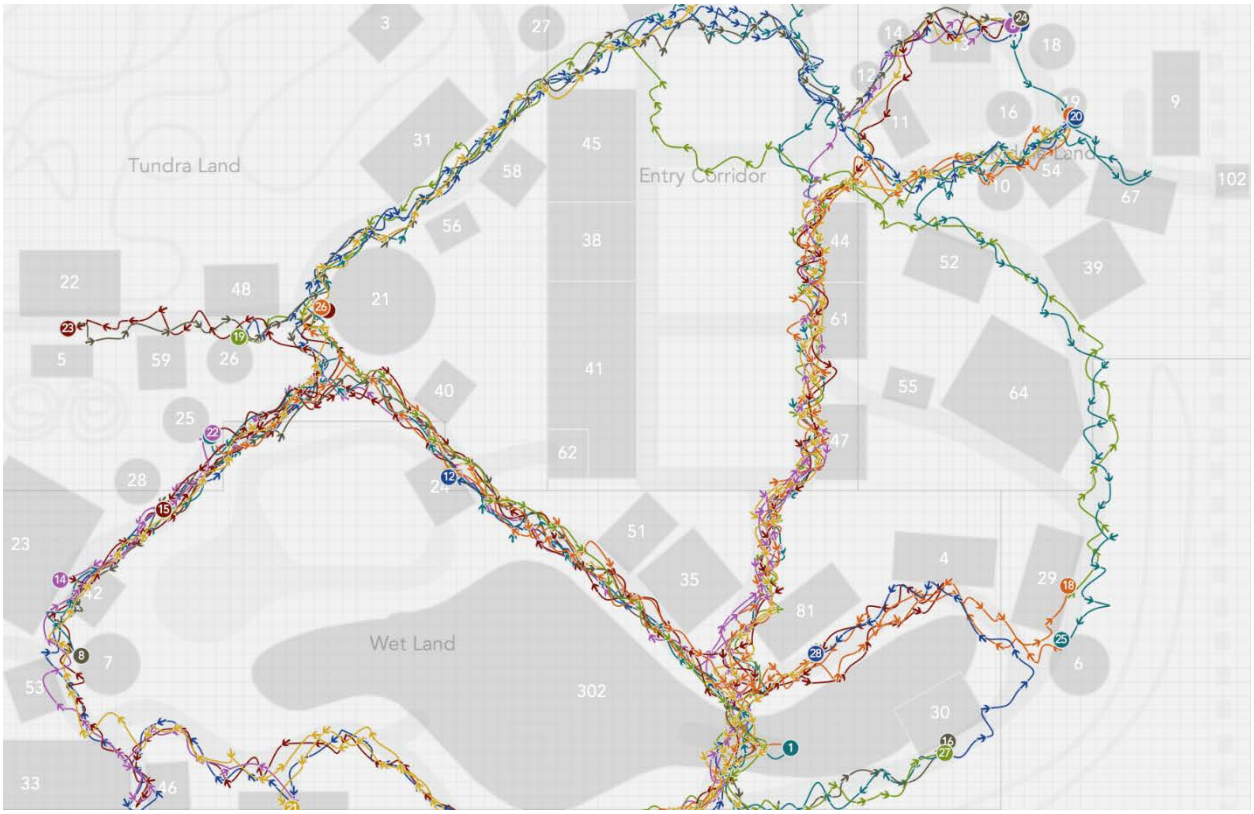

Figure 8. Visualization Showing All of the Visitor Travel Paths

SOLVING THE THEME PARK CHALLENGE

Combining the high-level results from our analytics with the detail visualizations we were able to identify visitors of interest related to the park vandalism, but we were unable to uncover anything other than circumstantial evidence showing that they spent a great deal of time at the Creighton Pavilion sometime Sunday morning. The crime was first discovered by the general public at 11:45 AM and the park authorities seem to become aware of the crime at 12:00 PM. We based these assumptions on large communications spikes in the data involving visitors clustered near the entrance to the pavilion.

We focused on describing broad types of visitors who had similar attraction attendance in the park, instead of focusing on groups that moved in similar ways. This approach allowed us to describe possible improvement to the park in terms of visitor preferences. We assigned the names The Masses, Audience Members, Thrill Seekers, Wanderers, and Small Fries to the clusters that were found by examining attraction type frequency distributions. We also identified visitors who were members of a very large group that traveled together and referred to them as Herds.

WHAT WE LEARNED

We were unable to perform more robust geo-temporal clustering. The path analysis analytic object is only able to detect groups that checked in at the same attractions through the entire day and standard Clustering cannot directly account for the sequenced nature of movement data. This led to us missing 30-40% of the group behaviors embedded in the data.

The other weakness in our process was the separation between identifying visitors of interest using SAS Visual Analytics and then visualizing the movement of those visitors using our Python script. This separation between the analytics process and the custom

visualization slowed down our ability to iteratively explore events. In this case, we were able to find the visitor ID of the culprit for the theft and vandalism but were unable to piece together the exact timeline of the crime due to this disconnect.

We were able to successfully explore a large amount of time series data and compare the distributions and outliers of any metric easily due to the nature of data assignment and auto charting systems in SAS Visual Analytics, which validated their value to the data discovery process. This allowed us to provide more detailed insights into possible efficiency improvements, such as the fact that visitors who want to ride the more exciting rides are forced to walk faster throughout the park in order to make it to all of them.

2017 MINI-CHALLENGE 1 – THE PIPITS KICK IT

The 2017 challenge asked participants to study the movement of vehicles through a fictional nature preserve to look for the cause of a drop in the local population of the Rose-crested Blue Pipit. This was similar to the 2015 challenge except that the position of the vehicles is only known when they pass through a gate that records their RFID tag number.

From this data, the goals were the following:

1. Find and explain the various patterns of activity seen within the preserve.
2. Describe the longer-term changes and patterns found in the data.
3. Identify any unusual patters in the data, especially those that could be responsible for declining bird populations.

For a quick video overview of the SAS team's entry, see: https://youtu.be/sOFfqlxVFi8.

PREPARING VEHICLE MOVEMENT DATA

The source data for the challenge provided four columns that tracked when each vehicle passed through a gate and tracked the type of the vehicle based on weight and axle count.

Timestamp	Vehicle Id	Vehicle Type	Gate Name
5/1/15 0:43	20154301124328-262	4	entrance3
5/1/15 1:03	20154301124328-262	4	general-gate1
5/1/15 1:06	20154301124328-262	4	ranger-stop2
5/1/15 1:31	20153101013141-937	1	entrance3
…	…	…	…

Table 3. Source Data for the Challenge

We again started with Python for our data preparation to continue with the tradition of simulating a team with mixed tool preferences. Having a more general-purpose language made it easier to extract a vector version of the preserve's map and calculate metrics, such as the distance between gates using the actual shape of the roads.

To enable a wide variety of exploration in SAS Visual analytics, we created three different tables: one where each row was a gate check-in, another summarized so that each row represented a single vehicle, and another represented the park's gates as a network, such that each row was connection between gates.

Timestamp	Vehicle Id	Type	Axles	Ownership	Gate	…
5/1/15 1:03	262	Truck	3	Park	General Gate 1	…
5/1/15 1:06	262	Truck	3	Park	Ranger Stop 2	…
5/1/15 1:31	937	Car	2	External	Camping 6	…
…	…	…	…	…	…	…

Table 4. Prepared Gate Sensor Data

Vehicle Id	Type	Time in Park (s)	Gate (arrive)	Gate (depart)	...
797	Truck	3332	Entrance 3	Entrance 2	...
937	Car	482043	Entrance 2	Entrance 4	...
1033	Bus	221469	Entrance 4	Entrance 0	...
...	...	...	...	...	...

Table 5. Prepared Vehicle Summary Data

From Gate	To Gate	Distance	Frequency	...
General Gate 1	Ranger Stop 2	26	7248	...
General Gate 4	Entrance 3	129	1	...
Ranger Stop 0	General Gate 2	33	7311	...
...	...	...	...	...

Table 6. Prepared Gate Network Data

For this data, we made use of the new Python SWAT library to automatically upload the processed data to CAS after each run of the Python data prep code.

EXPLORING TRAFFIC IN THE PRESERVE

Using SAS Visual Analytics, we examined the seasonal and daily traffic patterns for each type of vehicle: cars, buses, heavy trucks, and park-owned trucks. The ability to create small multiple time series was especially effective in quickly revealing interesting differences in activity.

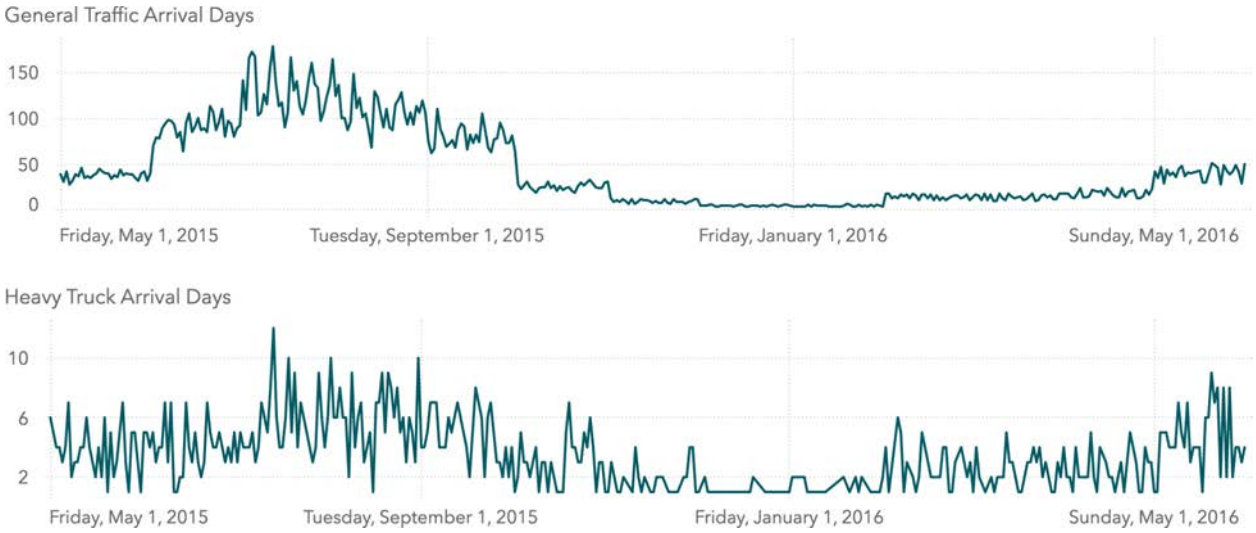

Figure 9. Example of a Time Series Showing Preserve Traffic

Another effective technique was to use a schedule plot or a needle plot as way to visualize the timeline for an individual vehicle's behavior. Especially if aligned with a table showing the detail information for a vehicle it provided an effective comparison of the activities between multiple vehicles. Comparisons of vehicle could reveal if vehicles were potentially meeting within the preserve or just had similar patterns of behavior.

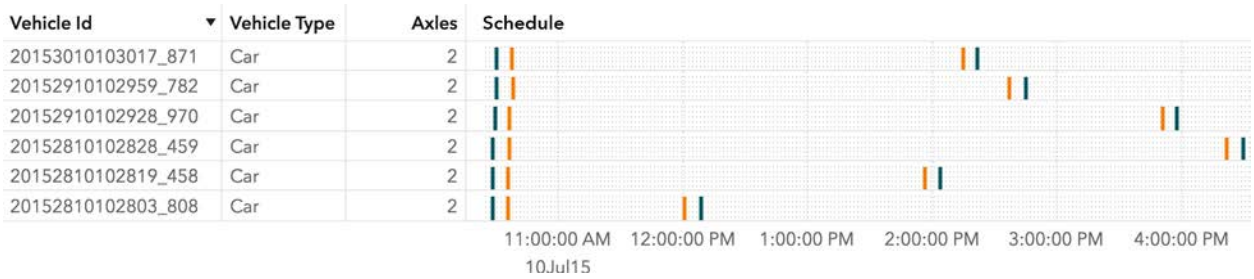

Figure 10. Detail Table with Aligned Schedule

> TIP: If you work with a fixed number of rows in a table, you can line up an adjacent graph and size it vertically to where it matches closely enough for exploratory needs.

Given that the preserve map and road paths were in an imaginary coordinate system and that even with the new support for custom polygons we couldn't easily represent road paths, we decided to use the new data-driven content system in SAS Visual Analytics to create a map visualization that supported changing the color and thickness of the road segments and the gates.

Figure 11. Example Roadway Map Data-Driven Content Objects

SOLVING THE NATURE PRESERVE CHALLENGE

Our analysis of the activity patterns discovered that traffic stayed consistently high from 6AM to 6PM every day and exhibited seasonal patterns where the summer months, June through September, accounted for the majority of through-traffic. Visitors to a campsite within the preserve diminished during the colder months and saw a bump in utilization on the weekends, as one would expect.

Looking for anomalies in the more detailed data we were able to find several interesting events. One camper in the preserve had been moving between campsites for nearly a year straight without ever leaving, staying about a month at each site. We also uncovered occasions where several vehicles would meet at the same location or would revisit the preserve at regular intervals.

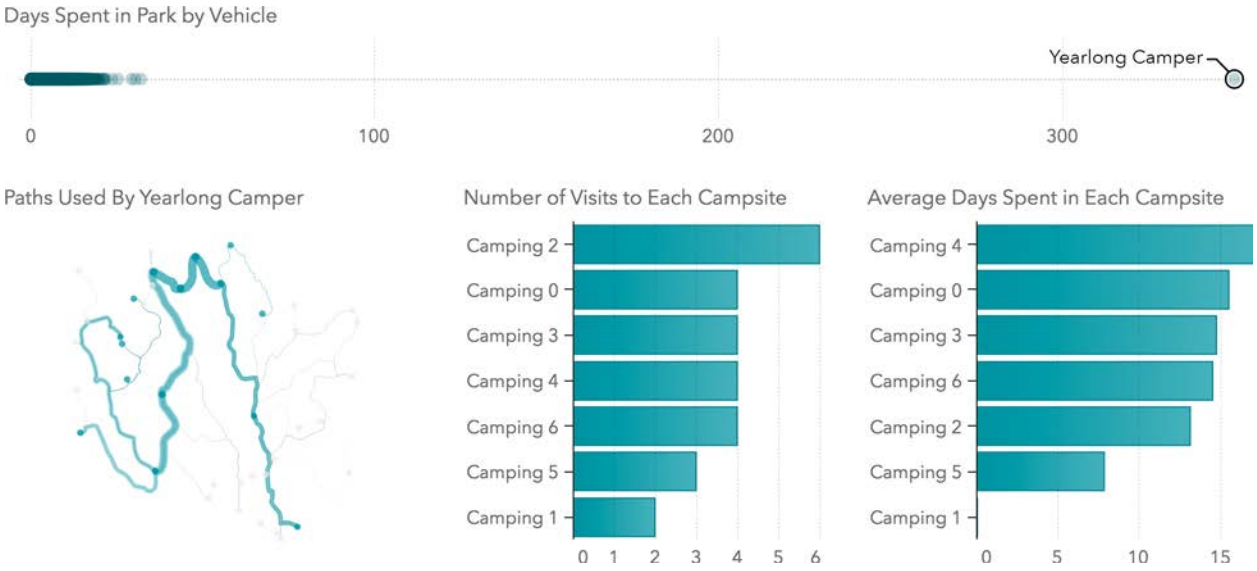

Figure 12. Visualization of the Year-Long Camper's Behavior

The biggest discovery was the identification of heavy trucks that entered from the city entrance, passed through a few restricted gates, and paused for a while at a particular campsite in the preserve before returning. This happened once or twice a week and always in the early morning when park vehicles had all returned to base. We hypothesized that this was an illegal activity such as waste dumping, which turned out to be true.

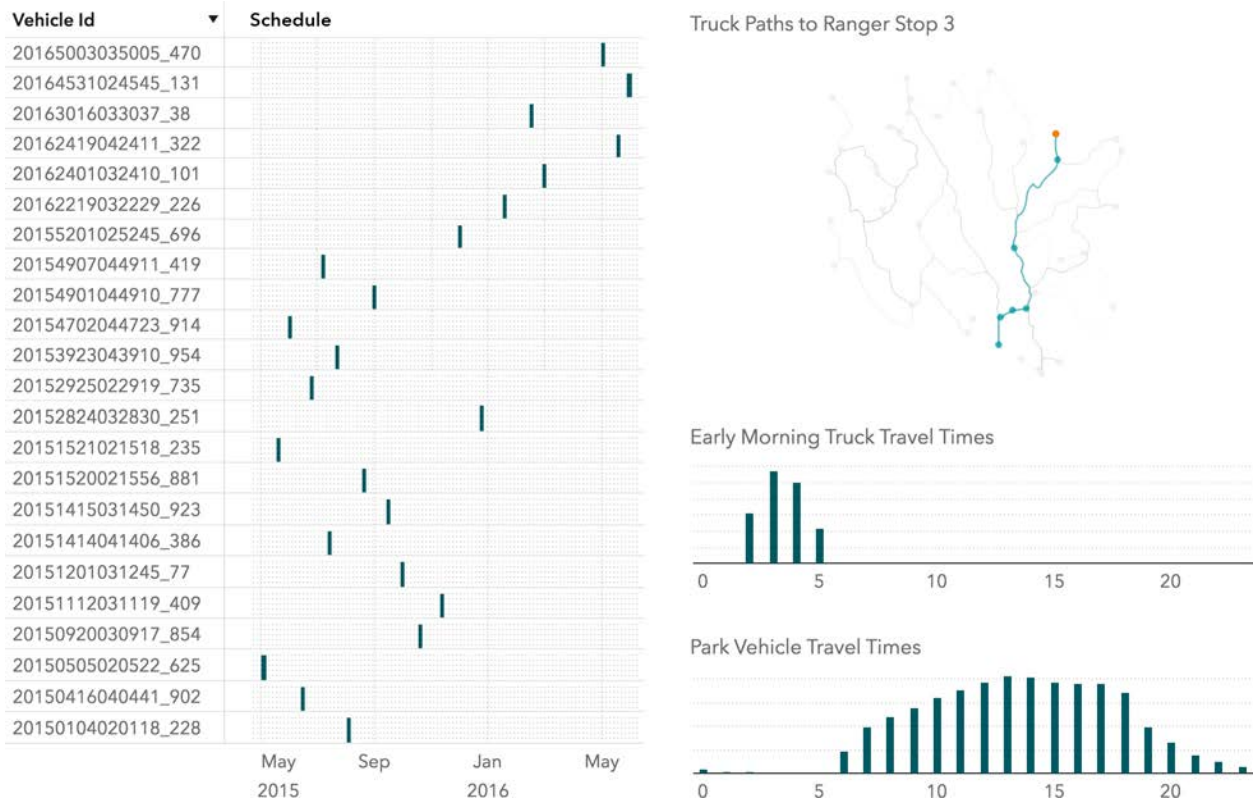

Figure 13. Visualization Showing the Early Morning Waste Dumping

> TIP: Schedule plot can provide a unique way of viewing event data by discretely dividing the events. It can also sort the events by a category instead of by time.

WHAT WE LEARNED

The addition of the data-driven content object to SAS Visual Analytics greatly increased the efficiency of the exploratory process once the custom visualization was integrated. Having easy access to a specialized visualization led us to try to apply it in more creative ways that we would have if it required moving the data out to another tool.

We only missed one major anomaly in the data. Two cars drag raced through the preserve and could be detected by their abnormally high-speed traversing between the gates. We did look for outliers in the travel speed but because we looked for outliers in the average speed of the vehicle on its entire journey a momentary outlier on a single gate check-in went unnoticed. Having a way to more robustly check for outliers across various groupings and time windows would have helped in this case.

2019 MINI-CHALLENGE 3 – EARTHQUAKE IN ST. HIMARK

The 2019 challenge was for the fictional city of St. Himark. It asked participants to reconstruct the timeline of a major earthquake disaster and to provide a system that could be used by first responders to more quickly deploy assistance to the correct areas of the city. The third challenge provided data from the Y*INT social media platform of all the public posts by city residents with usernames, timestamps, and neighborhood information.

From this data, the goals were the following:

1. Summarize conditions in the city both 5 hours and 30 hours after the earthquake.
2. Find at least three moments when the priorities of the city's first responders should have changed.
3. Provide a general assessment of the morale and conditions of the city's residents throughout the earthquake.
4. Account for and include uncertainty in as much of the analysis as possible.

For a quick video overview of the SAS team's entry, see: https://youtu.be/JHrUbod70-E.

PREPARING SOCIAL MEDIA POST DATA

The data gave only the context of the sender, the time, and the approximate location of the sender.

Time	Location	Account	Message
4/6/20 0:04	Broadview	RasoHorse49	billeeeeer, i miss ytouou !
4/6/20 0:07	West Parton	CuriousPlateBobbie	You obviously need to use rumble! #rumble #toWonder
4/6/20 0:11	Old Town	Moore1961	I hate all the fawning apple-johns lying around...
...	...	...	...

Table 7. Source Data for Y*INT Social Media Messages

For the 2019 challenge, we used only Python to extract the @mention network and hashtags from the messages. Almost all of the structuring of the data was done using SAS Visual Text Analytics. We used many of the pre-built concept detection models and also built additional concept that were able to specifically detect outages in utilities like power and water as well as determine if the report was direct or indirect. For example, "I heard that the power was out" would be classified as indirect while "my power is out" is direct. We extracted sentiments using SAS Visual Text Analytics but we used SAS Visual Analytics to detect and extract topics.

Text analytics output is often difficult to fit into a tabular form. It would be convenient if each message cleanly discussed only one event, but it isn't uncommon for a message to potentially mention power outages, along with drinking water problems and structural damage. Because of this, detected concepts have to be represented by their own column in the data, the same way that topics are derived in SAS Visual Analytics.

Id	Tag	Mention Id	Reposts	Outage	Fatality	Collapse	...
25917		27831	0	power			...
26024	NewsWaste	30123	12		Naomi		...
40464	Rumble		4			hospital	...
...	...	...	...	...	...	...	...

Table 8. Sample of Structured Text Data

For this challenge, the data was uploaded as a SAS data table for use in SAS Visual Text Analytics. The transposing and joining of the structuring of the text was done in SAS code before being loaded in CAS.

EXPLORING THE DISASTER TIMELINE

The temporal nature of the data was explored using time series that show the frequency of messages about each of the concepts as detected by the SAS Visual Text Analytics model. Topics were also derived using the text topics object in SAS Visual Analytics and topic relevance used a time series object as well.

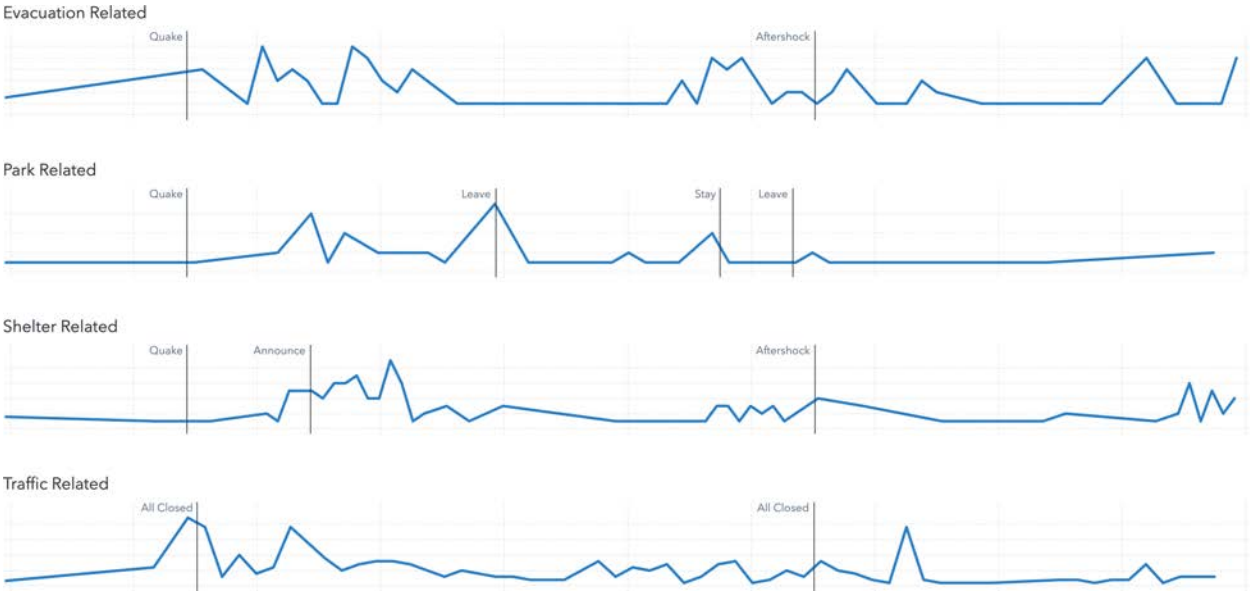

Figure 14. Time Series of Extracted Topic Concept Relevance

> TIP: Use labeled reference lines to show important events. This often provides better context than an axis showing precise dates and times.

The network formed from the mentions within the messages was displayed using network analysis, which revealed the subgroups and central influencers within the city's social network. A bar chart that compared users by how frequently their messages were reposted helped reveal who was often the source for information shared among citizens.

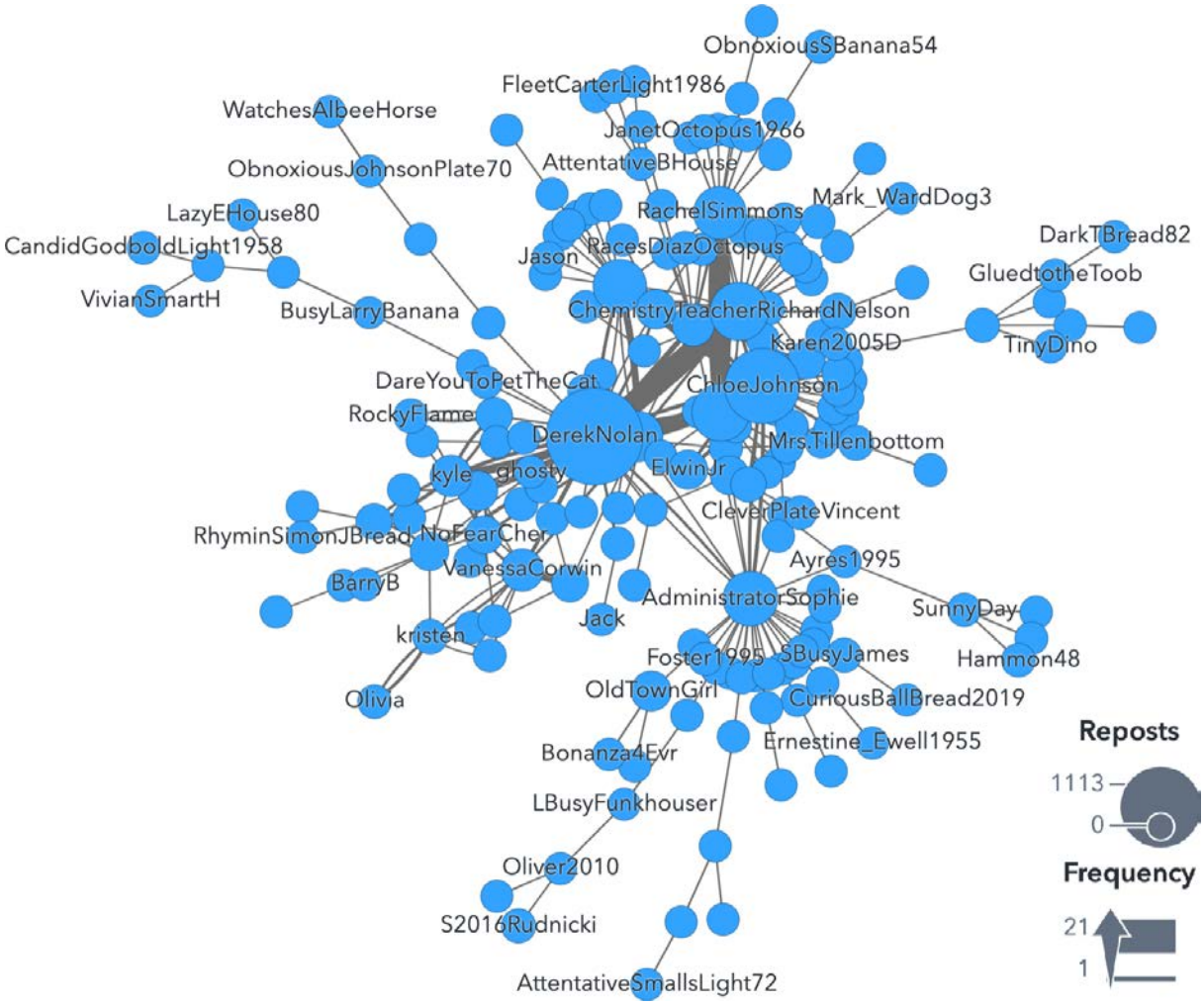

Figure 15. Network Analysis of the City's Core Social Network

To analyze the spatial aspect of the data, the provided map was converted into an SVG and used as a choropleth via the data-driven content object. We first created a map by using custom shape files used with a geographic map object, but this didn't allow as polished a presentation since we needed the roads and neighborhood names overlaid.

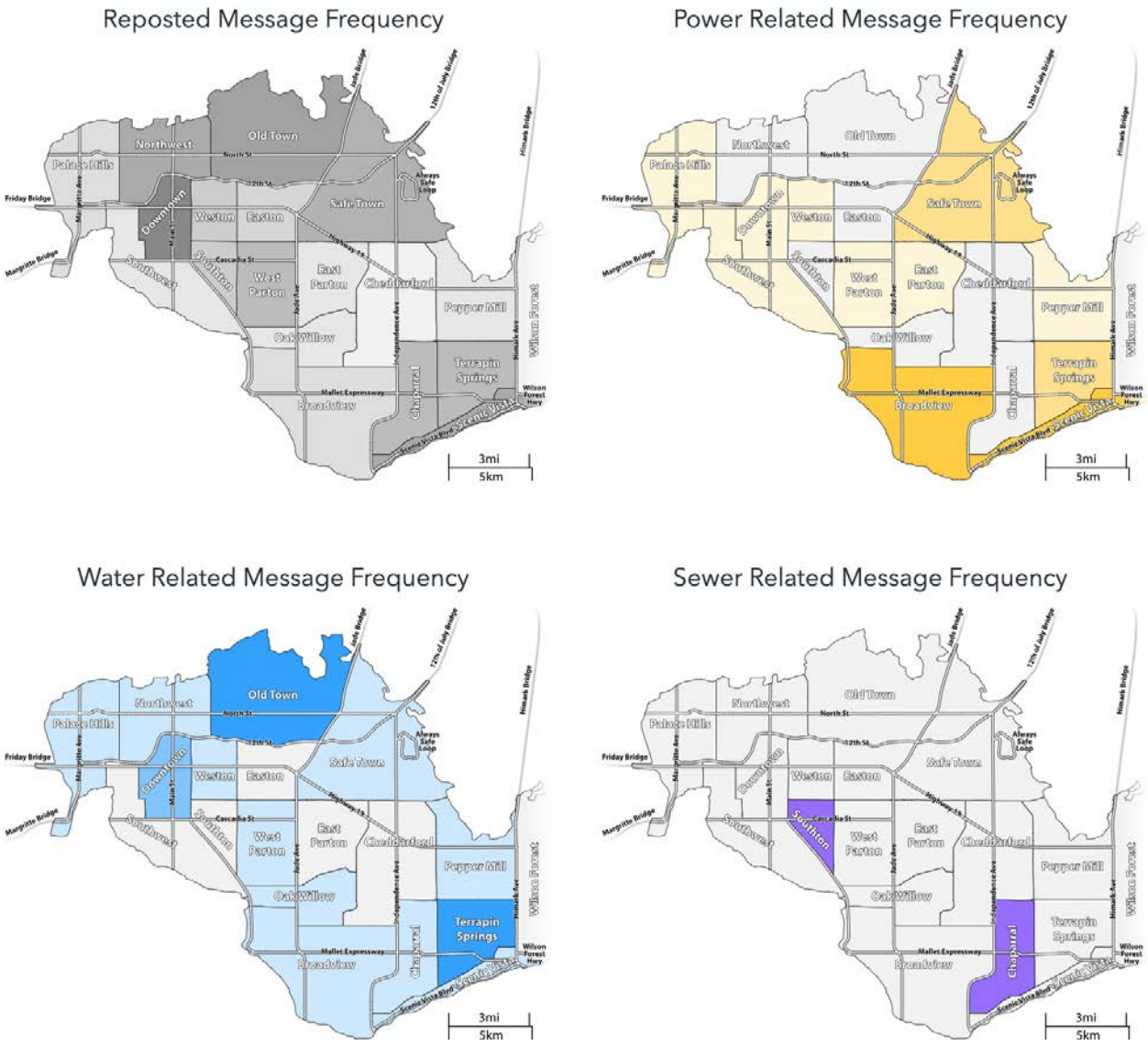

Figure 16. Maps Showing Location of Outage Reports

We needed to be able to color the map so that it would match the colors used for different types of problems in the city. We used standard URL parameter formatting to pass the desire colors to the data-driven content object, which allowed for it to be modified without leaving SAS Visual Analytics. Example of the URL style used:

```
https://www.sas.com/path/to/content/map.html?c1=5a2bc&c2=7492ae
```

Within the data-driven content object's code, these parameters can be retrieved by parsing the `window.location.search` variable.

> TIP: Use URL parameters as a convenient way to customize data-driven content objects.

The more detailed views were achieved using schedule plots of the individual message timings, which helped to follow conversations between multiple users or track down the original mention of an emerging problem.

SOLVING THE EARTHQUAKE CHALLENGE

Since the challenge asked for evaluations relative to when the earthquake occurred, we first had to discover when the earthquake actually occurred. The message data had official announcements for the foreshock and mainshock of the earthquake, but not for the aftershock. Searching for users commenting on “shaking” furniture or “vibrating” objects could provide a more robust and crowdsourced way to detect all earthquake events, but rules had to be created because the data contained misleading messages from advertisers regarding “vibrating deals” or “rumbling sales”.

Figure 17. A Time Series Plot Showing the Earthquake Related Messages

Once the timing of the earthquake was determined, we could then focus on evaluating the state of the city at both 5 and 30 hours. We were able to find evidence for broken sewer lines and contaminated water supply, as well as areas with an increase in power outages.

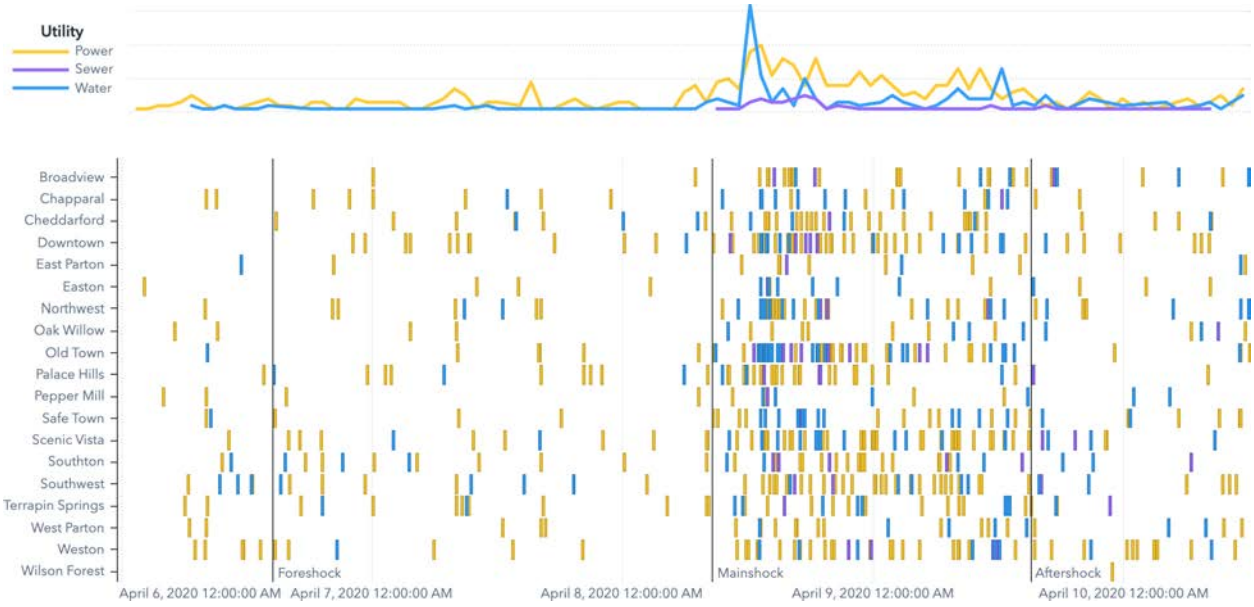

Figure 18. Overview of All of the Utility Outage Reports

> TIP: A single SAS Visual Analytics object cannot show both detail and aggregated data. However, two objects can be aligned to achieve the same effect.

Using text topics over a restricted time window often provided different results than the topics that were the most important to the entire corpus. Using this technique, we were able to discover a discussion around using the city parks as a place to set up tents away from

compromised buildings and another discussion about meeting plans for hobbyist scientists to check for radiation from the local nuclear power plant.

> TIP: When exploring text data, look at both topics found in the entire corpus and in small segments of interest, such as in a time window.

Using the repost counts and mention network, we identified key sources of information in the city, many of which were unreliable despite being the accounts of news hosts or users with "Administrator" in their usernames. There were also a lot of smaller discussions between users trying to reconnect with family or users fighting over whether the city was trying to cover up problems with the local power plant. Finding and reading through these conversations provided a window into the morale of the citizens.

WHAT WE LEARNED

We were able to find a lot of issues like utility outages and fatalities, only missing a few like broken gas lines and fires. This happened because we manually created the concept-detection rules in SAS Visual Analytics. Since we did not discover the fire and gas line messages in our initial exploration, we did not include them in the structuring stage of the data prep. A machine learning addition to filtering topics might have helped with uncovering the types of concepts in the data that deserved attention.

Sentiment analysis was unreliable due to the interference of some of the synthetic text generation. In spite of that SAS Visual Text Analytics was able to detect a few things like the increase in negative sentiment at 1AM when people were sending angry insults to each other and there was a slight drop in overall sentiment following the aftershock.

Our greatest shortcoming was our inability to analyze and report on the uncertainty of our results. The most we did was to capture whether messages provided direct or indirect reports. Being able to express uncertainty in the various aggregations and analytics in SAS Visual Analytics is an area that would benefit from additional capabilities.

CONCLUSION

We have demonstrated how SAS Visual Analytics can be used with other SAS tools and with open-source software to find solutions to an academic-sponsored analytics challenge. Several generally applicable tips can be applied from our work on these solutions to other analytics tasks.

We also hope to have shown the value in having data sets based on a ground truth available for testing analytical methods. You can use the VAST Benchmark Repository as a resource for finding data that may apply to your work.

REFERENCES

Alper, Basak, N. Riche, G. Ramos, and M. Czerwinski. 2011. "Design Study of LineSets, a Novel Set Visualization Technique" in *IEEE Transactions on Visualization and Computer Graphics*, vol. 17 , no. 12, pp. 2259-2267. https://ieeexplore.ieee.org/document/6064991.

IEEE VIS. 2020. "VIS: The premier forum for advances in visualization." Accessed February 16, 2020. http://www.ieeevis.org/.

Ramarajan, Rajiv. "DinoFun World." 2015. https://youtu.be/BIPscav27Js.

Ramarajan, Rajiv. "Earthquake in St. Himark" 2019. https://youtu.be/JHrUbod70-E.

Ramarajan, Rajiv. "Mystery at Lekagul Preserve." 2017. https://youtu.be/sOFfqlxVFi8.

SAS Institute Inc. 2020. "SAS Scripting Wrapper for Analytics Transfer (SWAT)." https://github.com/sassoftware/python-swat.

ACKNOWLEDGMENTS

Many at SAS have contributed their time and talents to the challenge entries over the years, including: Falko Schulz, Nascif Abousalh-Neto, Rajiv Ramarajan, Jon Nemargut, Biljana Belamaric-Wilsey, Shaun Kurian, Jesse Ollie, Paul Vezzetti, Matthew Horn, Rachael Nisbet, and Bill Pecic.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Riley Benson
SAS Institute Inc.
Riley.Benson@sas.com

Lisa Everdyke
SAS Institute Inc.
Lisa.Everdyke@sas.com

Karl Prewo
SAS Institute Inc.
Karl.Prewo@sas.com