Interactive data visualization on the web using R

Carson Sievert

Abstract

Some summary

1 Introduction

Cook, Buja, and Swayne [5] proposed a taxonomy of interactive data visualization based on three fundamental data analysis tasks: finding Gestalt, posing queries, and making comparisons. The top-level of the taxonomy comes in two parts: *rendering*, or what to show on a plot; and *manipulation*, or what to do with plots. Under the manipulation branch, they describe three branches of manipulation: focusing individual views, arranging many views, and linking multiple views. Of course, each of the three manipulation branches include a set of techniques for accomplishing a specific task (e.g., within focusing views: controlling aspect ratio, zoom, pan, etc), and they provide a series of examples demonstrating techniques using the XGobi software toolkit [18]. This paper applies similar interactive techniques to analyze data from three different sources using interactive web graphics.

Traditionally, interactive graphics software toolkits are available as desktop applications, but more recently, toolkits have used a web-based approach. In addition to being easier to share and embed within documents, a web-based approach opens up more potential for linking views between different interactive graphics toolkits. This ability grants a tremendous amount of power to the analyst since they may combine the strengths of several systems at once. However, unfortunately, there has been a surprising lack of work done on enabling graphical queries between multiple views (like the work done by Buja et al. [2]; **MANET**; Cook, Buja, and Swayne [5]; Cook and Swayne [6], but in a web-based environment).

For a number of years, R users have been able to link arbitrary views via the **shiny** package – a reactive programming framework for authoring web applications entirely within R [3]. Although **shiny** is a powerful tool for prototyping, it may introduce unnecessary computational complexity, and lacks semantics for performing graphical queries on web-based graphics. As a result, when linking views in a **shiny** app, one typically has to resort to a naive updating rule – when a query is made, the entire image/graph has to be redrawn. By adding semantics for graphical queries, it allows the underlying graphing libraries to be more intelligent about updating rules which generally leads to a more responsive graphical query.

The R package **plotly** is one such project that has semantics for linking views with and without **shiny** [17]; [15]. All of the examples in the exploring pedestrian counts section are available as standalone HTML files (i.e., without **shiny**) and were created entirely within R

via **plotly** and **leaflet** (a package for creating interactive web-based maps) [4]. There are, of course, limitations to the types of links one may create without **shiny**, and some of the examples in tracking disease outbreak and exploring Australian election data embed **plotly** graphs within **shiny** to dynamically perform customized R computations in response to user events.

2 Case Studies

2.1 Exploring pedestrian counts

The first example uses pedestrian counts from around the city published on the City of Melbourne's open data platform [11]. The City currently maintains at least 43 sensors (spread across the central business district), which record the number of pedestrians that walk by every hour. The analysis presented here uses counts starting in 2013 when all 42 of these sensors began recording counts, all the way through July of 2016. This code for obtaining and pre-processing this data, as well as the (cleaned-up) data is made available in the R package **pedestrians** [14]. The main dataset of interest is named `pedestrians` and contains nearly 1 million counts, but over 400,000 counts are missing:

```
data(pedestrians, package = "pedestrians")
summary(is.na(pedestrians$Counts))
```

```
##    Mode   FALSE    TRUE    NA's
## logical  942624  407189       0
```

2.1.1 Exploring missingness

Trying to visualize time series of this magnitude in its raw form simply is not useful, but we can certainly extract features and use them to guide our analysis. Figure 1 shows the number of missing values broken down by sensor. Southbank has the most missing values by a significant amount and the hand-full of stations with the fewest missing values have nearly the same number of missing values. One thing that Figure 1 can not tell us is *where* these missing values actually occur. To investigate this question, it is helpful to link this information to the corresponding time series.

Again, visualizing the entire time series all at once is not realistic, but we can still gain an understanding of the relationship between missingness and time via down-sampling techniques. Figure 2 displays an interactive version of Figure 1 linked to a down-sampled (stratified within sensor location) time series. Clicking on a particular bar reveals the sampled time series for that sensor location. The top one-third of all sensors are relatively new sensors, the middle third generally encounter long periods of down-time, while the bottom third seem to have very little to no pattern in their missingness.

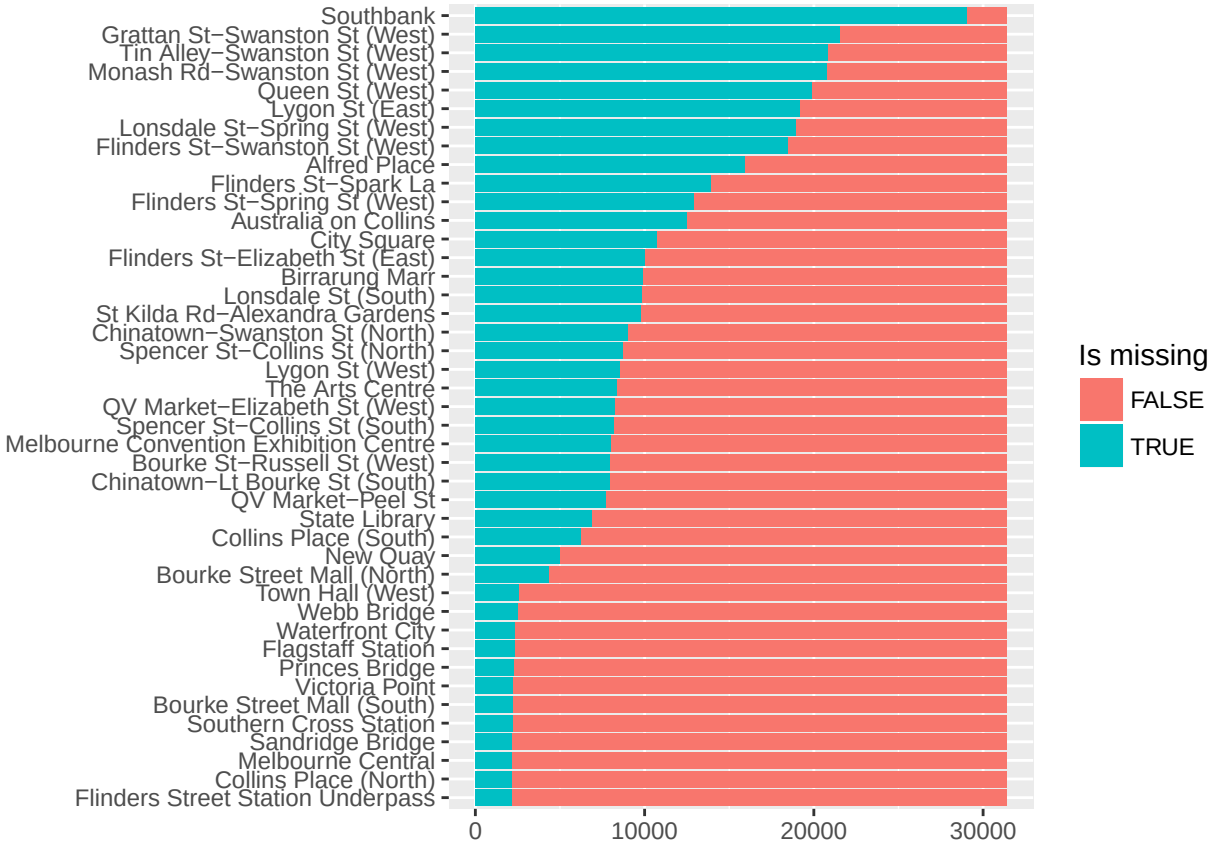

Figure 1: Missing values by station.

Figure 2: An interactive bar chart of the number of missing counts by station linked to a sampled time series of counts. See here for the corresponding video and here for the interactive figure.

2.1.2 Exploring trend and seasonality

A time series Y_t can be thought of as a linear combination of at least three components:

$$Y_t = T_t + S_t + I_t, \quad t \in \{1, \ldots, T\}$$

where T_t is the trend, S_t is the seasonality, and I_t is the "irregular" component (i.e., remainder). For the sensor data, we could imagine having multiple types of seasonality (e.g., hour, day, month, year), but as Figure 2 showed, year doesn't seem to have much effect, and as we will see later, hour of day has a significant effect (which is sensible for most traffic data), so we focus on hour of day as a seasonal component. Estimating these components has important applications in time series modeling (e.g., seasonal adjustments), but we could also leverage these estimates to produce further time series "features" to guide our graphical analysis.

There are many to go about modeling and estimating these time series components. Partly due to its widespread availability in R, the `stl()` function, which is based on LOESS smoothing, is a popular and reasonable approach [**stl**]; [12]. Both the **anomalous** and **tscognostics** R packages use estimates from `stl()`[1] to measure the strength of trend (as $\hat{Var}(T_t)$) and strength of seasonality (as $\hat{Var}(S_t)$) [8]; [19]. From these estimates, they produce other informative summary statistics, such as the seasonal peak ($Max_t(\hat{S}_t)$), trough ($Min_t(\hat{S}_t)$), spike ($Var[(Y_t - \bar{Y})^2]$); as well as trend linearity ($\hat{\beta}_1$) and curvature ($\hat{\beta}_2$) (coefficients from a 2nd degree polynomial fit to the estimated trend $\mu = \beta_0 + \beta_1 \hat{T}_t + \beta_2 \hat{T}_t^2$).

Projecting each time series into the 6 dimensional space spanned by these "STL features" allows us to graphically examine the sensor activity in a reasonable number of interpretable dimensions. Touring is a graphical technique for viewing such a feature space through animation. Similar to the rendering and perception of 3D objects in computer graphics, a tour smoothly interpolates through 2D projections of numeric variables – allowing the viewer to perceive the overall structure and identify clusters and/or outliers [6]. Figure 3 shows a couple frames taken from a tour of random 2D projections – also known as a grand tour [1]. The first frame (the top row) displays a projection with large weight toward linearity, trough, curvature, and season. From this frame, it is clear that one sensor (Tin Alley, highlighted in red) is unusual – especially along the linearity/curvature/trough dimensions. The second frame (the bottom row) displays a projection with large weight towards trend. Along this dimension, another unusual sensor appears (Bourke St).

In addition to highlighting observations by painting them directly on the tour, it can also be useful to link a tour to other views of the data, such as a parallel coordinates plot. In a parallel coordinates plot, each observation is represented by a line, and each line intersects numerous parallel axes (one for each measurement variable) [9]; [20]. Figure 4 links the grand tour from Figure 3 to a parallel coordinates plot of the same data, which provides another way of performing graphical queries (via individual measurements rather than linear combinations of measurements) [**brushing-pcp**]. In a later section, we leverage this "linked

[1] If no seasonal component exists, a Generalized Additive Models is used to estimate the trend component [23].

Figure 3: Two frames from a grand tour of measures generated from seasonal, trend, and irregular time-series components. The first frame (the top row) displays the state of the tour roughly 16 seconds into the animation while the second frame (the bottom row) is at roughly 60 seconds. A given frame displays both a 2D projection (on the left) and the linear combination of variables used for the projection (on the right). In both frames, the Tin Alley-Swanson St (West) sensor is highlighted in red – a useful technique for tracking interesting or unusual point(s) throughout a tour.

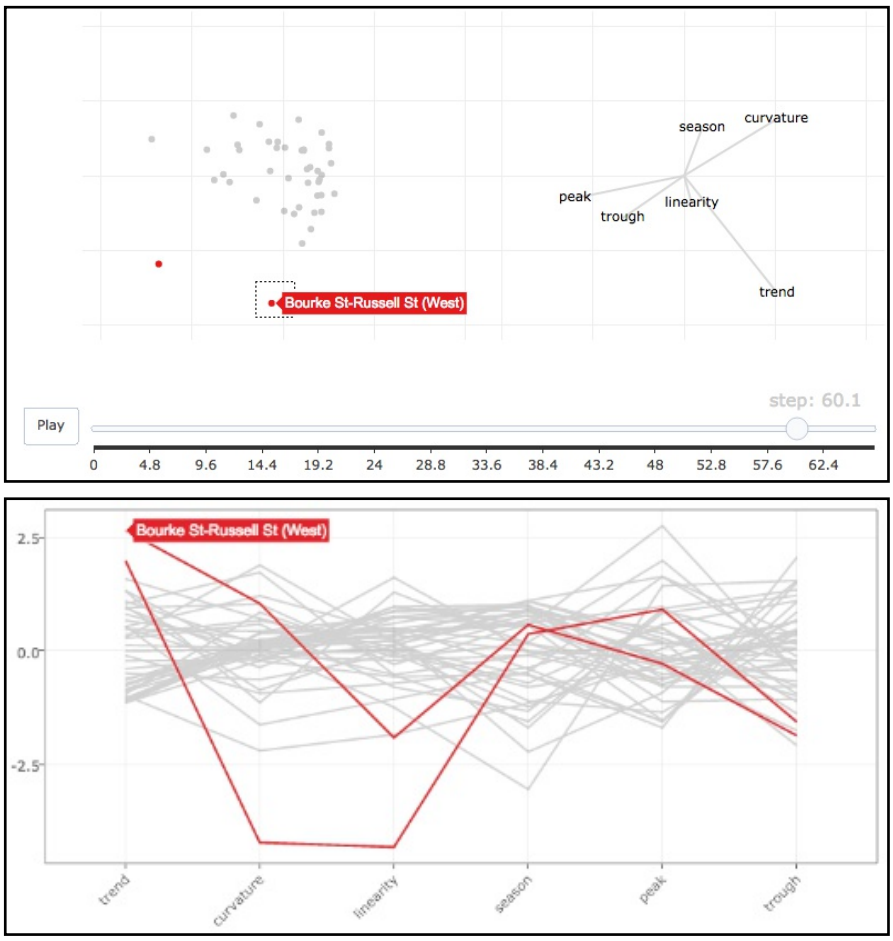

Figure 4: Identifying and comparing unusual sensors (Tin Alley and Swanson St) using linked highlighting between a grand tour and a parallel coordinates plot. The second frame of Figure 3 helped to point out Bourke St as a somewhat unusual sensor with respect to trend. Highlighting that point and linking it to a parallel coordinate plot makes it easier to compare trend across sensors and compare the other measures among sensors of interest.

highlighting" technique to identify clusters, but for now, we focus on highlighting unusual sensors.

The appearance of any parallel coordinates plot is effected by at least two choices – the ordering of the axes and the scale used to align the axes. The ordering of axes effects which relationships we end up seeing – a d-dimensional dataset has $d^2 - \sum_{i=1}^{d} i$ relationships, but parallel coordinates can only represent $d - 1$ relationships at a time. In this case, we have interpretable "groups" of variables (trend and seasonality), so Figure 4 uses an ordering to preserve the grouping. Furthermore, since most of the measurements are roughly normally distributed, Figure 4 centers and scales each variable to have mean 0 and standard deviation 1. As a result, it is really obvious that Tin Alley is really unusual with respect to curvature and linearity, but Bourke St and Tin Alley are only slightly unusual with respect to trend.

Now that we have a couple sensors of interest, it would help to link to other views that reveal more details about their sensor activity. Figure 5 adds two more linked views to Figure 4, including the inter-quartile range (IQR) of counts per hour, and a sample of the raw counts. The former display is useful for gaining an understanding of the magnitude and variation in sensor activity (IQR), while the latter is useful for discover outliers or unusual patterns.[2] Highlighting the different sensors with different colors in each view using a persistent brush fosters comparison and helps viewers track which graphical markers belong to which sensor. As a result, it becomes apparent that Tin Alley (in red) experiences relatively low traffic compared to Bourke St (in blue), and overall traffic (black).

Hundreds of comparisons and a fair amount of insight could be extracted from the interactive graphic in Figure 5, but focusing just on the STL-based features is somewhat limiting. There are certainly other features that capture aspects of the time series that these features have missed. In theory, the mathematics and the visualization techniques behind Figure 5 can be extended to any number of dimensions. In practice, technology and time typically limits us to tens to hundreds of dimensions. The next section incorporates more time-series features and also links this information to a geographic map so we can investigate the relationship between geographic location and certain features.

2.1.3 Exploring many features

Figure 6 is an extension of Figure 5 to incorporate 10 other time series features, as well as a map of Melbourne. In this much larger feature space, Tin Alley (in red) is still an unusual sensor, but not quite as unusual as Waterfront City (in blue). Also, rather interestingly, both of these sensors are located on the outskirts of the city, relative to the other sensors. It appears Waterfront City is so noticeably unusual due to its very large value of lumpiness (defined as the variance of block variances of size 24). Inspecting the unusually high raw counts for this station reveals some insight as to why that is case – the counts are relatively low year-round, but then spike dramatically on new years eve and on April 13th. A Google search reveals that Waterfront City is a popular place to watch fireworks. This is a nice example of how interactive graphics can help us discover and explain *why* unusual patterns occur.

In addition to discovering interesting details, we can use the same interactive display that generated Figure 6 to make meaningful comparisons between groups of sensors. Figure 7 uses a persistent linked brush to compare sensors with a high first order autocorrelation ($Corr(Y_t, Y_{t-1})$), in red, against sensors with low autocorrelation, in blue. A few interesting observations can be made from this selection state.

The most striking relationship with respect to autocorrelation in Figure 7 is in the geographic locations. Sensors with high autocorrelation (red) appear along Swanson St. – the heart of the central business district in Melbourne. These stations experience a fairly steady flow of traffic throughout the day since both tourists and people going to/from work use nearby trains/trams to get from place to place. On the other hand, sensors with a low

[2]The same sampling strategy used in Figure 2 is used to generate the dotplot (count versus hour of day).

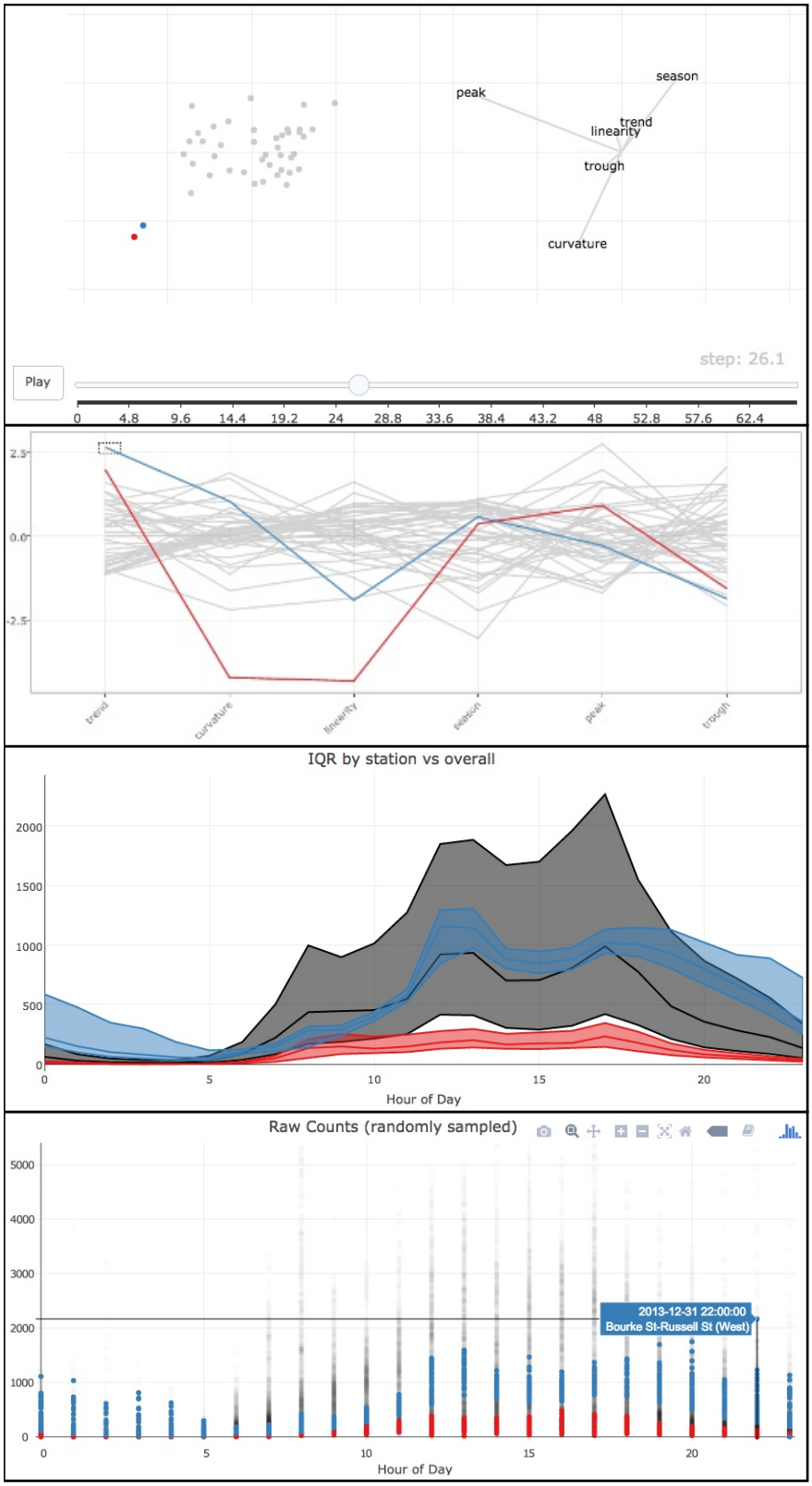

Figure 5: Linking views of seasonal trend decomposition summaries (first two rows) to the actual time series (last two rows). By linking raw counts and the hourly IQR, we can see that Tin Alley (in red) experiences relatively low traffic compared to Bourke St (in blue), and overall traffic (black). See here for the corresponding video and here for the interactive figure.

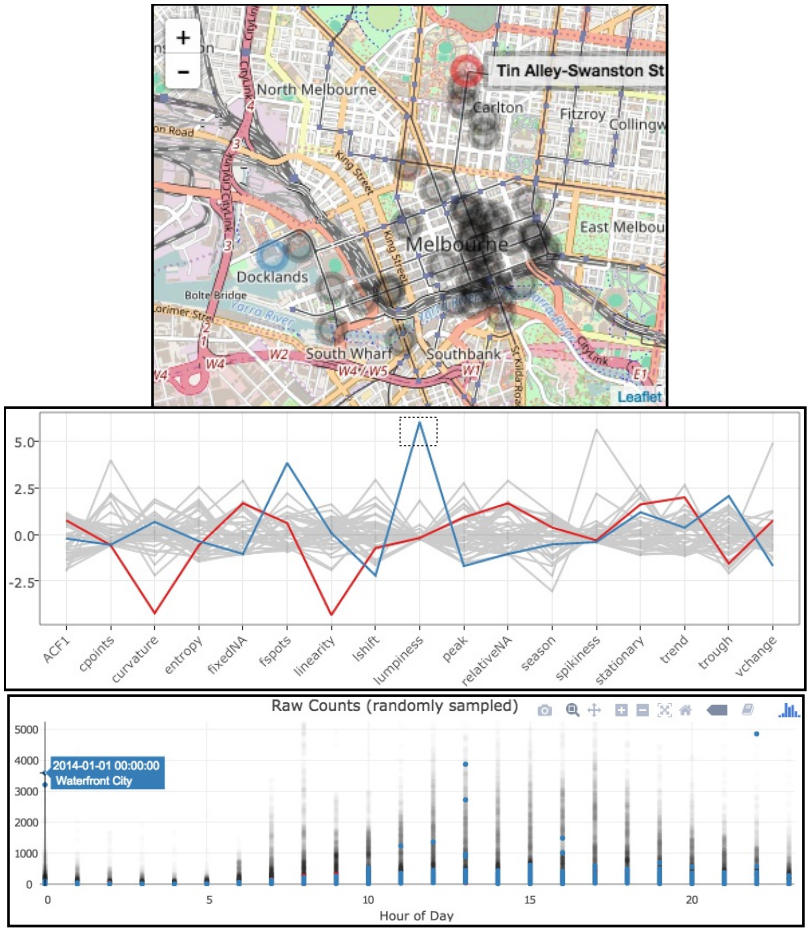

Figure 6: Seventeen time series features linked to a geographic map as well as raw counts. This static image was generated using a persistent brush to compare Tin Alley-Swanson St. (in red) to Waterfront City (in blue). In addition to being unusual in the feature space, these sensors are also on the outskirts of the city. The corresponding video and interactive figure (available here and here) also includes a grand tour and raw counts by day of the year.

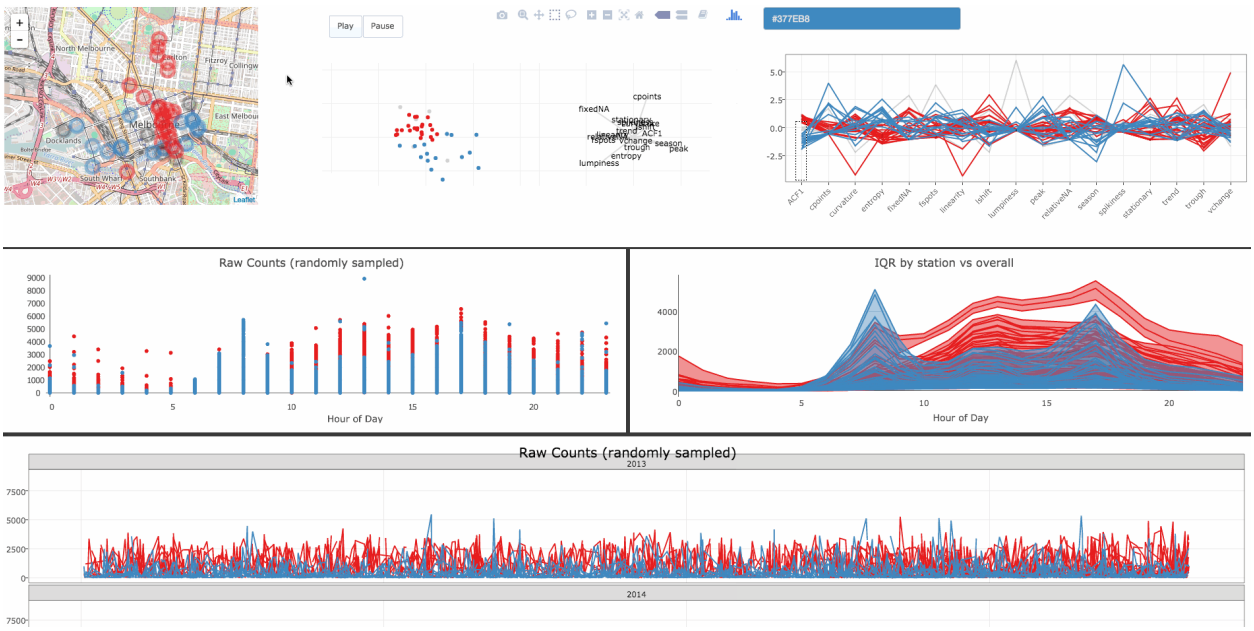

Figure 7: Sensors with high first order autocorrelation (in red) versus sensors with low autocorrelation (in blue). See here for the corresponding video and here for the interactive figure.

autocorrelation[3] see the bulk of their traffic at the start and end of the work day. It seems that this feature alone would provide a fairly good criteria for splitting these sensors into 2 groups, which we could verify and study further via hierarchical clustering.

Figure 8 links a dendrogram of a hierarchical cluster analysis (using the complete linkage method via the `hclust()` function in R) to other views of the data. A persistent brush selects all the sensors under a given node – effectively providing a tool to choose a number of clusters and visualize model results in the data space (in real-time). Splitting the dendrogram at the root node splits the sensors into 2 groups (red and green) which confirms prior suspicions – sensors on Swanson St (high autocorrelation) are most different from sensors on the outskirts of the city (low autocorrelation). Increasing the number of clusters to 3-4 splits off the unusual sensors that we identified in our previous observations (Waterfront City, Birrarung Marr, and Tin Alley-Swanson St).

This case study on pedestrian counts uses interactive graphic techniques for numerous data analysis tasks. In fact, Figure 6 alone provides at least one example of each task outlined by Cook, Buja, and Swayne [5]: finding Gestalt, posing queries, and making comparisons. Furthermore, as Figure 6 shows, and Wickham, Cook, and Hofmann [22] writes, these same interactive techniques can also be a helpful for understanding, inspecting, and diagnosing statistics models. The next case study on Zika virus infections demonstrates how interactive graphics can be useful for tracking disease outbreak and detecting data quality issues.

[3]It should be noted that the (raw) autocorrelation is positive for each station with a minimum of 0.66, median of 0.83, and max of 0.94.

Figure 8: Linking a dendrogram of hierarchical clustering results to multiple views of the raw data. See here for the corresponding video and here for the interactive figure.

2.2 Tracking disease outbreak

The next case study investigates Zika disease outbreaks across North, Central, and South America. The data was obtained from a publically available repository that curates data from numerous public reports across numerous countries and regions [13]. Of course, each country has a different reporting method, so reported cases can and do fall under many different categories. Thankfully, Rodriguez et al. [13] have done the tedious work of standardizing these codes so we can combine all of these reports into a single dataset. In some countries, reports are broken down to by different demographics, and include reports of similar diseases such as Flavi virus and GBS, but this analysis focuses specifically on suspected/confirmed Zika cases at the location level.

The R package **zikar** bundles suspected/confirmed Zika cases at the location level and provides specially designed tools for visualizing it [16]. All the graphics in this section were generated via the `explore()` function from **zikar**, which invokes a **shiny** app with linked interactive graphics [3].[4] Figure 9 displays the default view of the web application, which provides a concise overview of the reported counts. The map on the left-hand side shows the different reporting locations. The non-black markers represent multiple locations, and hovering over a marker reveals the entire region that the marker represents. From this, we can see that the bulk of reporting locations are in the northern part of South America and the southern part of Central America. The right hand side of Figure 9 shows the overall density of weekly cases (on a log scale) as well as the weekly median over time.

[4] A hosted version of this web application is avaliable here.

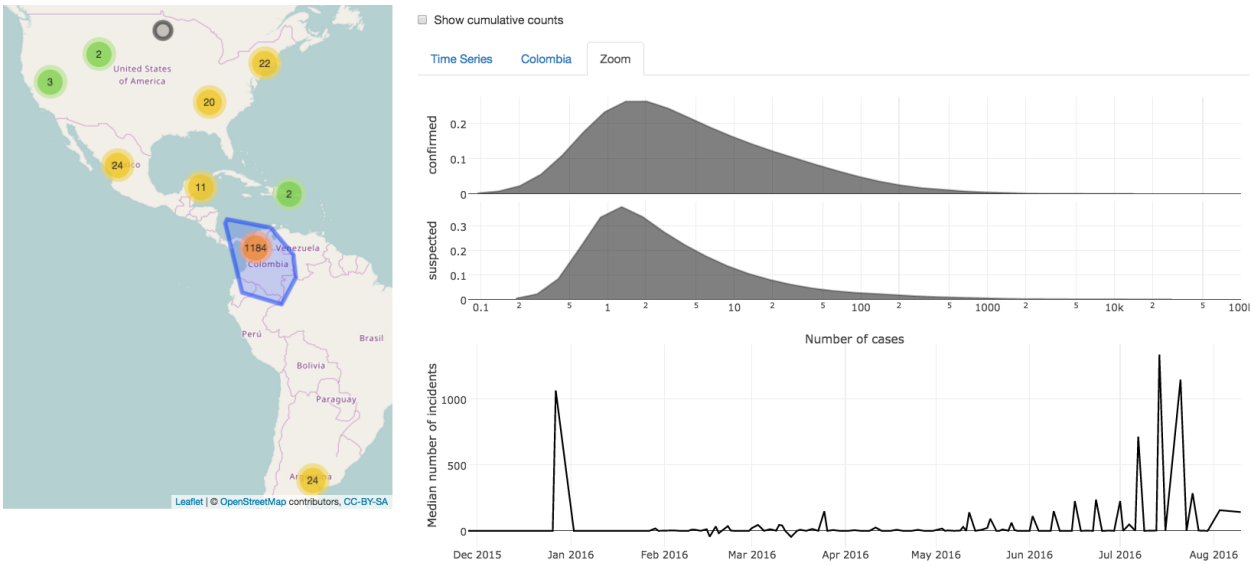

Figure 9: Multiple views of the Zika outbreak data. On the left-hand side is a map of the reporting locations. On the right is the overall density of suspected/confirmed cases reported per week (on a log scale), and the overall weekly median over time.

Zooming and panning to a particular region on the interactive map reveals more information conditioned on the bounding box of the map. Figure 10 displays information specific to the Dominican Republic. In the map itself, the "marker clusters" have updated for a more granular view of the number of locations reporting within the area. In the other views, statistics conditional upon this region (in red) are overlaid against overall statistics (in black) for fast and easy comparison(s). Figure 10 shows the density of suspected cases in the Dominican is much higher than the overall density, and the density for confirmed cases has a much larger amount of variation. Furthermore, the weekly median within this region is consistently higher from March 2016 to July 2016.

Figure 9 helps to point out an issue with the data – in some weeks, the median number of all reported cases is negative. Using the zooming and panning capabilities of Figure 9, one may quickly find a sub-region of the map that reflects the same overall issue, which helps to guide an investigation into why this issue exists. Figure 11 uses this functionality to find that both El Salvador and Nicaragua report negative counts, at different times of the year. Considering that these countries report a cumulative count of currently infected people, these negative non-cumulative counts indicate mis-diagnosis or death. Since deaths caused by Zika in adults are quite rare, and symptoms from the Zika virus are hard to differentiate from the more common Dengue virus (another disease spread via infected mosquitoes), mis-diagnosis seems to be the more likely reason for the negative counts [10].

As it turns out, Nicaragua and El Salvador are not the only countries that have reported a lower cumulative count from one week to the next. Figure 12 shows cumulative confirmed (in red) and suspected (in blue) counts by location within 9 different countries. Argentina is another country that has clearly encountered mis-diagnosis issues – there is a consistent dip in counts across all locations within the country on two particular dates in May 2016.

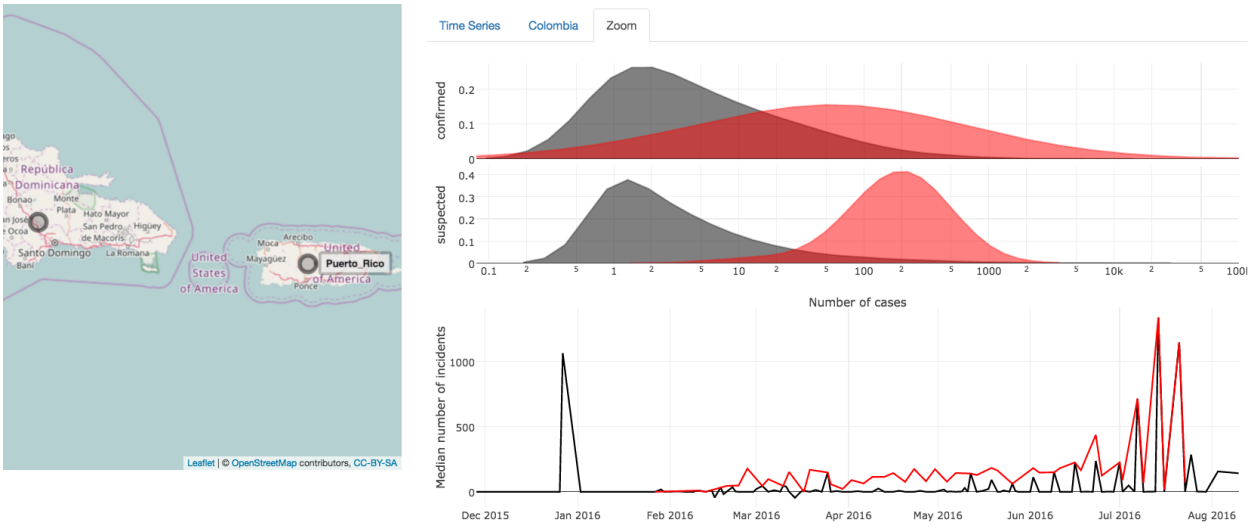

Figure 10: A comparison of the overall cases (in black) to the cases conditional on the map bounds (in red). Zooming and panning the interactive map dynamically updates the density estimates and median number of incidents.

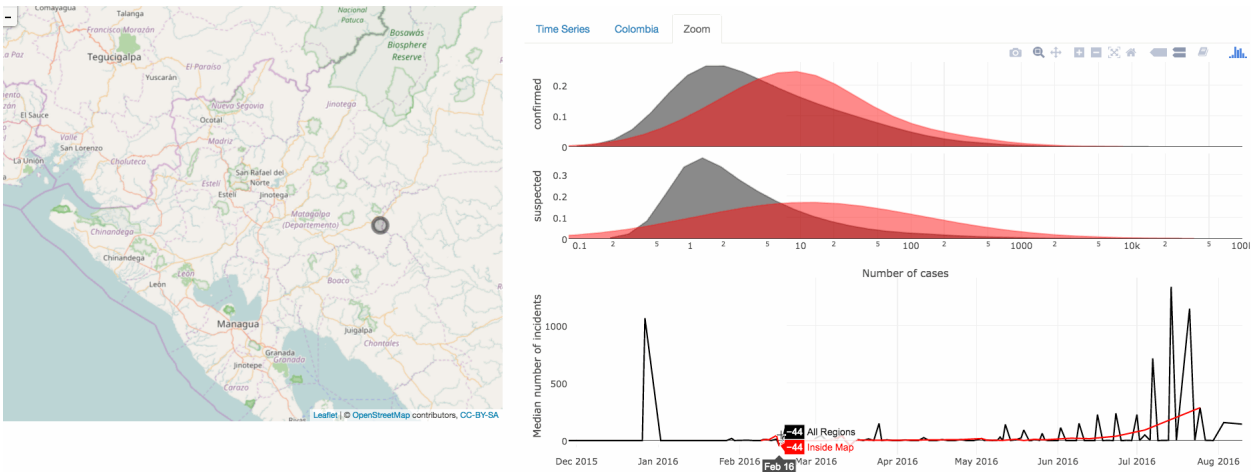

Figure 11: Zooming and panning to a region of the map that has a negative median of overall cases (Nicaragua). A video of the zooming and panning may be viewed here.

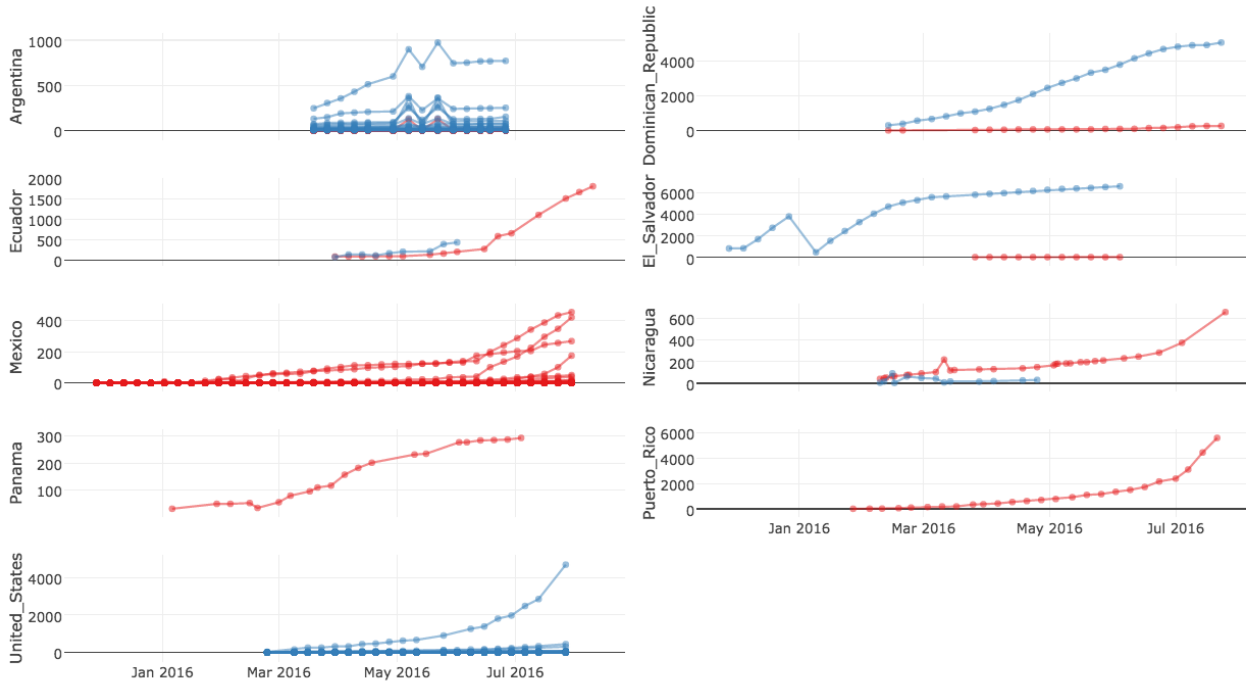

Figure 12: Cumulative confirmed (in red) and suspected (in blue) counts by location within 9 different countries.

Although Figure 12 covers almost all the countries in this data, it only covers ~5% of all reporting locations. Colombia alone accounts for ~95% of all locations found in this data and has some unique reporting issues of its own.

Figure 13 shows the cumulative number of confirmed (in red) and suspected (in blue) cases for every reporting location in Colombia. From the static version of Figure 13, it seems plausible that every location re-classified all confirmed cases to suspected around mid-March. By clicking on a particular line (shown in the video of Figure 13) to highlight the confirmed/suspected counts for a particular location, it becomes even more obvious that every Colombian location simply changed all their cases from confirmed to suspected.

This case study shows how interactive graphics can be useful to discover issues in data that should be addressed before any statistical modeling occurs. For this reason, they are particularly useful for analysts coming at the problem with a lack of domain expertise, and can provide insight helpful for downstream analysis. The next case study uses interactive graphics to explore Australian election data and provides a nice example of combining numerous data sources into a single dashboard of linked views.

2.3 Exploring Australian election data

The next case study takes a look at the relationship between demographics and voting behavior in the 2013 Australian general election. Demographic information was obtained

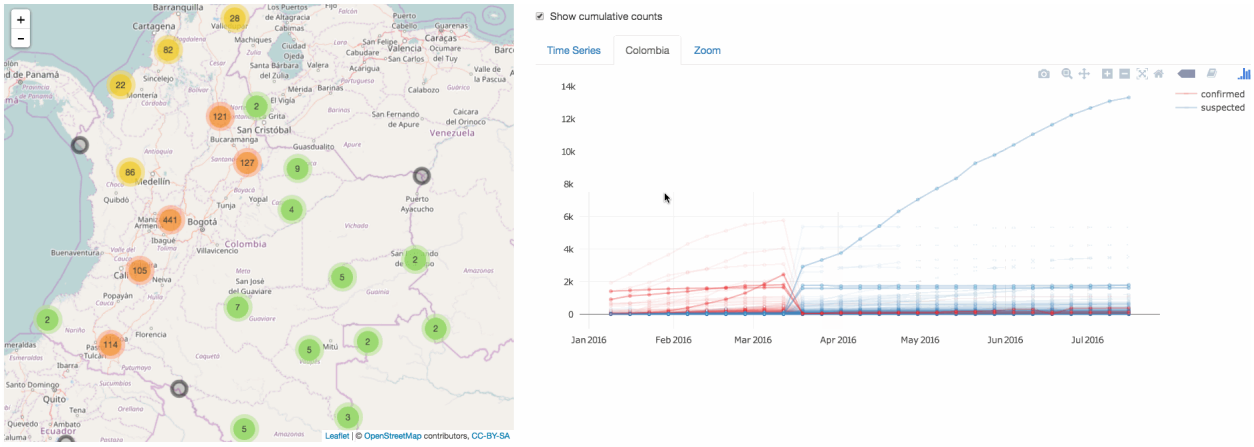

Figure 13: Highlighting cumulative confirmed (in red) and suspected (in blue) counts by location within Colombia to verify re-classifications from confirmed to suspected. A video of the interactive highlighting may be viewed here.

from the Australian Bureau of Statistics (ABS)[5], voting information was obtained from the Australian Electoral Commission (AEC)[6], and all the data as well as the interactive graphics presented here are available via the R package **eechidna** [7]. Thankfully, these data sources can be linked via electoral boundaries from the 2013 election (and the geo-spatial boundaries are also available[7]), making it possible to explore the relationship between voting behavior and demographics across electorates.

Figure 14 shows demographics of electorates that elected candidates (for the House of Representatives in the 2013 general election) from the Liberal Party in green, the Australian Labor Party in orange, and all other parties in black. As shown in this video, Figure 14 was generated via the `launchApp()` function from **eechidna**, which invokes an interactive visualization where electorates may be graphically queried according to voting outcomes, geography, and/or demographics. To foster comparisons, density estimates for all the 32 different demographics are displayed by default, but the number of demographics may also be restricted when invoking `launchApp()`.

The density estimates in the lower portion of Figure 14 suggest that Labor electorates tend to be younger (in particular, they have a higher percentage of 20-34 year olds, and lower percentage of 55-64), are more unemployed, have lower income, and are more populated. Most of these characteristics are not surprising given the ideologies of each party; however, it is surprising to see a relationship between the population within electorate and the elected party. In theory, electorates should be divided roughly equally with respect to population so that a single vote in one electorate counts as much as a vote in another electorate. However, in practice, there has to be some variation in population; and as it turns out, more populated electorates tend to vote Labor, implying that (on average) a vote towards the Labor party counts less than a vote for the Liberal party.

[5]Downloaded from https://www.censusdata.abs.gov.au/datapacks/
[6]Downloaded from http://www.aec.gov.au/elections/federal_elections/2013/downloads.htm
[7]http://www.aec.gov.au/Electorates/gis/gis_datadownload.htm

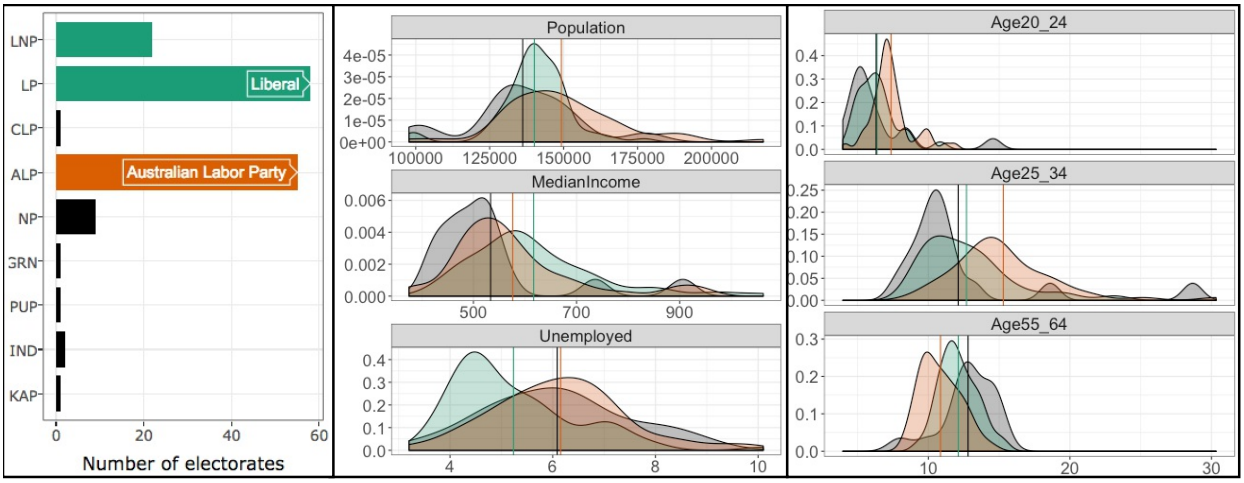

Figure 14: Electorate demographics among the Liberal Party (in green), the Australian Labor Party (in orange), and other parties (in black). The vertical lines represent the mean value within each group. The interactive application used to generate this image may be accessed here and a video of the interactive highlighting may be viewed here.

Figure 14 was created by painting the relevant bars in the upper-left hand panel of Figure 15. Note that, due to the similarities of the parties, both the Liberal National Party of Queensland (LNP) and the Liberal Party are grouped into a single liberal party (colored in green). The interaction with the bar chart not only populates relevant density estimates, as in Figure 14, but it also colors every graphical mark representing the corresponding electorates in the other views shown in Figure 15.

The line chart in the upper-right hand panel of Figure 15 shows the proportion of 1st preference votes for each party within a given electorate – showing an expected difference between ALP/LP/LNP voting as well as other interesting patterns (e.g. liberal party electorates tend to have a higher proportion of 1st preference votes going to the PUP party). The left-hand panel of Figure 15 displays the absolute difference in vote totals within electorates, making it easy to identify closely contested electorates (more on this later). The right-hand panel of Figure 15 displays a map of Australia with polygons outlining the electorate boundaries. Since the boundaries within highly populated areas (e.g., Melbourne, Sydney, and Brisbane) are so small, the location of points (on top of the map) were adjusted by a force layout algorithm in order to avoid too much overplotting.

Although interaction with the bar chart in Figure 15 helped to generate Figures 14 and 15, electorates may be queried via directly manipulation with any of these plots. For example, Figure 16 was generated by brushing the electorates that were determined by less than 10% of the vote total (via the plot with the absolute difference in vote totals).

Figure 16 highlights closely contested electorates (determined by less than 10% of the vote total) which provides insight into the demographics of voters that future political campaigns should target in order to maximize their campaigning efforts. Voters in these competitive areas tend to have a high proportion of 25-34 year olds, have more diverse religious backgrounds,

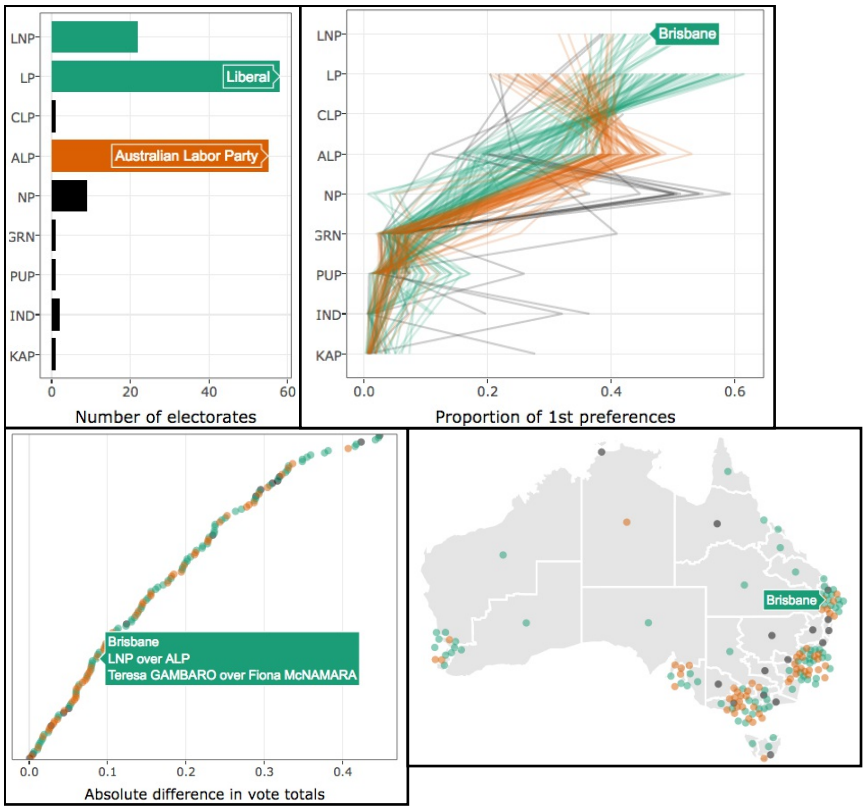

Figure 15: Comparing voting outcomes and geographic location among the Liberal Party (in green), the Australian Labor Party (in orange), and other parties (in black). The bar chart in the upper-left hand panel shows the number of electorates won by each party. The upper-right hand panel shows the proportion of 1st preference votes for each party for given electorate. The lower-left hand panel shows the absolute difference in vote totals for each electorate. The lower-right hand panel show the locations of electorates.

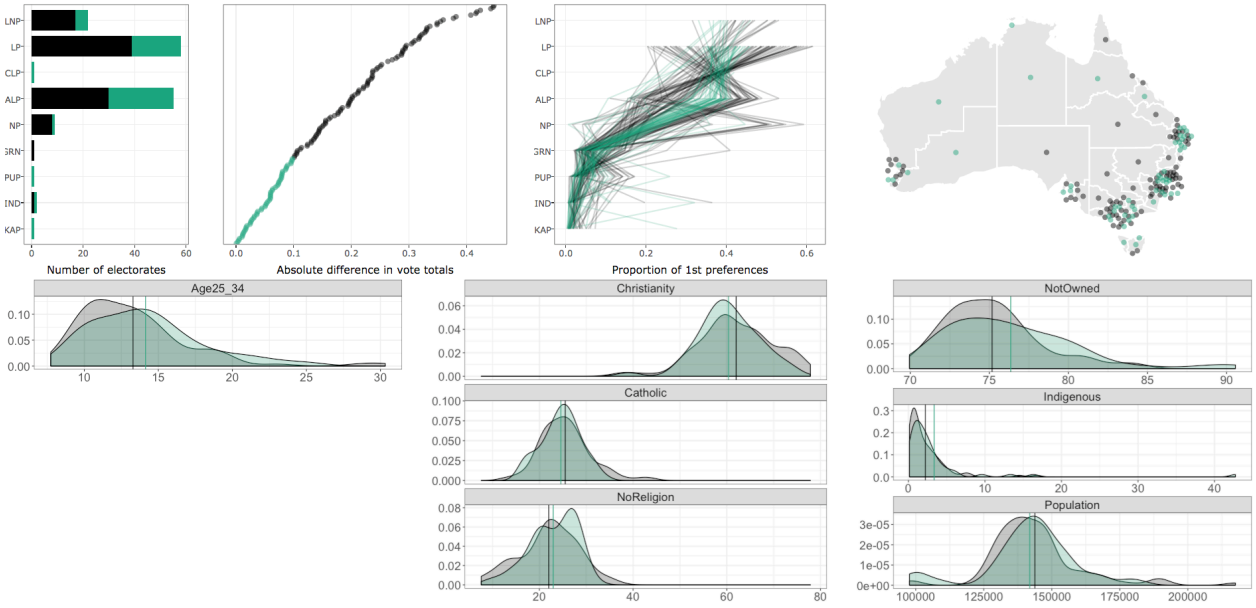

Figure 16: Electorates that were determined by less than 10 percent of the total vote. These electorates tend to have voters that are younger, less religious, are less likely to own property, and lean towards the Labor party.

are less likely to own property, and a higher percentage of the population are Indigenous. Unsurprisingly, these electorates lean towards the Labor party, but there are a few electorates within this group that have a surprising population (around the minimum of 100,000 people). Figure 17 paints electorates with a population around the minimum orange, which reveals a striking geographic relationship among these electorates – almost all of them are in Tasmania (with the exception of one in the Northern Territory). Furthermore, as 17 shows, all of these electorates (except for Denison), experienced a close election, so campaign efforts would be well spent in these electorates.

3 Conclusion

Interactive graphics, particularly multiple linked plots with support for direct manipulation, provide a powerful data analysis tool for posing queries and making comparisons. This paper gives three different case studies applying interactive web graphics to real-world data sets to extract insights and present the graphics themselves in an accessible and convenient format. Furthermore, these graphical techniques can be widely useful not only for the analyst conducting exploratory data analysis, but also for understanding and diagnosing statistical models, and presenting results to a wider audience.

Interactive web graphics are already widely used for presenting and communicating results of an analysis (where the visualization type is already known), but are less often used for exploring data – mostly due to a lack of tools for iteration within a larger statistical computing environment. The **plotly** package aims to address this lack of tools by enabling R users

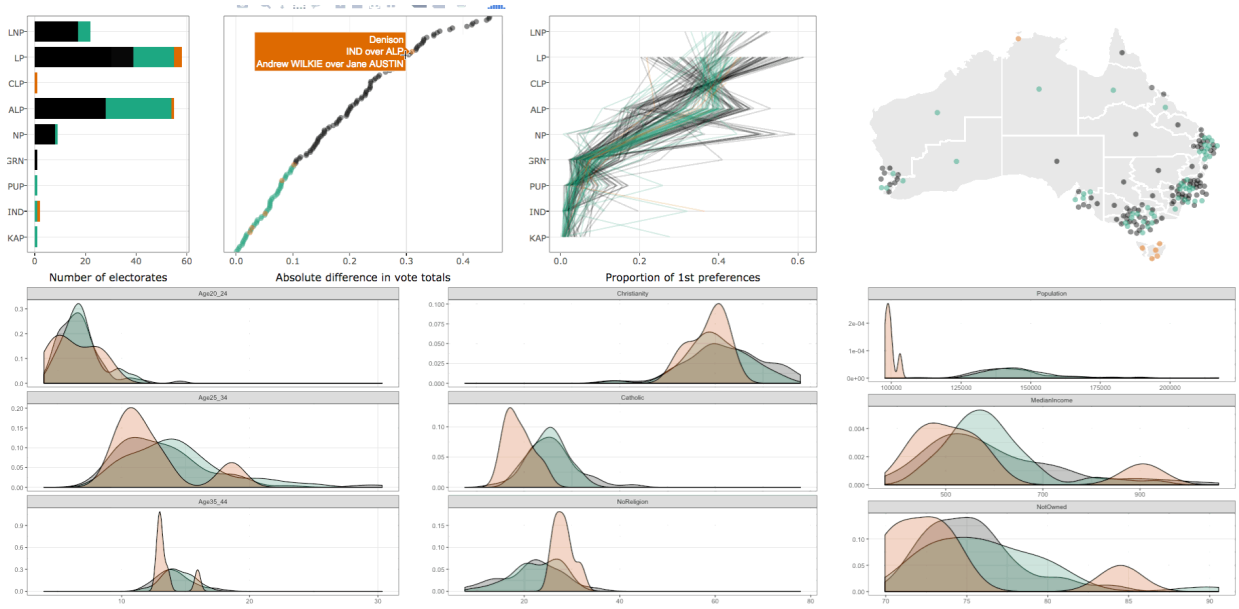

Figure 17: Electorates that experienced a close election as well as electorates with small populations (in orange).

to produce highly interactive and dynamic web graphics by leveraging already well-known and widely used interfaces for exploratory data analysis. In particular, the **plotly** package makes it easy to translate **ggplot2** graphics[8] to a web-based version, and enable interactive techniques such as highlighting and linked brushing [21].

All of the examples in the exploring pedestrian counts section were created with **plotly** and are available as standalone HTML files that can be easily deployed and shared with well-established web technologies. The sections tracking disease outbreak and exploring Australian election data link **plotly** graphs using the **shiny** package for authoring web applications to enable linked interactions that compute statistical summaries based on graphical queries defined by users. Furthermore, all of the examples across all of these sections were created purely within R, and requires no knowledge of web technologies such as HTML/JavaScript/CSS from the user. As a result, projects such as **plotly** and **shiny** help analysts focus on their primary task (data analysis) rather than the implementation details typically involved when creating interactive web graphics.

References

[1] Daniel Asimov. "The Grand Tour: A Tool for Viewing Multidimensional Data". In: *SIAM J. Sci. Stat. Comput.* 6.1 (Jan. 1985), pp. 128–143. ISSN: 0196-5204. DOI: 10.1137/0906011. URL: http://dx.doi.org/10.1137/0906011.

[8] Some recent estimates suggest that 100,000s of people use **ggplot2**, a graphing interface which is especially well suited for creating exploratory graphics.

[2] Andreas Buja et al. “Interactive data visualization using focusing and linking”. In: *IEEE Proceedings of Visualization* (Feb. 1991), pp. 1–8.

[3] Winston Chang et al. *shiny: Web Application Framework for R*. 2015. URL: http://CRAN.R-project.org/package=shiny.

[4] Joe Cheng and Yihui Xie. *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. 2015. URL: http://rstudio.github.io/leaflet/.

[5] Dianne Cook, Andreas Buja, and Deborah F Swayne. “Interactive High-Dimensional Data Visualization”. In: *Journal of Computational and Graphical Statistics* (Dec. 1996), pp. 1–23.

[6] Dianne Cook and Deborah F. Swayne. *Interactive and dynamic graphics for data analysis : with R and GGobi*. Use R ! New York: Springer, 2007. ISBN: 978-0-387-71761-6. URL: http://www.ggobi.org/book/.

[7] Di Cook et al. *eechidna: Exploring Election and Census Highly Informative Data Nationally for Australia*. 2016. URL: https://github.com/ropenscilabs/eechidna.

[8] Rob J Hyndman, Earo Wang, and Nikolay Laptev. *anomalous: Unusual Time Series Detection*. 2016.

[9] Alfred Inselberg. “The Plane with Parallel Coordinates”. In: *Visual Computer* 1 (4 1985), pp. 69–91. DOI: 10.1007/BF01898350.

[10] Donald G. McNeil and Daniel Victor. *First U.S. Death Tied to Zika Is Reported in Puerto Rico*. 2016. URL: http://www.nytimes.com/2016/04/30/health/zika-virus-first-death-in-us-puerto-rico.html.

[11] City of Melbourne. *City of Melbourne's open data platform*. Accessed: 2016-10-15. 2016. URL: https://data.melbourne.vic.gov.au/.

[12] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: http://www.R-project.org/.

[13] Dania M. Rodriguez et al. *zika: September 7, 2016*. Sept. 2016. DOI: 10.5281/zenodo.61753. URL: https://doi.org/10.5281/zenodo.61753.

[14] Carson Sievert. *pedestrians: Tools for Exploring Melbourne's Pedestrian Data*. 2016. URL: https://github.com/cpsievert/pedestrians.

[15] Carson Sievert. *plotly for R*. 2016. URL: https://cpsievert.github.io/plotly_book.

[16] Carson Sievert. *zikar: Tools for exploring publicly available Zika data*. 2016.

[17] Carson Sievert et al. *plotly: Create Interactive Web Graphics via 'plotly.js'*. 2016. URL: https://CRAN.R-project.org/package=plotly.

[18] Deborah F Swayne, Dianne Cook, and Andreas Buja. “XGobi: Interactive Dynamic Data Visualization in the X Window System”. In: *Journal of Computational and Graphical Statistics* 7.1 (Mar. 1998), pp. 113–130.

[19] Earo Wang. *tscognostics: Time Series Cognostics*. 2016. URL: https://github.com/earowang/tscognostics.

[20] E.J. Wegman. “Hyperdimensional data analysis using parallel coordinates”. In: (85 1990), pp. 664–675.

[21] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009. ISBN: 978-0-387-98140-6. URL: http://ggplot2.org.

[22] Hadley Wickham, Dianne Cook, and Heike Hofmann. “Visualizing statistical models: Removing the blindfold”. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8.4 (2015), pp. 203–225.

[23] S.N. Wood. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2006.