ggparliament:

Simple Parliament Plots in R

Zoe Meers[*]

Draft as of October 02, 2018

[*]United States Studies Center, University of Sydney

Introduction

In this article, I introduce a new data visualization tool in `R` for graphing legislatures. This `R` package, `ggparliament`, allows for easy visualization of legislatures, including district-level descriptive data, election results; in addition, it can be used to analyse the structure and design of political institutions (Hickman, Meers, and Leeper 2017). Graphing legislatures and parliaments provide information that may be easy to miss in tabular form but nearly impossible to miss visually. `ggparliament` is a `ggplot2` (Wickham et al. 2018) extension that takes aggregate election returns, expands the rows of the data frame to the number of individual seats in the legislature, and plots the parliament according to a specified parliament layout. Such graphs are known as parliament plots.

Visualizing legislatures can be challenging. Users must find the x and y coordinates for each individual seat, and plot the points on a coordinate plane. While this task is not impossible, parliament plots are cubersome to create in most statistical software packages. Until relatively recently, parliament plots were plotted using various JavaScript frameworks. Programs such as 'd3-parliament' (Brossard 2016), 'Parliament SVG' (Tens 2017a) and 'Westminster SVG' (Tens 2017b) give the end-user greater freedom to draw directly on the Document Object Model[1] using Scalable Vector Graphics (SVG). Click-and-point options also exist for non-programmers, including 'Parliament Diagram' (Richfield 2018) which is primarily used for writing Wikipedia articles. These tools, while excellent, are not particularly appropriate or useful choices for political scientists.[2] `ggparliament` seamlessly integrates into the quantitative political scientist's toolkit and allows for easy composition of legislatures using `ggplot2` syntax in the `tidyverse` (Wickham 2017), a set of packages driven by an opinionated data science philosophy in `R`.

To highlight the utility of `ggparliament` in political science, I first situate this `R` package

[1] In common parlance, an empty webpage.

[2] 'D3.js' and other 'JavaScript' frameworks are more often used by front end web developers, not academics. Likewise, Wikipedia charts do not fit well with the workflow of the modal political scientist.

in the context of other data visualization tools used in the discipline. I then show how to visualize parliament plots. Finally, I expand on several cases in which `ggparliament` may improve comprehension of descriptive data in political science.

Literature review

Data visualization in political science is not new. Several scholars have encouraged use of graphs vis-à-vis tables in communicating scientific research (King, Tomz, and Wittenberg 2000; Tomz, Whittenberg, and King 2003). Kastellec and Leoni follow up on this effort several years later, proposing ways to graph statistical results in political science that would otherwise be presented as a table (Kastellec and Leoni 2007). Others have written fairly extensively on graphics and presenting information in political science and in statistics (Gelman 2011). Visualizing data has increased in popularity in the social sciences as computational tools become more accessible. Political scientists, like Edward Tufte, are prolific data visualizers, having transformed how political scientists visualize information (Tufte and Graves-Morris 1983).

Yet, while statistical research computing in political science has increased in use since the 1970s (see **???**, (**???**), (**???**), (**???**), (**???**), (**???**), (**???**) for examples of statistical computing languages or developments made for and by social scientists.), data visualization tools still remain the domain expertise of people in computational statistics and/or Human Computer Interaction. Specialist data visualization tools in political science are rare. However, there are a few noteable data visualization achievements that originate from political science, including small multiples (Tufte, Goeler, and Benson 1990), marginal effects plots (Brambor, Clark, and Golder 2006; Hainmueller, Mummolo, and Xu 2018; Pepinsky 2018) and the separation plot (Greenhill, Ward, and Sacks 2011).

Data visualization in political science

An analysis of journal articles from the *American Journal of Political Science*, *American Political Science Review*, *Journal of Politics*, and *Political Analysis* published in 2018 show that the inclusion of figures and graphs are common-place in political science.

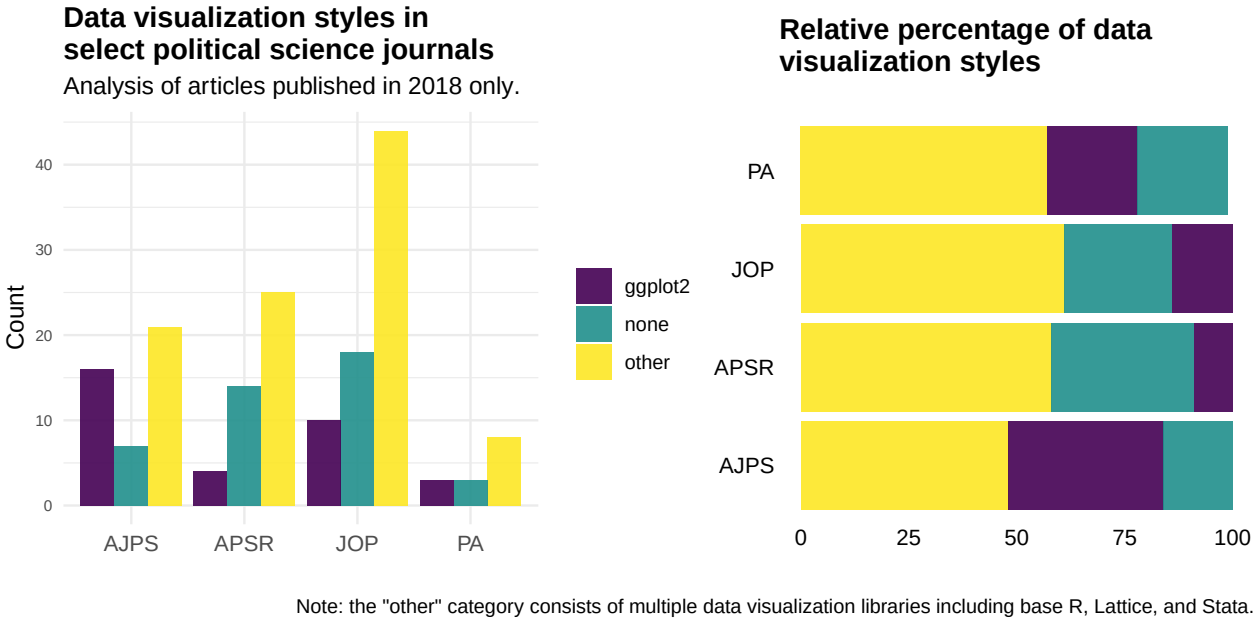

Figure 1: Political science and data visualization

While nearly half of the articles in the four journals do not contain graphs, this is an unsurprising result given that there are subfields or areas of research within subdisciplines in political science in which there is little need to graph results. Of the articles that do contain data visualizations, a significant proportion are made in `ggplot2`. In the *American Journal of Political Science*, 36% of articles published in 2018 include `ggplot2` graphics.

While `ggplot2` is just one data visualization tool out of many, this analysis provides evidence of the popularity of the `tidyverse` packages in political science. The use of data visualization as a means to communicate results is strongly encouraged in political science. The relative accessibility of tools like `ggplot2` enable social scientists to easily communicate their results graphically.

Simple Parliament Plots

`ggparliament` is an extension to the `tidyverse`, a subset of packages in `R` which build off a common grammar of graphics and data analysis, enabling users to quickly understand their data through chained functions (Wilkinson 2006, Wickham (2010)). The output can be visualized with ease in `ggplot2`. `ggparliament` is the first data visualization tool in a statistical framework for displaying election returns, party structures, and legislative data in a parliament plot. `ggparliament` offers five legislative chamber layouts for plotting parliament plots.

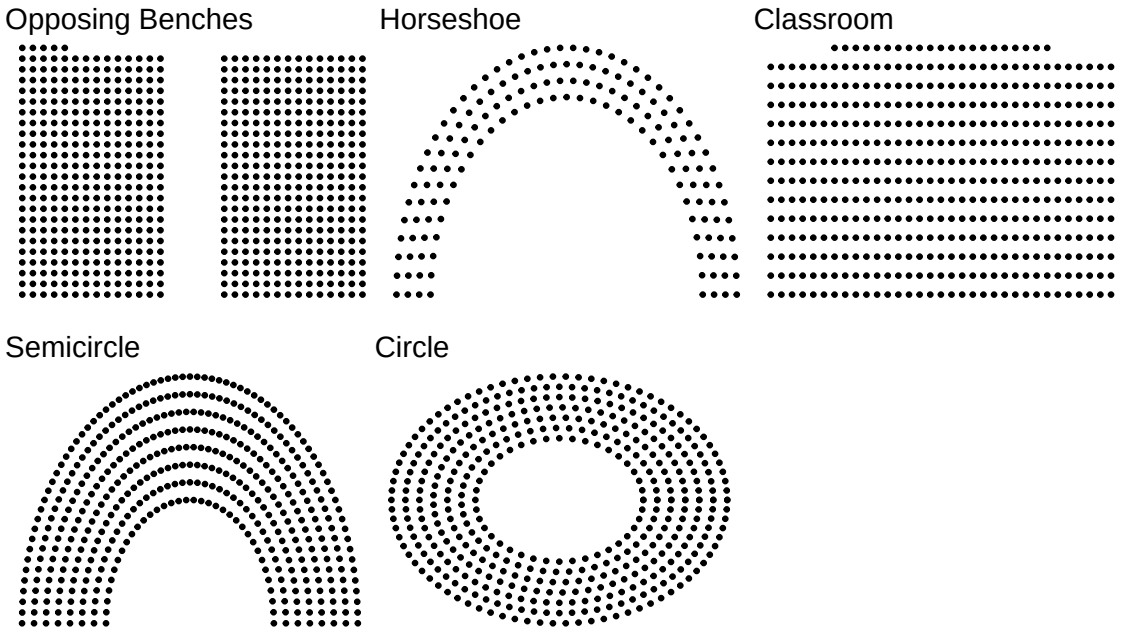

Figure 2: Basic parliament plots

Simple descriptive statistics such as election results are often best represented in tables. However, I argue that parliament plots do have some utility. Graphs provide a accurate summmary of the distribution of seats in a legislature. They can be used to visually group parties together when the number of parties exceeds a two-party system or can make explicitly clear the seat swing between two elections. As `ggparliament` provides several parliament layouts, one can create a cross-country comparative analyis. Moreover, parliament plots allow readers to visualize the layout of the legislature. Moving away from the primary intended

purpose of plotting election results, `ggparliament` can display information about legislative districts, roll call data, or information about legislators themselves in a compact format.

The Concept

Constructing a parliament plot starts with aggregate political data. To illustrate how to use `ggparliament`, I take the 115th Congress as an example. As of August 2018, there are 236 Republicans, 193 Democrats and 6 vacant seats in the House of Representatives which are equally divided by the two political parties. The data frame is structured as follows:

Table 1: The United States House of Representatives

party	seats	color
Vacancies	3	gray
Republicans	236	red
Democrats	193	blue
Vacancies	3	gray

Next, we input the data frame into `ggparliament::parliament_data()`, defining the parameters of interest:

- the type of parliament layout (hemicycle, circle, horseshoe, classroom, or oppposing benches)
- the number of rows in the legislative chamber
- the numerical 'seats' variable which tells `R` the total number of seats per party
- the name of the original aggregate data frame.[3]

[3]If you are plotting an opposing benches style parliament for the House of Commons, you must add an

Table 2: `ggparliament::parliament_data()` output

	party	seats	color	x	y	row	theta
1	Vacancies	3	gray	-2.000000	0	8	3.141593
1.1	Vacancies	3	gray	-1.857143	0	7	3.141593
1.2	Vacancies	3	gray	-1.714286	0	6	3.141593
2	Republicans	236	red	-1.571429	0	5	3.141593
2.1	Republicans	236	red	-1.428571	0	4	3.141593
2.2	Republicans	236	red	-1.285714	0	3	3.141593

The function `parliament_data()` is primarily dependent on two arguments: the number of rows in the legislative chamber and specified layout. The data frame expands to individual rows for any given legislator with x and y coordinates assigned to their seat. The user can then bind other information to the seat by appending new columns to the data. Plotting the output in `ggparliament` leads to the following graph (see Figure 3).

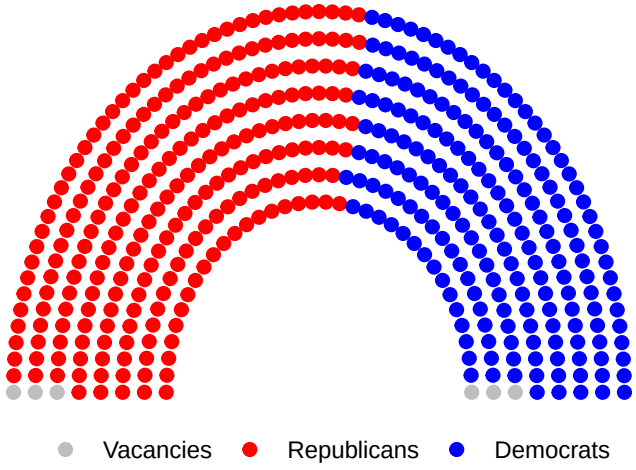

Figure 3: United States House of Representatives parliament plot

additional grouping variable to split the two benches.

Additional Information

The primary purpose of `ggparliament` is to plot election returns. However, there exists a lot of potential for plotting other types of descriptive political data. We can append additional information to parliament plots, including faceting legislatures over time or pieces of legislation, plotting majority threshold lines, or grouping coalition governments together.

The `tidyverse` facilitates easy matrixing of variables through faceting, which is helpful for analysing change in legislative bodies over time. Note that we can facet the graph by any variable - a good use for `ggparliament` may be plotting roll call votes over a number of legislative sessions - although it is best to facet by a variable in which observations exist for all cases and, in general, faceting temporally makes intuitive sense. In the following example, I plot the House of Commons in the United Kingdom from 2010 to 2017 (see Figure 4). In doing so, it allows us to see the seat swing in the legislature over several election cycles.

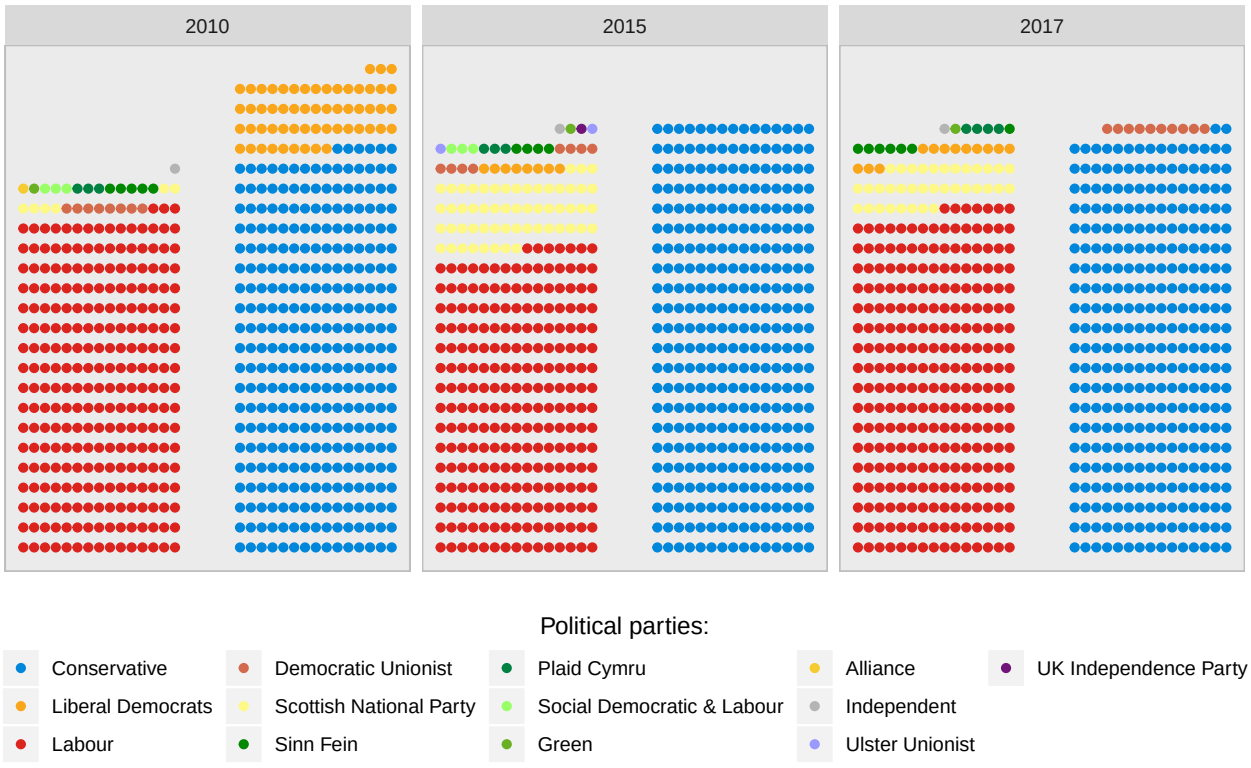

Figure 4: House of Commons small multiples

Another key advantage of `ggparliament` versus presenting descriptive data in a table is that users can collapse or add variables to plot, appending an additional visual layer each time. The parliament plot can be helpful when highlighting coalitions as it adds a visual dimension to the data. To show how visually highlighting seats can appeal, I provide the 2016 election results for the Australian House of Representatives in tabular format below as well as a `ggparliament` object.

Table 3: 2016 Australian federal election results

party	seats
Labor	69
Greens	1
Xenophon Team	1
Independent	2
Katter's Australian	1
Liberal	45
Liberal National	21
National	10

Table 4: Simplified election results

party	seats
Labor	69
Other	5
Coalition	76

There is nothing *incorrect* about either table. However, a parliament plot highlights members of the coalition in addition to plotting all of the parties that make up the coalition. For the purpose of visually displaying the parties that make up the Australian government compared to the opposition, a *simple* parliament plot provides as much utility as the tables above (i.e. very little – indeed, a visual drawing of Table 3). However, using visual tools to highlight subsets of the data allow for near-instant comprehension (see Figure 5). We gain the additional information from visually grouping and highlighting the coalition. In doing this, we do not *lose* information about individual parties.

Another quantity of interest is the absolute majority threshold. We can calculate this by halving the total number of seats in the legislature, rounding to the next integer to reach an absolute majority of seats. The formula for this metric is simply:

$$Absolute\ majority = \lfloor \frac{seats}{2} \rceil$$

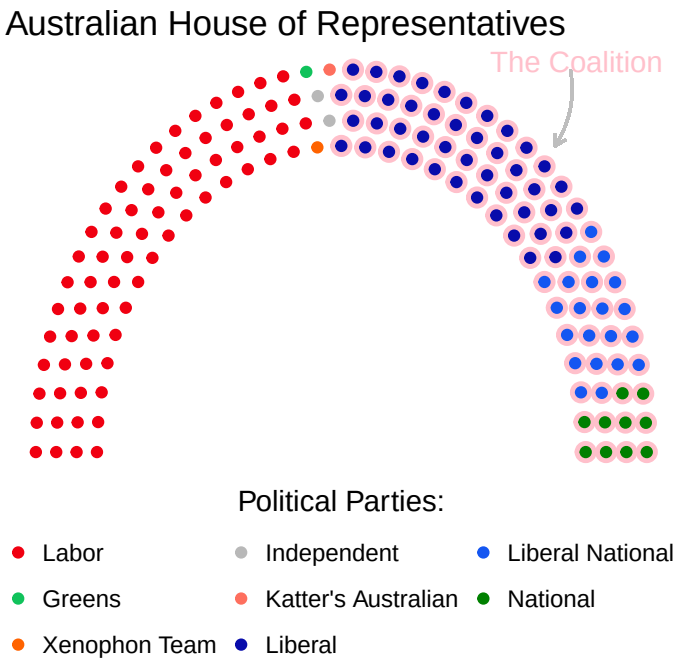

Figure 5: Highlighting the Coalition government in the Australian House of Representatives

Graphically, `ggparliament` draws a line through the majority threshold seat, with an optional annotation noting the number of seats required to claim a majority.

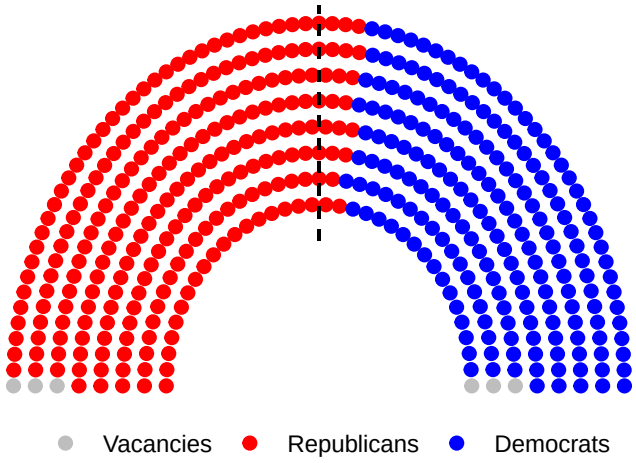

Figure 6: United States House of Representatives with a majority threshold line

Drawing the majority threshold may be of interest when the majority margin is slim. However, it also offers additional advantages in explicitly making clear the difference between the number of seats won by the government versus the bare minimum needed to call the election or vote.

The package also comes with several smaller functions. `ggparliament` provides a function to distinguish overhang seats, where the number of national-level party votes exceeds the number of division-level seats won in MMP electoral systems; the excess is allocated to the party to make up the difference. Additionally, the `R` package includes a function to calculate and plot seat share in the legislative body.

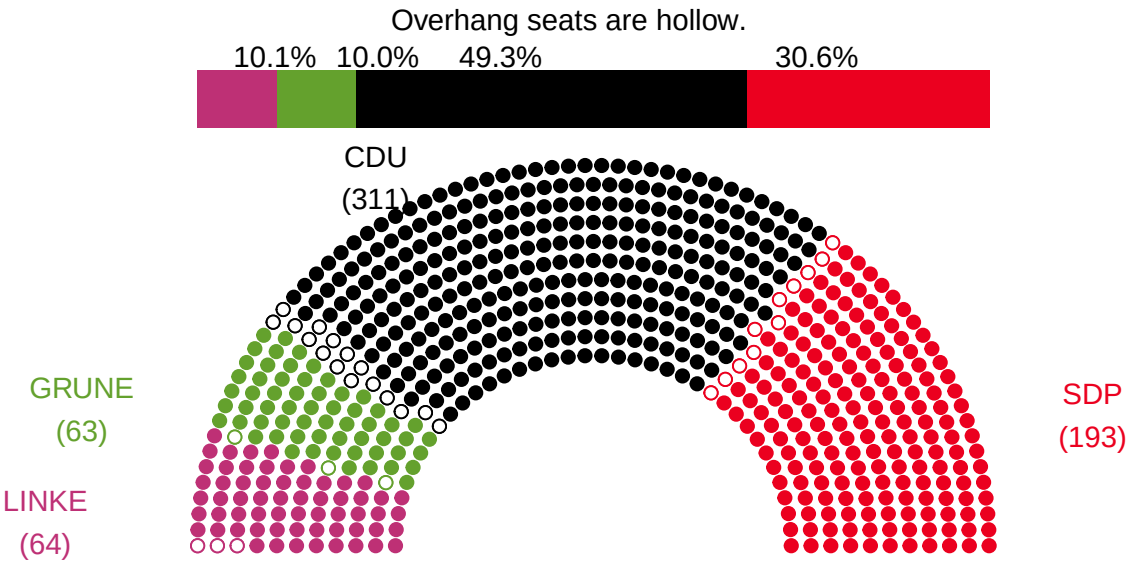

Figure 7: German Bundestag - 2013 election

Case Studies

I turn to additional examples where `ggparliament` may be a useful descriptive tool for political scientists. In one case study, I replicate an existing diagram of a legislature using `ggparliament`. In the other, I show how a `ggparliament` plot adds to the written description of the data.

The 115th Congress and Questions of Party Unity in a Polarized Era (Lee 2018)

Lee argues that despite the Republican majority in Congress and near-uniform party unity, the Republican party in the 115th Congress has been less effective at passing its legislative agenda because of internal party politics (2018). Given that parties often do not call a vote until a majority is ensured, roll-call data may overstate party cohesion.

A parliament plot showing Republican party cohesion in the 115th Congress looks like the following.

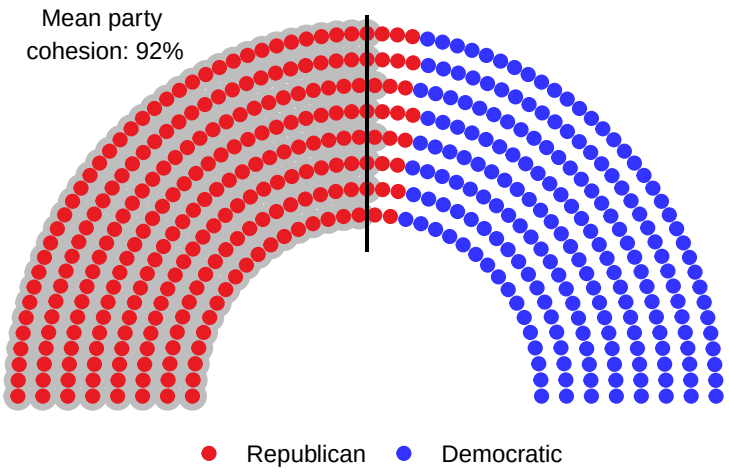

Figure 8: Party cohesion in the 115th Congress

Where You Sit is Where You Stand: The Impact of Seating Proximity on Legislative Cue-Taking (Masket 2008)

Masket shows that deskmate pairs in the California legislative assembly vote identically even if they are not ideologically similar (2008). This suggests that vote cue-taking, to some extent, is influenced by the location of fellow legislators. The author provides a diagram which depicts the 1949 California assembly. As evidenced in the figures below, seating was not always partisan. I replicate Masket's figure in `ggparliament` immediately below.

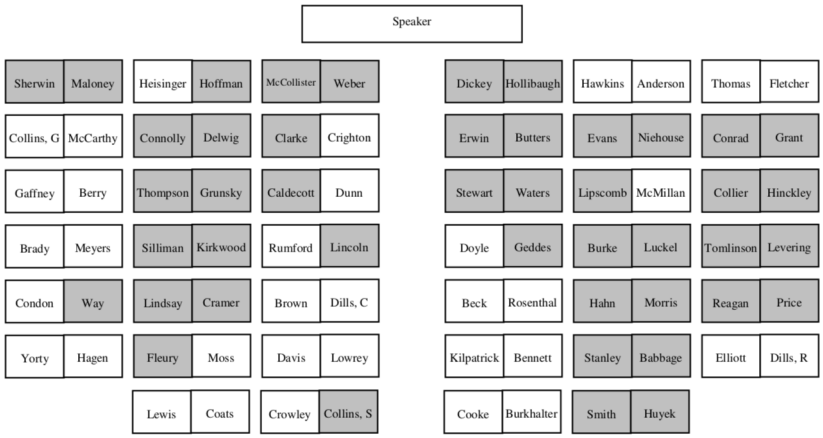

Figure 1. Seating assignments in the 1949 California assembly. Republican members are marked with gray desks, Democratic members are marked with white desks.
Source: *Assembly Final History, 1949 Regular Session*, pp. 30, 31.

Figure 9: Figure 1 in Masket (2008)

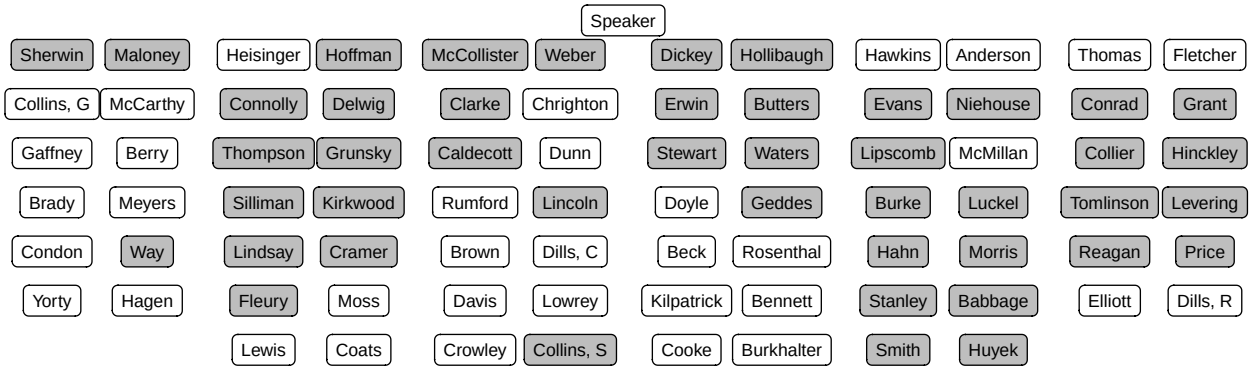

Figure 10: Replication of Figure 1 in Masket (2008)

Conclusion

Blah blah Future direction, etc.

References

Brambor, Thomas, William Roberts Clark, and Matt Golder. 2006. "Understanding Interaction Models: Improving Empirical Analyses." *Political Analysis* 14 (1). Cambridge University

Press: 63–82.

Brossard, Geoffrey. 2016. *D3-Parliament: A Parliament Chart Based on D3*. https://github.com/geoffreybr/d3-parliament.

Gelman, Andrew. 2011. "Why Tables Are Really Much Better Than Graphs." *Journal of Computational and Graphical Statistics* 20 (1): 3–7.

Greenhill, Brian, Michael D Ward, and Audrey Sacks. 2011. "The Separation Plot: A New Visual Method for Evaluating the Fit of Binary Models." *American Journal of Political Science* 55 (4): 991–1002.

Hainmueller, Jens, Jonathan Mummolo, and Yiqing Xu. 2018. "How Much Should We Trust Estimates from Multiplicative Interaction Models? Simple Tools to Improve Empirical Practice."

Hickman, Robert, Zoe Meers, and Thomas J. Leeper. 2017. *Ggparliament: Parliament Plots*. https://github.com/robwhickman/ggparliament.

Kastellec, Jonathan P., and Eduardo L. Leoni. 2007. "Using Graphs Instead of Tables in Political Science." *Perspectives on Politics* 5 (4): 755–71. http://www.jstor.org/stable/20446574.

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." *American Journal of Political Science*, 347–61.

Lee, Frances E. 2018. "The 115th Congress and Questions of Party Unity in a Polarized Era." *The Journal of Politics* 80 (4). University of Chicago Press Chicago, IL: 000–000.

Masket, Seth E. 2008. "Where You Sit Is Where You Stand: The Impact of Seating Proximity on Legislative Cue-Taking." *Quarterly Journal of Political Science* 3: 301–11.

Pepinsky, Thomas B. 2018. "Visual Heuristics for Marginal Effects Plots." *Research &*

Politics 5 (1).

Richfield, David. 2018. *Parliamentdiagram*. https://tools.wmflabs.org/parliamentdiagram/parlitest.php.

Tens, Julius. 2017a. *Parliament Svg*. https://github.com/juliuste/parliament-svg.

———. 2017b. *Westminster Svg*. https://github.com/juliuste/westminster-svg.

Tomz, Michael, Jason Whittenberg, and Gary King. 2003. "Clarify: Software for Interpretng and Presenting Statistical Results." In. American Statistical Association.

Tufte, Edward R, Nora Hillman Goeler, and Richard Benson. 1990. *Envisioning Information*. Vol. 126. Graphics press Cheshire, CT.

Tufte, Edward, and P Graves-Morris. 1983. "The Visual Display of Quantitative Information."

Wickham, Hadley. 2010. "A Layered Grammar of Graphics." *Journal of Computational and Graphical Statistics* 19 (1). Taylor & Francis: 3–28.

———. 2017. *Tidyverse: Easily Install and Load "Tidyverse" Packages*. https://CRAN.R-project.org/package=tidyverse.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, and Kara Woo. 2018. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*.

Wilkinson, Leland. 2006. *The Grammar of Graphics*. Springer Science & Business Media.

Appendix

`R` package

See https://CRAN.R-project.org/package=ggparliament.

Paper GitHub repo

https://github.com/zmeers/ggparliament_paper