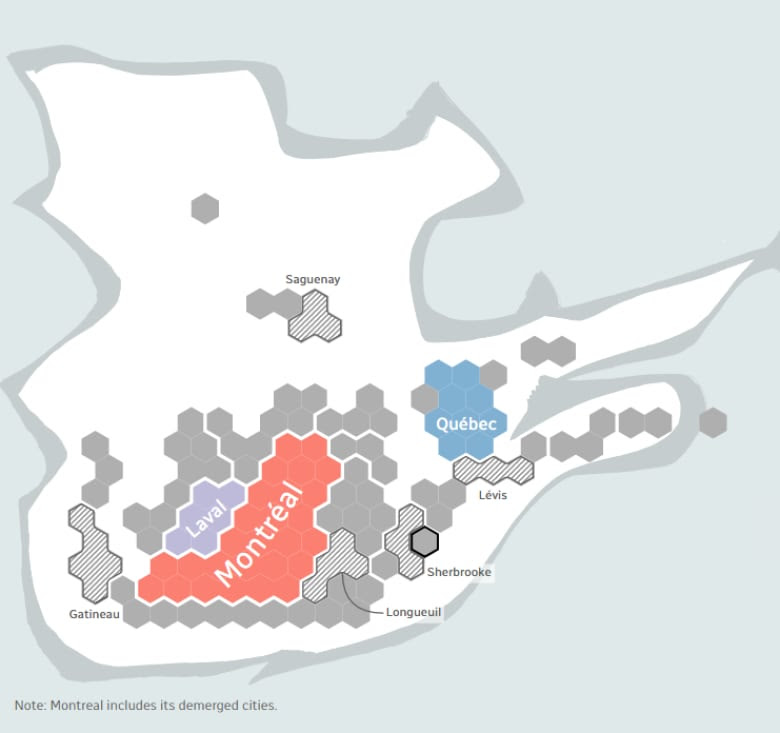
The CBC covered the recent (as of the original post-time on this blog entry) [Quebec elections](https://www.cbc.ca/news/canada/montreal/8-maps-that-explain-quebec-s-surprise-election-results-1.4848794) and used a well-crafted hex grid map to display results:

[](https://rud.is/b/2018/10/10/geojson-version-of-cbc-quebec-ridings-hex-cartograms-with-example-usage-in-r/quebec-hexagon-electoral-map-ridings/)

They have a [great ‘splainer](https://ici.radio-canada.ca/special/2018/elections-quebec/circonscriptions-portrait-cartes-vote-analyse-politique/en) on why they use this type of map.

Thinking that it may be useful for others, I used a browser Developer Tools inspector to yank out the javascript-created SVG and wrangled out the hexes using [svg2geojson[](https://www.npmjs.com/package/svg2geojson)](https://www.npmjs.com/package/svg2geojson) and put them into a [GeoJSON file](https://gitlab.com/hrbrmstr/quebec-hex-ridings) along with some metadata that I extracted from the minified javascript from the CBC’s site and turned into a data frame using the [V8[](https://cran.r-project.org/web/packages/V8/index.html)](https://cran.r-project.org/web/packages/V8/index.html) package. Since most of the aforementioned work was mouse clicking and ~8 (disjointed) lines of accompanying super-basic R code, there’s not really much to show wrangling-wise1, but I *can* show an example of using the new GeoJSON file in R and the [sf[](https://cran.r-project.org/web/packages/sf/index.html)](https://cran.r-project.org/web/packages/sf/index.html) package:

library(sf)

library(ggplot2)

# get the GeoJSON file from: https://gitlab.com/hrbrmstr/quebec-hex-ridings or https://github.com/hrbrmstr/quebec-hex-ridings

sf::st\_read("quebec-ridings.geojson", quiet = TRUE, stringsAsFactors = FALSE) %>%

ggplot() +

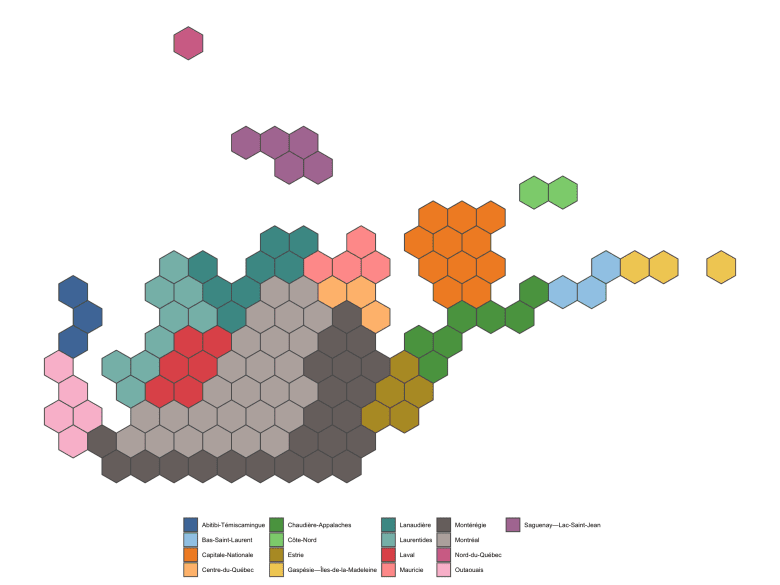
geom\_sf(aes(fill = regionname)) +

coord\_sf(datum = NA) +

ggthemes::scale\_fill\_tableau(name = NULL, "Tableau 20") +

ggthemes::theme\_map() +

theme(legend.position = "bottom")

[](https://rud.is/b/2018/10/10/geojson-version-of-cbc-quebec-ridings-hex-cartograms-with-example-usage-in-r/ggplot-sf/)

And, with a *little more* ggplot2-tweaking and some magick, we can even put it in the CBC-styled border:

library(sf)

library(magick)

library(ggplot2)

plt <- image\_graph(1488, 1191, bg = "white")

sf::st\_read("quebec-ridings.geojson", quiet=TRUE, stringsAsFactors=FALSE) %>%

ggplot() +

geom\_sf(aes(fill=regionname)) +

coord\_sf(datum=NA) +

scale\_x\_continuous(expand=c(0,2)) +

scale\_y\_continuous(expand=c(0,0)) +

ggthemes::theme\_map() +

theme(plot.margin = margin(t=150)) +

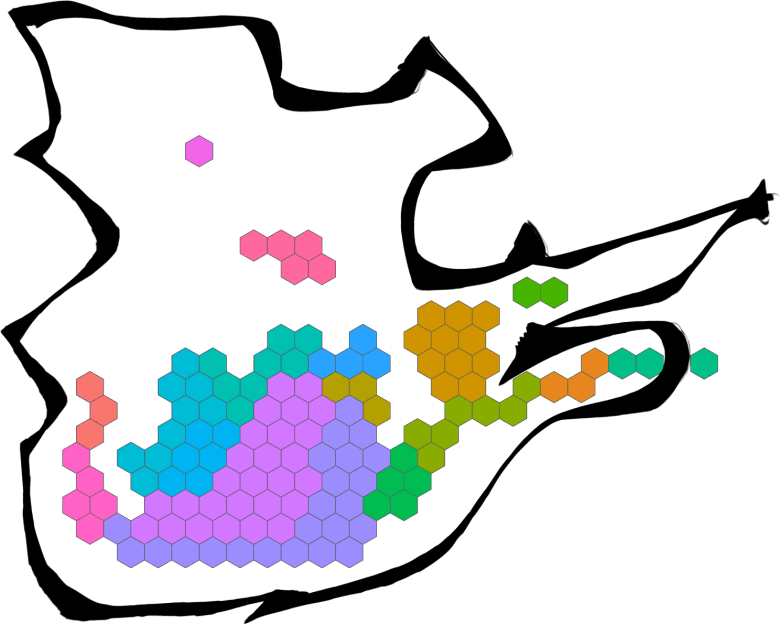
theme(legend.position = "none")

dev.off()

# get this bkgrnd img from the repo

image\_composite(plt, image\_read("imgs/background.png")) %>%

image\_write("imgs/composite-map.png")

[](https://rud.is/b/2018/10/10/geojson-version-of-cbc-quebec-ridings-hex-cartograms-with-example-usage-in-r/composite-map/)

You can tweak the border color with [magick[](https://cran.r-project.org/web/packages/magick/index.html)](https://cran.r-project.org/web/packages/magick/index.html) as needed and there’s a background2.png in the imgs directory in the repo that has the white inset that you can further composite as needed.

With a teensy bit of work you should be able adjust the stroke color via aes() to separate things as the CBC did.

**FIN**

It’s important to re-state that the CBC made the original polygons for the hexes (well, they made a set of grid points and open source software turned it into a set of SVG paths) and the background images. All I did was some extra bit of wrangling and conversionating2.

1 I can toss a screencast if there’s sufficient interest.  
2 Totally not a word.