Using R and Excel to analyze gerrymandering

Heather Dye

October 15, 2017

Background

This demonstration file was developed at the Geometry of Redistricting Conference in Wisconsin. https://sites.google.com/site/georedistrictingwisc/ Special thanks to the organizers!

Background information about gerrymandering and measures of compactness are available in this paper: https://polmeth.polisci.wisc.edu/Papers/compact.pdf The area formula defined in the paper is a variation of Heron's formula.

Background information about spatial mapping in R and the regeos package is available in the following sources. https://github.com/Robinlovelace/Creating-maps-in-R https://cran.rstudio.com/web/packages/rgeos/rgeos.pdf

The goal of this tutorial is to provide enough background so that a person with minimal coding experience in R and a general education course in statistics can study gerrymandering in their own community.

Software

The software that I used included Excel, Rstudio, and a program called tabula to convert pdfs to csv files. Tabula: http://tabula.technology/

R packages:

```
library(geosphere)
library(rgdal)
library(rgeos)
library(ggmap)
#library(tidyr)
library(dplyr)
library(tmap)
```

Shapefiles

In this demonstration, I used shapefiles that describe the outline of Missouri's congressional districts as "spatial polygons" with attached data.

Missouri: http://geoportal.missouri.edu/geoportal/catalog/search/search.page

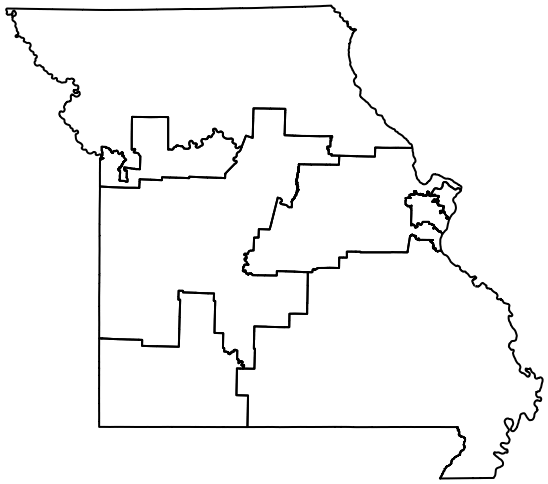
The missouri shapefiles are stored in a folder in the working directory called "MO_2016_TIGER_115th_Congressional_District
One problem that I experienced was that the projection co-ordinates were not correctly set. The projection co-ordinates (CRS) need to be set to mercator or longitude in order to compute area. Check that the shape file's co-ordinates are being correctly projected.

```
mo<-readOGR(dsn="MO_2016_TIGER_115th_Congressional_Districts_shp", layer ="MO_2016_TIGER_115th_Congressi
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "MO_2016_TIGER_115th_Congressional_Districts_shp", layer: "MO_2016_TIGER_115th_Congressional_
## with 8 features
## It has 12 fields
```

```
## Integer64 fields read as strings:  ALAND AWATER
```

```
plot(mo)
```



```
proj4string(mo)
```

```
## [1] "+proj=longlat +datum=NAD83 +no_defs +ellps=GRS80 +towgs84=0,0,0"
```

This command checks that the shapefile is equipped with a method of projection (Coordinate Reference System). In the next command, I set the CRS. I wound up switching between two methods of projection to do the computations

```
mo<-spTransform(mo, CRS=CRS("+proj=merc +ellps=GRS80 +units=us-mi"))

proj4string(mo)
```

```
## [1] "+proj=merc +ellps=GRS80 +units=us-mi"
```

The information included in the data files includes several data fields. We can strip off an individual district and examine it using commands from dplyr. In Rstudio, the command View(mo1) will set up a data table for you to inspect.

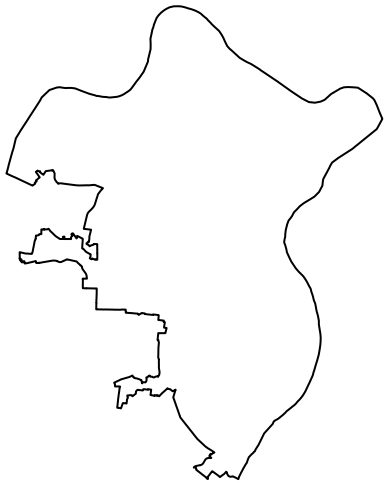
```
mo1<-mo[mo@data$CD115FP == "01", ]
head(mo1@data)
```

```
##   STATEFP CD115FP GEOID                NAMELSAD LSAD CDSESSN MTFCC
## 0      29      01  2901 Congressional District 1   C2     115 G5200
##   FUNCSTAT     ALAND   AWATER    INTPTLAT     INTPTLON
## 0        N 583631669 27126111 +38.7283860 -090.2962282
```

```
sapply(mo1@data, class)
```

```
##  STATEFP  CD115FP    GEOID NAMELSAD     LSAD  CDSESSN    MTFCC FUNCSTAT
## "factor" "factor" "factor" "factor" "factor" "factor" "factor" "factor"
##    ALAND   AWATER INTPTLAT INTPTLON
## "factor" "factor" "factor" "factor"
```

```
plot(mo1)
```



```
View(mo1@data)
```

The following commands are samples of the commands that I used to to compute compactness measures. To compute area using gArea, you need to access the invidual entries in the dat frame "mo". Using rgeos, the spatial commands required a mercator projection. The majority of the commands are from rgeos. To compute perimeter, I had to convert to a longitude/lattitude

```
mo1latlong<-spTransform(mo1,CRS = CRS("+proj=longlat"))
perimeter(mo1latlong)
```

```
## [1] 168445.3
```

```
gArea(mo1)
```

```
## [1] 386.4544
```

```
gLength(mo1)
```

```
## [1] 133.9591
```

```
gCentroid(mo1)
```

```
## class       : SpatialPoints
## features    : 1
## extent      : -6245.727, -6245.727, 2893.104, 2893.104  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=merc +ellps=GRS80 +units=us-mi
```

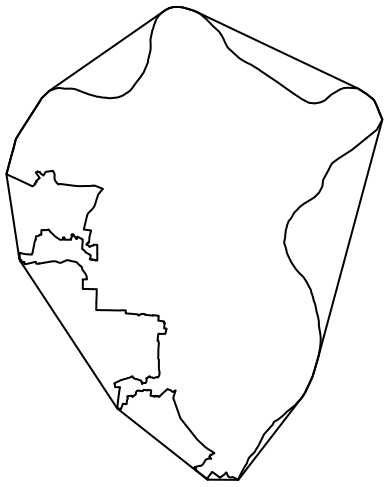
```
gConvexHull(mo1)
```

```
## class       : SpatialPolygons
## features    : 1
## extent      : -6258.714, -6233.495, 2875.832, 2907.555  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=merc +ellps=GRS80 +units=us-mi
```

We compute some measure of compactness and add them to the data frame. The column CD115FP identifies the congressional district as a character. I converted this to a numerical value for ease of reference. The Polsby-Popper measure is A(D)/P(D)^2. The idea is that a circle would have a score of 1/ 4 Pi: (Pi r^2)/(2 Pi r)^2. All other shapes would have a lower score. The convex hull measure is a ratio of A(D)/A(convexhull(D)). Convex objects would have a score of 1.

Here is a convex hull demonstration.

```
plot(mo1); plot(gConvexHull(mo1), add=TRUE)
```



```
mo@data$CD115FP<-as.numeric(mo@data$CD115FP)
mat<-matrix(numeric(), nrow=max(mo@data$CD115FP), ncol=6)
colnames(mat)<-c('cd', 'area', 'hullarea', 'perimeter', 'pols', 'hull')
```

```
for(i in array(1:max(mo@data$CD115FP))){mat[[i,1]]<-i
mat[[i,2]]<-gArea(mo[mo@data$CD115FP==i, ])
mat[[i,3]]<-gArea(gConvexHull(mo[mo@data$CD115FP==i, ]))}

molatlong<-spTransform(mo,CRS = CRS("+proj=longlat"))

for(i in array(1:max(mo@data$CD115FP))){mat[i,4]<- perimeter(molatlong)[i]
mat[i ,5]<-12*mat[ i, 2]/mat[ i,4]^2
mat[i,6]<-mat[i,2]/mat[i,3]}

print(mat)
```

```
##      cd       area   hullarea perimeter         pols      hull
## [1,]  1   386.4544   530.3919  168445.3 1.634413e-07 0.7286204
## [2,]  2   784.3427  1029.1155  280468.3 1.196519e-07 0.7621523
## [3,]  3 11414.3338 14753.0047 1006628.0 1.351742e-07 0.7736955
## [4,]  4 23790.1044 35188.1813 1439280.5 1.378119e-07 0.6760822
## [5,]  5  4078.6156  5919.5507  642279.4 1.186441e-07 0.6890076
## [6,]  6 31217.3981 40760.4558 1590926.7 1.480054e-07 0.7658746
## [7,]  7  9958.6009 12218.8454  677997.2 2.599704e-07 0.8150198
## [8,]  8 31555.3790 38853.1818 1362877.1 2.038643e-07 0.8121698
```

Attach the measures to the shapefiles

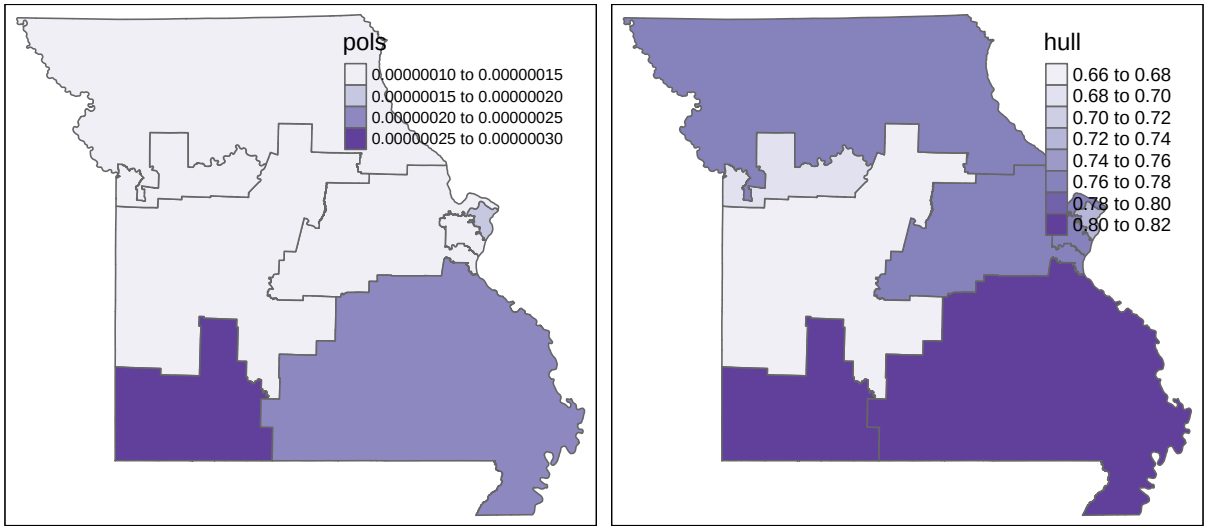
Commands from dplyr are used to attach the data to the shapefiles

```
mat<-data.frame(mat)
mo@data<-left_join(mo@data, mat, by=c('CD115FP'='cd'))
mo<-spTransform(mo, CRS=CRS("+proj=longlat"))
```

Plot the shapefiles

The next command plots the shapefiles with comparative information about the measures. Additional commands in the Lovelace tutorial can add in background map tiles.

```
#qtm(shp=mo, fill="pols", fill.palette ="Blues")
qtm(shp=mo, fill=c("pols", "hull"), fill.palette="Purples", ncol=2)
```



Adding in census data

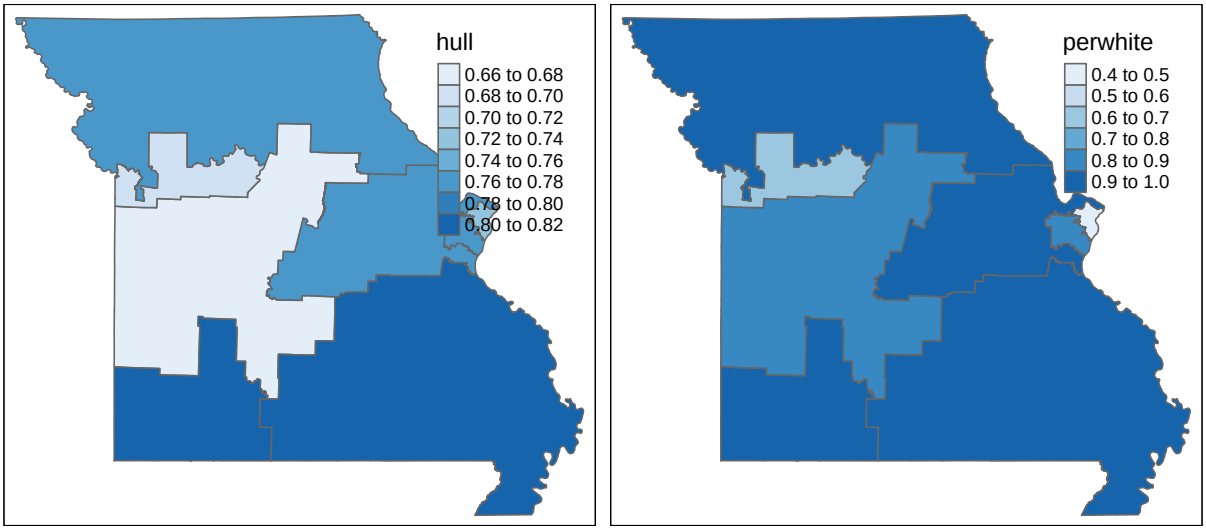
To obtain census data about congressional districts. https://www.census.gov/mycd/?st=17 Votes by county for the presidential election. http://www.cnn.com/election/results/states/missouri#president

To obtain counties by congressional districts https://www.census.gov/geo/maps-data/data/cd_state.html I did a very minimal amount of data cleaning in Excel - mainly computing totals by congressional district. In the case where a county was split between congressional districts, I simply added the county to one of the congressional districts. The data is stored in a csv file titled "MissouriCDcensusData.csv"

```
mocensus<-read.csv("MissouriCDCensusData.csv", header=TRUE)
mocensus<-mutate(mocensus, perwhite=mocensus$White/mocensus$Total.population)
```

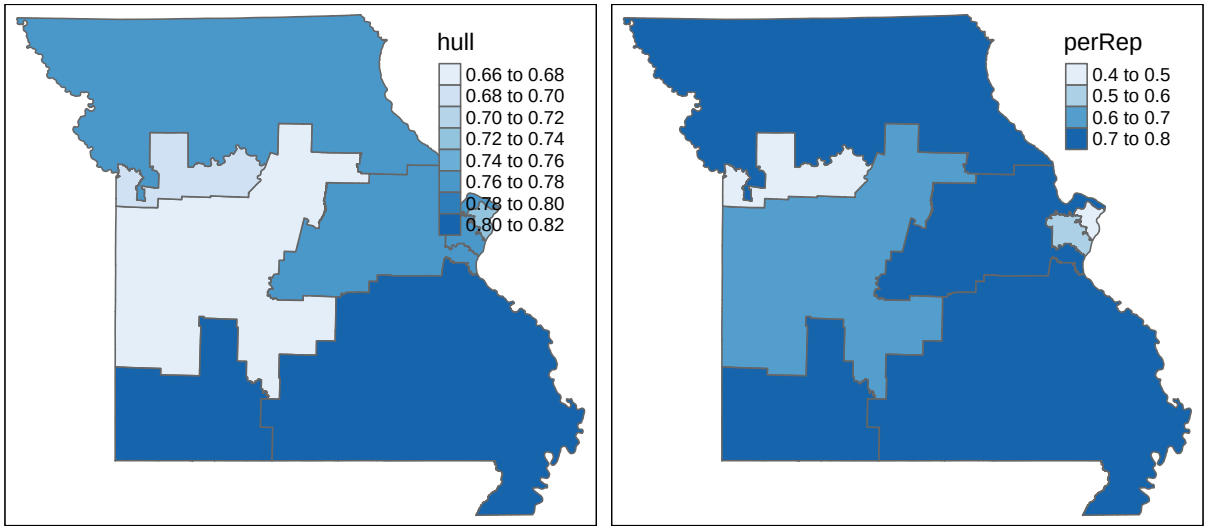
Now, I attach the data to my shapefile

```
mo@data<-left_join(mo@data,mocensus, by=c('CD115FP'='districts'))
qtm(shp=mo, fill=c("hull", "perwhite"), fill.palette="Blues", ncol=2)
```



Here are the results for the percent of voters who voted Republican in the 2016 election.

```
qtm(shp=mo, fill=c("hull", "perRep"), fill.palette="Blues", ncol=2)
```



Regression on convex hull measure versus race and voting results

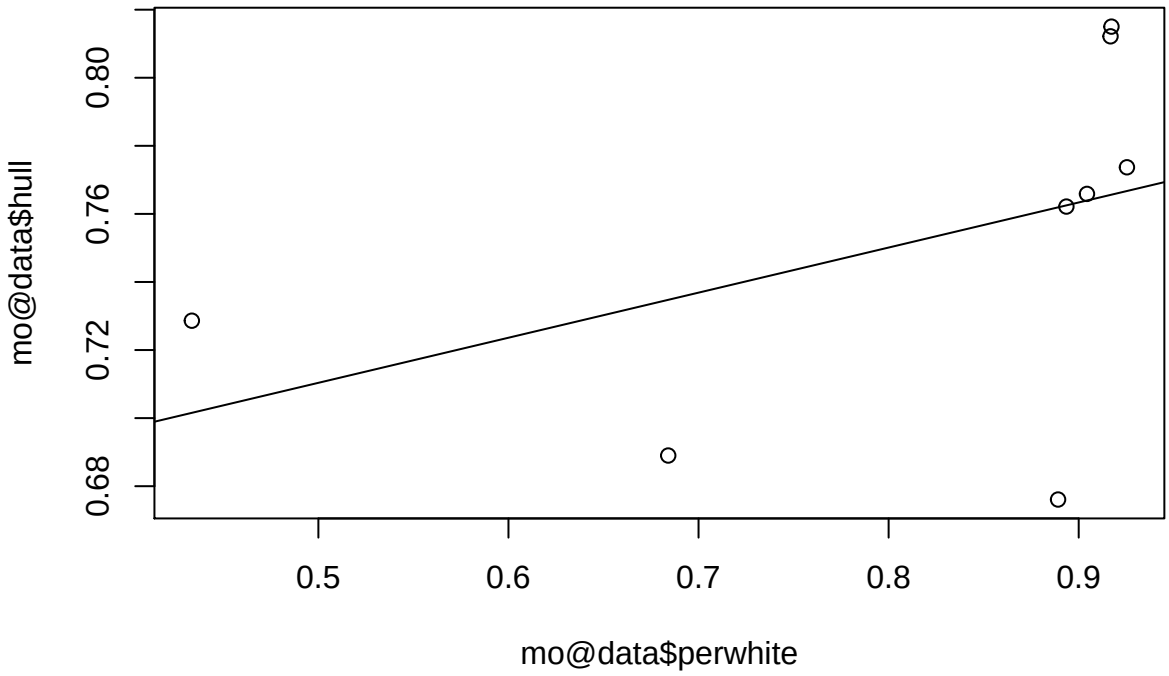
In the following two code snippets, we examine if there is a significant relationship between the convex hull measure and another variable.

```
glresults<-lm(mo@data$hull ~ mo@data$perwhite)
summary(glresults)
```

```
##
## Call:
## lm(formula = mo@data$hull ~ mo@data$perwhite)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.085841 -0.011701  0.004453  0.031949  0.049375
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.64413    0.08976   7.176  0.00037 ***
## mo@data$perwhite  0.13248    0.10728   1.235  0.26302
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04976 on 6 degrees of freedom
## Multiple R-squared:  0.2027, Adjusted R-squared:  0.06978
```

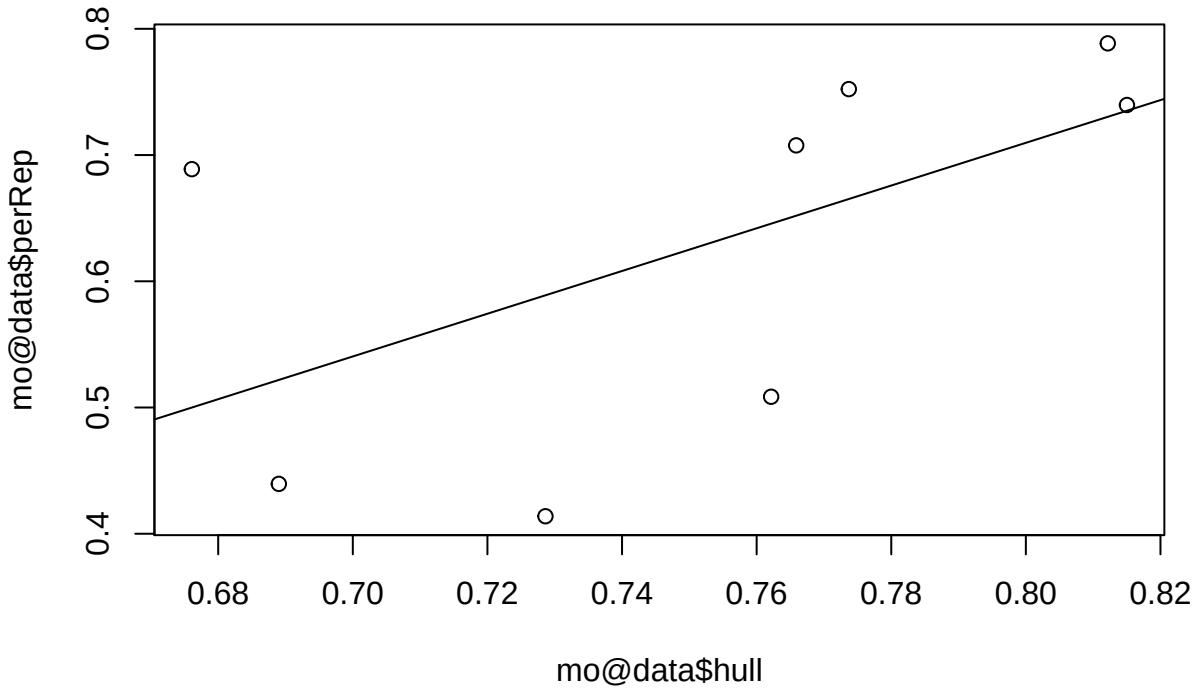
```
## F-statistic: 1.525 on 1 and 6 DF,  p-value: 0.263
```

```
plot(mo@data$perwhite, mo@data$hull)
abline(glresults)
```



regression on voting results

```
lmvotes=lm(mo@data$perRep ~ mo@data$hull)
plot(mo@data$hull, mo@data$perRep)
abline(lmvotes)
```




```
summary(lmvotes)
```

```
##
## Call:
## lm(formula = mo@data$perRep ~ mo@data$hull)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17502 -0.09603  0.03015  0.06543  0.18887
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.6443     0.7345  -0.877    0.414
## mo@data$hull   1.6925     0.9737   1.738    0.133
##
## Residual standard error: 0.1329 on 6 degrees of freedom
## Multiple R-squared:  0.3349, Adjusted R-squared:  0.2241
## F-statistic: 3.021 on 1 and 6 DF,  p-value: 0.1328
```