

# Spatial Statistics, Assignment 7

## South Dakota County Maps

Peter Claussen

February 9, 2020

## South Dakota Corn Yield

```
> library(maps)
```

I've frequently used the USDA National Agricultural Statistics Service web site for crop yield data. However, it becomes tiresome to download all the different types of data I'd like to examine, and to read these into R. So, I've gotten key that allows me to query the database directly.

We can get CSV formatted data, using a query of the form

```
http://quickstats.nass.usda.gov/api/api_GET/?key=api key&
commodity_desc=CORN&year__GE=2012&state_alpha=VA&format=CSV
```

I've defined `api_key` as a hidden variable in R and will use this to produce a query URL.

For county level data, we want to use the code `agg_level_desc` set to "COUNTY". "STATE" or "AG DISTRICT" as also supported. We only want the yearly total, so we need to include `freq_desc` set to "ANNUAL". More codes are available at <http://quickstats.nass.usda.gov/api>

```
> query = "http://quickstats.nass.usda.gov/api/api_GET/?key="
> query = paste(query, api_key, sep="")
> query = paste(query, "&commodity_desc=CORN&year__GE=2014&state_alpha=SD", sep="")
> query = paste(query, "&agg_level_desc=COUNTY&freq_desc=ANNUAL", sep="")
> query = paste(query, "&format=CSV", sep="")

> query = "http://quickstats.nass.usda.gov/api/api_GET/?key="
> query = paste(query, api_key, sep="")
> query = paste(query, "&commodity_desc=CORN&year=2014&state_alpha=SD", sep="")
> query = paste(query, "&agg_level_desc=COUNTY&freq_desc=ANNUAL", sep="")
> query = paste(query, "&format=CSV", sep="")
```

There are a couple packages for submitting web page queries via R, including one (`RCurl`) that formats parameters, but for this short URL, we can just use the `url` command and read from the returned CSV string into a data table.

```
> sdcorn.tab <- read.csv(url(query))
```

Check the statistics that are returned. There will be multiple types returned, so we will need to select a subset of the data.

```
> levels(sdcorn.tab$statisticcat_desc)
```

```
[1] "AREA HARVESTED" "AREA PLANTED" "PRODUCTION" "YIELD"
```

We'll consider Yield, first. Convert the NASS County Name to something maps can use.

```

> sdcorn.tab <- subset(sdcorn.tab, sdcorn.tab$statisticcat_desc == "YIELD")
> #sdcorn.tab <- subset(sdcorn.tab, sdcorn.tab$statisticcat_desc == "AREA PLANTED")
> sdcorn.tab <- subset(sdcorn.tab, sdcorn.tab$county_name != "OTHER (COMBINED) COUNTIES")
> sdcorn.tab$county_name <- tolower(as.character(sdcorn.tab$county_name))
> sdcorn.tab$Value <- as.numeric(sdcorn.tab$Value)
> sdcorn.tab$yield_ratio = sdcorn.tab$Value/max(sdcorn.tab$Value)
> #maps need appended state name
> sdcorn.tab$map_name <- paste("south dakota",sdcorn.tab$county_name,sep=",")

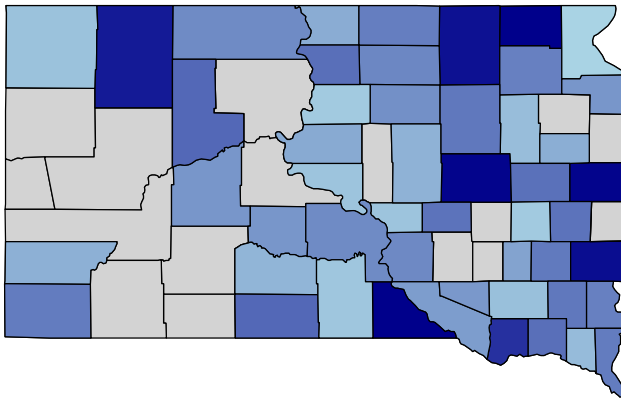
```

Yield data is missing for some counties, but we'll want to plot all county outlines.

```

> yield.percent <- floor(sdcorn.tab$yield_ratio*100)+1
> map.colors <- colorRampPalette(c('lightblue','darkblue'))(100)[yield.percent]
> map("county", "south dakota", col = "lightgray", fill = TRUE, lty = 1, lwd = 1)
> map("county", regions = sdcorn.tab$map_name,
+   col = map.colors, fill = TRUE, lty = 1,
+   lwd= 1, add = TRUE)

```



To generate some statistics for these plots, we start with a map object.

```

> corn.map <- map("county", regions = sdcorn.tab$map_name,
+   col = map.colors, fill = TRUE, plot=FALSE)

```

Then convert to polygons, using `maptools`

```
> library(maptools)
> corn.poly <- map2SpatialPolygons(corn.map, corn.map$names)
```

Next, get neighbor information. For this, we use `spdep`, and accept defaults.

```
> library(spdep)
> corn.nb <- poly2nb(corn.poly)
> corn.wt <- nb2listw(corn.nb)
```

Now we can compute a global Moran's  $I$ .

```
> moran.test(sdcorn.tab$Value, corn.wt)
```

Moran I test under randomisation

```
data:  sdcorn.tab$Value
weights: corn.wt
```

Moran I statistic standard deviate = 0.59036, p-value = 0.2775

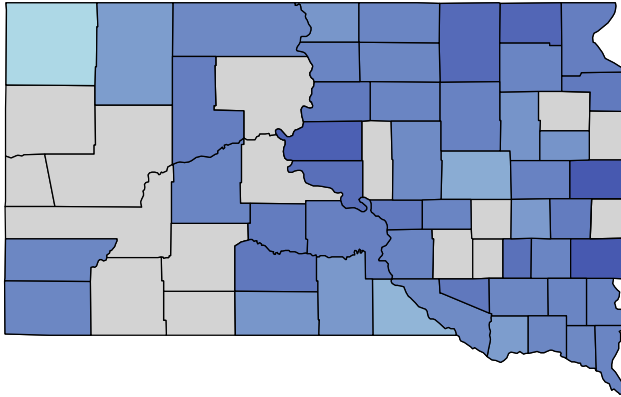
alternative hypothesis: greater

sample estimates:

Moran I statistic	Expectation	Variance
0.04121280	-0.02000000	0.01075113

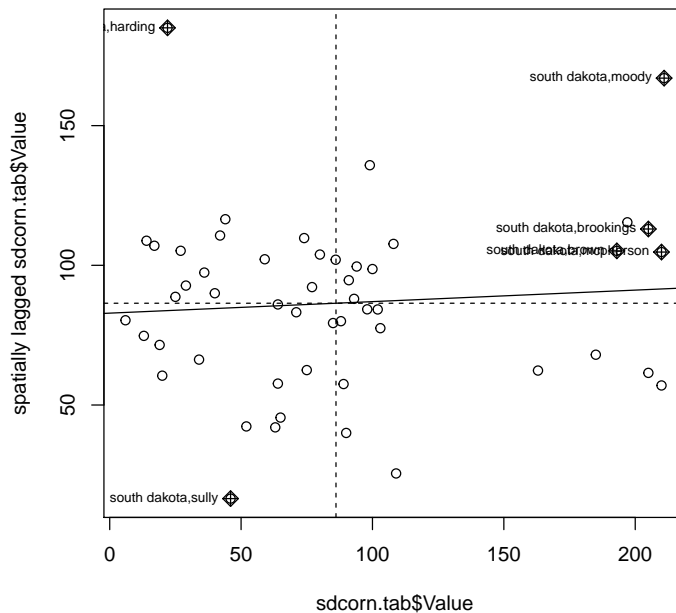
This is strange; it suggests that yield for corn is randomly distributed. Looking at the original map, I would have expected a negative value. So, we should plot the local  $I$ .

```
> corn.moran <- localmoran(sdcorn.tab$Value, corn.wt)
> iratio <- corn.moran[,1]
> min.i <- min(iratio)
> max.i <- max(iratio)
> rng.i <- max.i-min.i
> iratio <- (iratio - min.i)/rng.i
> yield.percent <- floor(iratio*100)+1
> map.colors <- colorRampPalette(c('lightblue', 'darkblue'))(100)[yield.percent]
> map("county", "south dakota", col = "lightgray", fill = TRUE, lty = 1, lwd = 1)
> map("county", regions = sdcorn.tab$map_name,
+     col = map.colors, fill = TRUE, lty = 1,
+     lwd = 1, add = TRUE)
```



We can consider the Moran's plot.

```
> moran.plot(sdcorn.tab$Value, corn.wt)
```



Compare this with Geary's and Getis-Ord

```
> geary.test(sdcorn.tab$Value,corn.wt)
```

Geary C test under randomisation

```
data: sdcorn.tab$Value
weights: corn.wt
```

```
Geary C statistic standard deviate = 0.73293, p-value = 0.2318
alternative hypothesis: Expectation greater than statistic
sample estimates:
```

Geary C statistic	Expectation	Variance
0.92073788	1.00000000	0.01169519

```
> corn.wb <- nb2listw(corn.nb,style="B")
> globalG.test(sdcorn.tab$Value,corn.wb)
```

Getis-Ord global G statistic

```
data: sdcorn.tab$Value
weights: corn.wb
```

```
standard deviate = 1.0989, p-value = 0.1359
```

alternative hypothesis: greater

sample estimates:

Global G statistic	Expectation	Variance
0.08978083	0.08156863	0.00005585

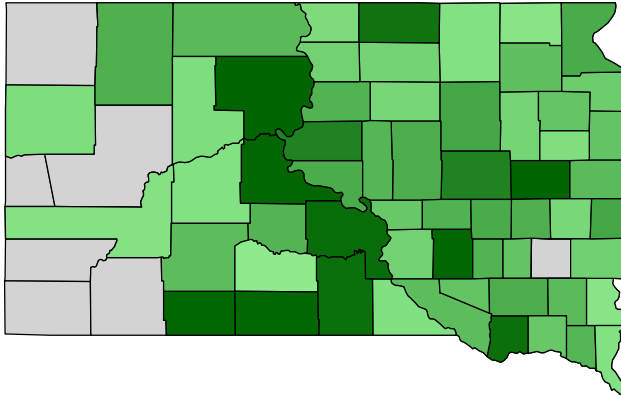
Neither show significance, so we conclude that corn acres planted in South Dakota in 2014 is randomly distributed.

## Soybeans

Let's try to repeat this with soybean data.

```
> query = "http://quickstats.nass.usda.gov/api/api_GET/?key="
> query = paste(query, api_key, sep="")
> query = paste(query, "&commodity_desc=SOYBEANS&year__GE=2014&state_alpha=SD", sep="")
> query = paste(query, "&agg_level_desc=COUNTY&freq_desc=ANNUAL", sep="")
> query = paste(query, "&format=CSV", sep="")
> sdsoybean.tab <- read.csv(url(query))
> sdsoybean.tab <- subset(sdsoybean.tab,
+   sdsoybean.tab$statisticcat_desc == "AREA HARVESTED")
> sdsoybean.tab <- subset(sdsoybean.tab,
+   sdsoybean.tab$county_name != "OTHER (COMBINED) COUNTIES")
> sdsoybean.tab$county_name <- tolower(as.character(sdsoybean.tab$county_name))
> sdsoybean.tab$Value <- as.numeric(sdsoybean.tab$Value)
> sdsoybean.tab$yield_ratio = sdsoybean.tab$Value/max(sdsoybean.tab$Value)
> sdsoybean.tab$map_name <- paste("south dakota", sdsoybean.tab$county_name, sep=", ")

> yield.percent <- floor(sdcorn.tab$yield_ratio*100)+1
> map.colors <- colorRampPalette(c('lightgreen', 'darkgreen'))(100)[yield.percent]
> map("county", "south dakota", col = "lightgray", fill = TRUE, lty = 1, lwd = 1)
> map("county", regions = sdsoybean.tab$map_name,
+   col = map.colors, fill = TRUE, lty = 1,
+   lwd = 1, add = TRUE)
```



Repeat the spatial statistics from above.

```
> beans.map <- map("county", regions = sdsoybean.tab$map_name,
+                 col = map.colors, fill = TRUE, plot=FALSE)
> beans.poly <- map2SpatialPolygons(beans.map, beans.map$names)
> beans.nb <- poly2nb(beans.poly)
> #beans.wt <- nb2listw(beans.nb)
```

So, here's a problem - we can't get weights for soybean data. Soybeans are not as commonly grown across South Dakota, and there is one isolated county. `nb2listw` won't accept this map. That's odd in itself. This county appears to have had a large number of acres harvested, but the neighboring counties reported no acreage planted in soybeans. There are two things I could do with this. I could delete the isolated county, but that result in a misleading analysis - this county would contribute to a negative Moran's  $I$ . I could generate a full list of county data, and include 0 values for the missing counties. But this is also unsatisfying, since we can't assume there were not soybeans planted, only that there were no acres reported. So, I'll just leave off here, for now.