This is the fun section to show how to manipulate and visualize spatial data in R. It seems more fun to start with a data story; and work through some basic analysis.

Data.gov is a data repository administered by the US government with over 445,000 geographic data sets. One dat set is the geographic coordinates and characteristics of 7,863 farmers markets in the United States.[1] Suppose we are interested in examining the effect of state boundaries on the characteristics of farmers markets. Do state boundaries have a substantive impact on the character of farmers markets, or are the no better than arbitrary lines. There are rigorous ways to address this question, but we will just classify and plot farmers markets, looking for spatial patterns that follow state boundaries.

First, we export the farmers market data set as a CSV file, saving it to the data directory as `farmers-mkts.csv`. Let's just plot the distribution of farmers markets on a a base map of the United States. To do this, you will need to install and then load the following libraries:

```
library(maps)
library(maptools)
library(RColorBrewer)
library(classInt)
library(gpclib)
library(mapdata)
```

The following code plots the map, and presents the results in Figure 1.

```
data <- read.csv("../data/farmers-mkts.csv", header = TRUE)
map("state", interior = FALSE)
map("state", boundary = FALSE, col = "gray", add = TRUE)
points(data$x, data$y, cex = 0.2, col = "blue")
```

The distribution of farmers markets across the US is neat to see, but there are so many points that it is difficult to visually glean any useful information, as seen in the following figure. So, instead, let's consider farmers markets in Colorado, Utah, New Mexico, and Arizona. There are 354 farmers markets in these four states.

```
statelist <- c("New Mexico", "Colorado", "Arizona", "Utah")
state.data <- data[is.element(data$State, statelist), ]
dim(state.data)
```

```
 [1] 354  32
```

Each column of the `state.data` data frame contains information on a different attribute of the farmers markets. The last 24 columns are binary variables with entries `"Y"` or `"N"`, indicating whether the market sells cheese, for example, or accepts credit cards. These are the attributes of interest. The idea is whether we can predict the state of the farmers market, purely based on the characteristics. We can extract these characteristics into a matrix $\mathbf{X}$ and recode the string variables to 0-1 binary variables. Note that the rows are still labelled according to the countrywide index of the farmers market.

```
X <- state.data[, 8:ncol(state.data)]
X <- apply(X, 2, function(col) { ifelse(col == "Y", 1, 0) })
X[1:6, c("Honey", "Jams", "Poultry")]
```

[1] `https://explore.data.gov/d/wfna-38ey`

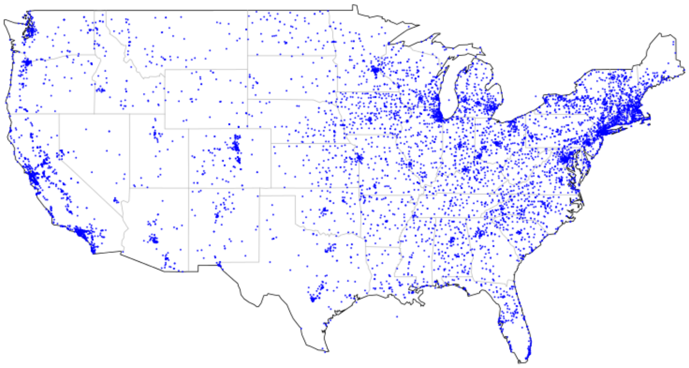

Figure 1: Distribution of US farmers markets

```
   Honey Jams Poultry
23     0    0       0
52     0    0       0
53     0    0       0
54     1    1       0
76     0    0       0
77     1    1       1
```

We want to categorize the markets based on the market features. Specifically, we would like to tag each market with one of the four states, or four distinct buckets. For this, we can use K-Means clustering, which is a simple call in `R`. The resulting object `cl` is a list with various output attributes of the clustering, including to which of the four clusters each market was assigned, indicated by `cl$cluster`. This is the vector we will use to color the points on the map. There are many other attributes that are collected in `cl`; three of them are listed below.

```
cl <- kmeans(X, 4)
names(cl)[1:3]
```

```
[1] "cluster" "centers" "totss"
```

The following code plots the points by zooming in on the four states, and coloring each market by the predicted state. There is slight variation in the output map, as `R` selects the exact colors on the fly. Below, we save the resulting image to `inserts/limited-mkts.png` and display one such image, produced previously. We set the number of colors to 4, one for each state; and the `brewer.pal()` function sets separate color codes for the number of supplied classes according to the color scheme `Set1`.

```
nclr <- 4
clr.set <- brewer.pal(nclr, "Set1")
```

There exactly four cluster values to match the four-color palette. However, the `classIntervals()` assigns each color in the palette to a range of values for the more general case. The `findColours()` function then maps the color codes to each point in the cluster vector; and this is the schema that is used to color the points on the map.

```
class <- classIntervals(cl$cluster, nclr, style = "pretty")
colcode <- findColours(class, cl$cluster)
```

The map is generated in much the same way as the full US map, except that the extent is limited by a latitude-longitude bounding box.

```
map("state", interior = FALSE,
    xlim = c(-117, -101), ylim = c(28, 43))
map("state", boundary = FALSE, col="gray", add = TRUE,
    xlim = c(-117, -101), ylim = c(28, 43))
points(state.data$x, state.data$y, pch = 16, col = colcode, cex = 1)
```

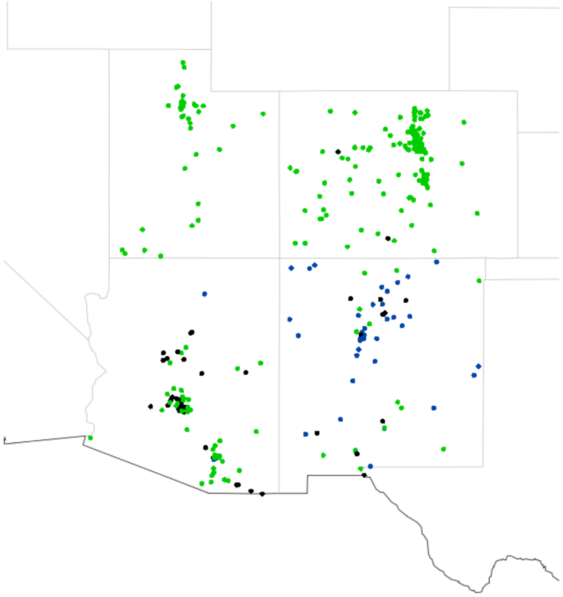

Figure 2: Distribution of US farmers markets

This is cool. It is clear that there is some separable variation in the characteristics of the markets. Even markets right across the border have characteristics that are different enough to "correctly" cluster them by state. There must be something about state regulations that are particularly imposing in New Mexico. To validate this observation, we can calculate the proportion of the total variance that can be explained by the clustering.

```
cl[["betweenss"]] / cl[["totss"]]
```

```
[1] 0.6123948
```

K-means clustering minimizes the within-group sum of squared errors by effectively maximizing the between group sum of squares. We have achieved a 60% reduction in the sum of squares through clustering. Further evaluation of the fit is probably beyond the scope of this section. Scratch that, I am leading this section, so I can say for certain that it is beyond the scope of this section. Boom. The purpose of this section, anyway, was just to introduce the mapping facilities in R, available geospatial data, and some basic analysis outside the scope of econometrics.

One last thing, though: Suppose we wanted to find out whether the likelihood of selling certain products at market is related to the latitude, conditional on the longitude. In otherwords, suppose we wanted to identify whether cheese is sold more often in the North than in the South. Try this (but don't read into it too much):

```
cheese.model <- glm(Cheese ~ 1 + x + y, data = data, family = "binomial")
coef(summary(cheese.model))
```

```
               Estimate  Std. Error   z value     Pr(>|z|)
(Intercept) -0.99920466 0.260410677 -3.837034 1.245292e-04
x            0.00790756 0.001485875  5.321821 1.027336e-07
y            0.01948108 0.005278786  3.690447 2.238602e-04
```

I guess you'd conclude that as you move north and west, more cheese is sold; but there is likely a lot more to it than this. We will explore the data a bit more in section.

This is just one of many neat examples about how to use R to examine spatial variation; and to effectively uncover some spatial data generating processes that are not obvious from looking at the data in a matrix or spreadsheet. More to come!