Creating Rate Change Maps in R

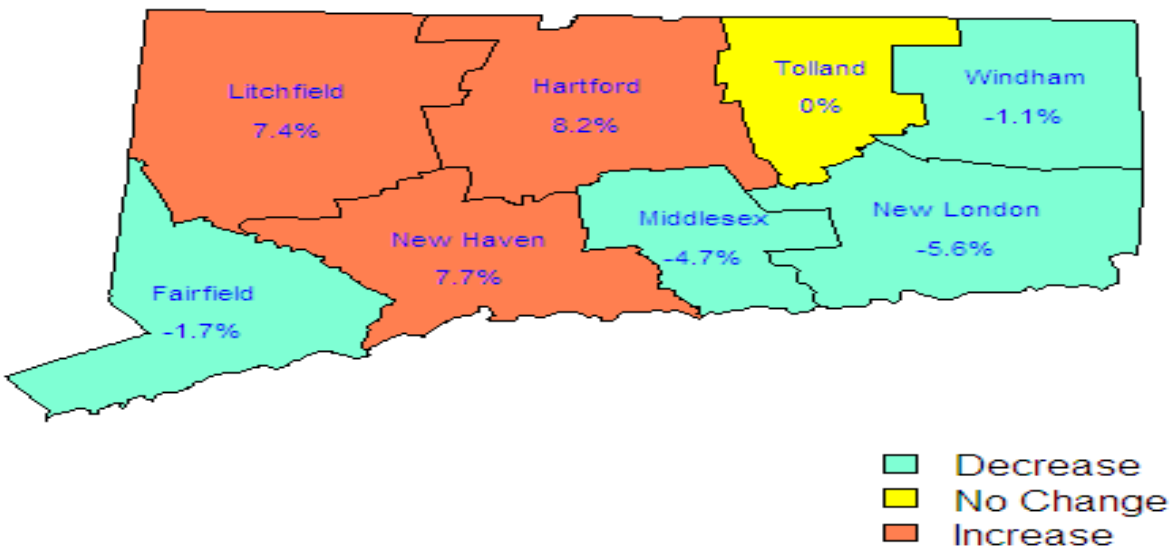
Kevin Burke, Ph.D., ARe, CPCU, AU, FCAS, MAAA
Actuary
Alfa Mutual Insurance Company
Montgomery, AL

Comments on R

- There’s a learning curve associated and techniques are ad hoc.
- R gives the user much more control over a process than Excel.
- R is case sensitive.
- Writing efficient code involves a lot of trial and error.

Finished Product

Connecticut Rate Change by County



Statewide Change is 1.275%

Map Project Outline

- Generate random rate change for eight counties
- Create a map with different colors depending upon positive, negative, or neutral rate change
- Label each county with county name and rate change
- Create a legend

Creating a Map of Rate Changes

Install and load the maptools package.

```
>library(maptools)
```

Go to the US census web site and download the Connecticut county shapefile.

Main site:
http://www.census.gov/geo/www/cob/bdy_files.html

Shapefiles

Shapefiles have three mandatory components

- *.shp contains the geometric characteristics
- *.shx is a conditional index of the geometry
- *.dbx contains attributes for each shape

Further details are beyond the scope of the presentation (as well as my expertise.)

Maps continued

Select the shapefile co09_d00.shp

```
>f1<-choose.files()
>f1
"R:\\Kevin\\R\\Connecticut\\co09_d00.shp"
```

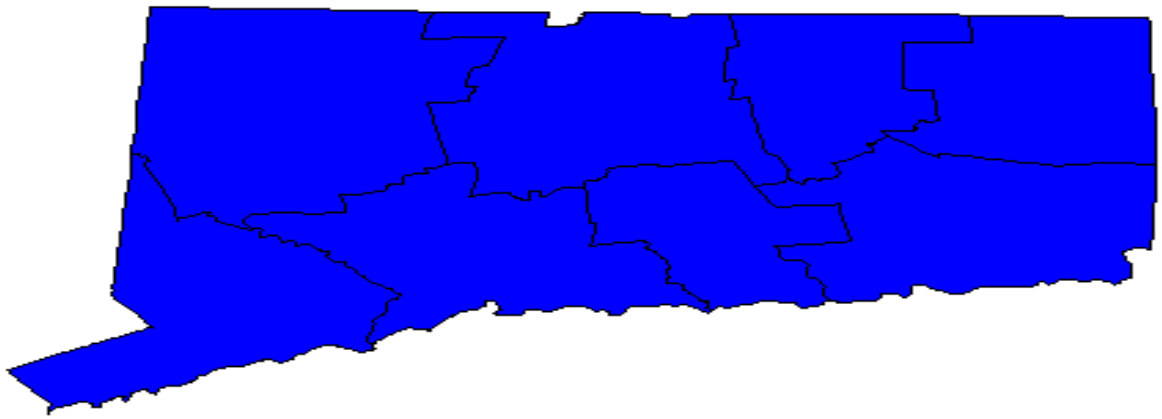
Read it as a polygon

```
>CT.county<-readShapePoly(f1)
>plot(CT.county,col="blue")
```

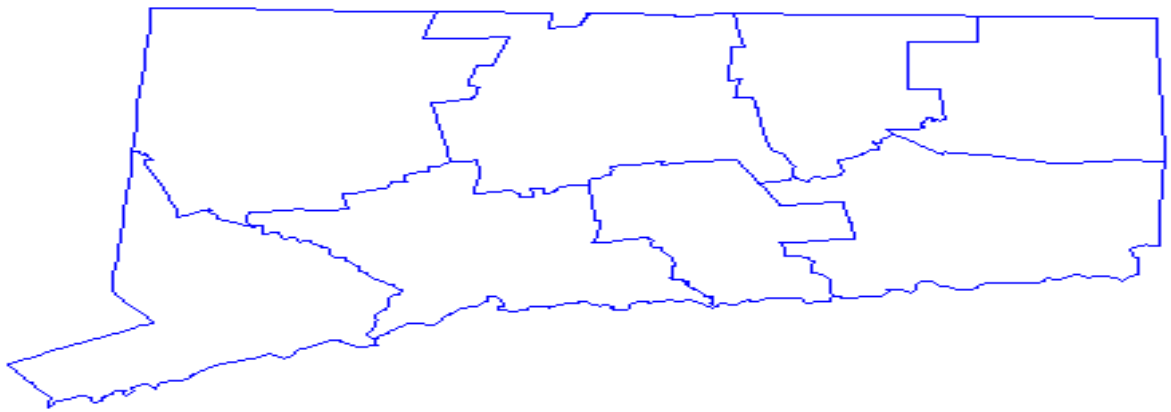
You can also read it as a line file to get boundaries.

```
>CT.boundary<-readShapeLines(f1)
>plot(CT.boundary,col="blue")
```

Polygon Shapefile



Line Shapefile



Generate Rate Change Data

```
>set.seed(123)

> ratechange<-runif(8,min=-.10,max=.10)
```

runif stands for Random Uniform

```
>runif

[1] -0.042484496  0.057661027 -0.018204616  0.076603481
    0.088093457
[6] -0.090888700  0.005621098  0.078483809
```

We've sampled 8 times from a uniform distribution on the interval [-0.10, 0.10]

Generate Rate Change Data

Multiply the list by 100 and round it to 1 decimal place.

You don’t have to loop through the entire list!

```
>ratechange<-round(100*ratechange,1)

>ratechange

[1] -4.2  5.8 -1.8  7.7  8.8 -9.1  0.6  7.8
```

Force one of the changes to zero

```
>ratechange[3]<-0

>ratechange

[1] -4.2  5.8  0.0  7.7  8.8 -9.1  0.6  7.8
```

Write a simple function

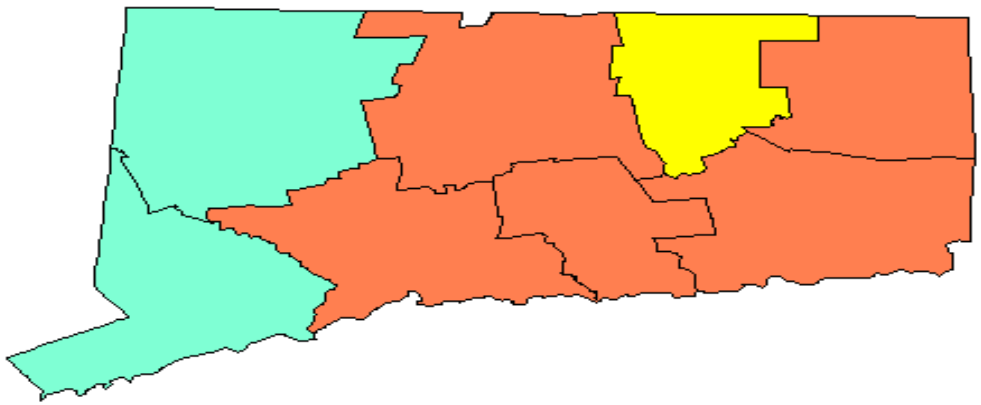
```
ChangeColor<-function(x){
    c1<-"yellow"
    if (x<0) c1<-“aquamarine"
    if (x>0) c1<-“coral"
    return(c1)}
```

Apply the Function to the list

```
ratechangecolors<-
    sapply(ratechange,ChangeColor)

>ratechangecolors

[1] "aquamarine" "coral"      "yellow"     "coral"      "coral"
[6] "aquamarine" "coral"      "coral"

>plot(CT.county,col=ratechangecolors)
>title("Connecticut Rate Change by County")
```

Halfway There

Connecticut Rate Change by County



Find the coordinates of each county

The maptools ‘coordinates’ function gives a set of coordinates you can use to plot.

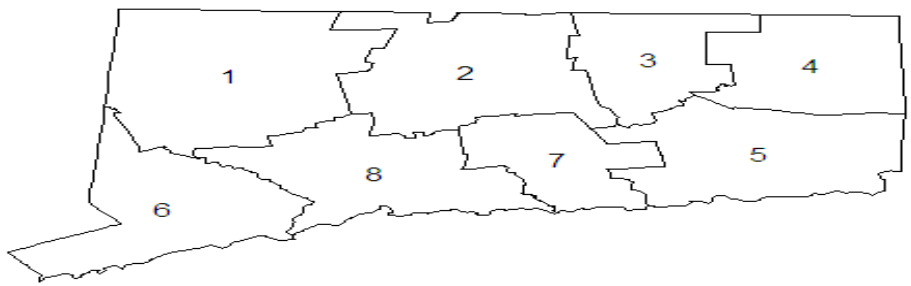
```
>CT.coords<-coordinates(CT.county)
>CT.coords
```

```
          [,1]     [,2]
[1,] -73.24531 41.79250
[2,] -72.73286 41.80640
[3,] -72.33650 41.85505
[4,] -71.98752 41.82995
[5,] -72.10211 41.48707
[6,] -73.38921 41.27188
[7,] -72.53491 41.46313
[8,] -72.93230 41.41054
```

Order is important

We can look at the order that the counties are plotted.

```
>plot(CT.county)
>text(CT.coords,labels=c(1,2,3,4,5,6,7,8))
```



Structure of data

```
>summary(CT.county)

Object of class SpatialPolygonsDataFrame

Coordinates:
        min       max
r1 -73.72777 -71.78699
r2  40.98517  42.05059
Is projected: NA
proj4string : [NA]
```

Structure of Data

Data attributes:

AREA		PERIMETER		CO09_D00_		CO09_D00_I		STATE
Min.	:0.1077	Min.	:1.766	Min.	:2.00	Min.	:1.00	09:8
1st Qu.	:0.1391	1st Qu.	:1.816	1st Qu.	:3.75	1st Qu.	:2.75	
Median	:0.1771	Median	:2.378	Median	:5.50	Median	:4.50	
Mean	:0.1744	Mean	:2.237	Mean	:5.50	Mean	:4.50	
3rd Qu.	:0.1984	3rd Qu.	:2.524	3rd Qu.	:7.25	3rd Qu.	:6.25	
Max.	:0.2650	Max.	:2.664	Max.	:9.00	Max.	:8.00	

COUNTY		NAME		LSAD	LSAD_TRANS
001	:1	Fairfield	:1	06:8	County:8
003	:1	Hartford	:1		
005	:1	Litchfield	:1		
007	:1	Middlesex	:1		
009	:1	New Haven	:1		
011	:1	New London	:1		
(Other)	:2	(Other)	:2		

Access the data

>CT.county@data

(Output too busy for presentation)

Name is one of the fields.

>CT.county@data$NAME

Litchfield Hartford Tolland Windham New London Fairfield Middlesex
[8] New Haven
8 Levels: Fairfield Hartford Litchfield Middlesex New Haven ... Windham

Working With Real Data

Add a column, call it “Change”, and set the values equal to zero.

```
>CT.county@data$Change<-0
```

Manually edit the data using the data editor.

```
>CT.county@data<-edit(CT.county@data)
```

Working With Real Data

Make the rownames equal to the county names

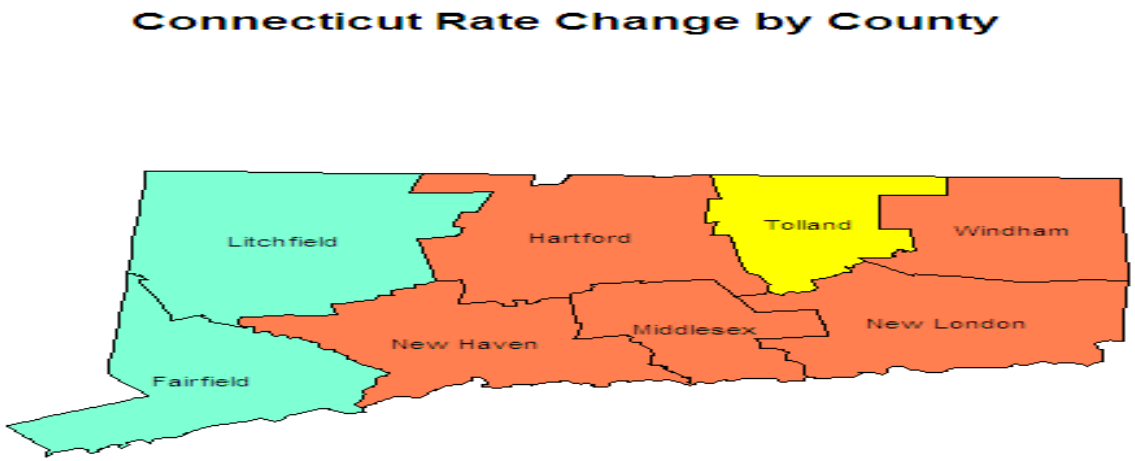
>rownames(CT.county@data)<-CT.county@data$NAME

Create a data set with changes.

>plot(CT.county,col=ratechangecolors)

>title("Connecticut Rate Change by County")

>text(CT.coords,labels=CT.county@data$NAME,cex=.7)



```
>ratechangelabels<-paste(ratechange,"%",sep="")
>ratechangelabels
[1] "-4.2%" "5.8%"  "0%"    "7.7%"  "8.8%"  "-9.1%" "0.6%"
    "7.8%"
```

Move the coordinates up and down so that the county name and rate change label aren't on top of each other.

```
>name.coords<-CT.coords+t(matrix(c(0,.05),2,8))
>change.coords<-CT.coords+t(matrix(c(0,-.05),2,8))
```

Compute a statewide average

```
>SW.text<-paste("Statewide Change is
    ",mean(ratechange),"%",sep="")
>SW.text
```

"Statewide Change is 2.175%"

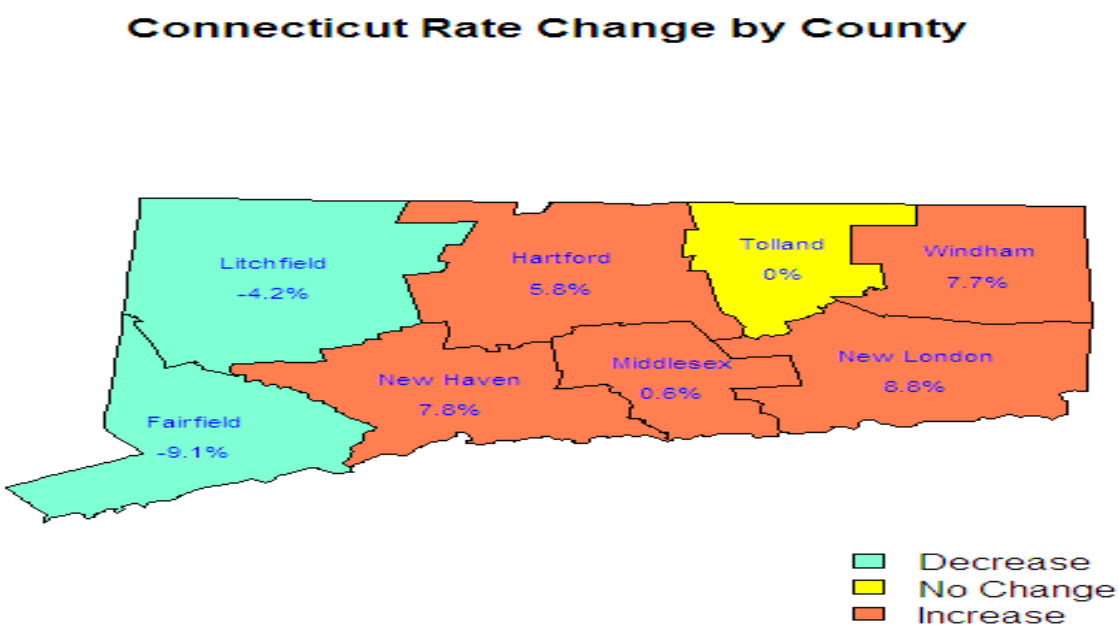
Create Text and Colors for legend

```
>legend.text<-c("Decrease","No Change","Increase")
>legend.color<-c("aquamarine","yellow","coral")
```

Now plot the graph

```
>plot(CT.county,col=ratechangecolors)
>title(main="Connecticut Rate Change by County",sub=SW.text)
>text(name.coords,labels=CT.county@data$NAME,cex=.7,col="blue")
>text(change.coords,labels=ratechangelabels,cex=.7,col="blue")
>legend(x="bottomright",legend=legend.text,fill=legend.color,bty="n")
```

Finished Product



Statewide Change is 2.175%

Connecticut Shapefile

County and County Equivalents

http://www.census.gov/geo/www/cob/co2000.html

Connecticut (unzip to folder)

http://www.census.gov/geo/cob/bdy/co/co00shp/co10_d00_shp.zip

Mapping Resources

National Weather Service shapefiles

http://www.nws.noaa.gov/geodata/

R-sig-Geo -- R Special Interest Group on using Geographical data and Mapping

https://stat.ethz.ch/mailman/listinfo/r-sig-geo

Maps in R

http://geography.uoregon.edu/GeogR/topics/maps.htm