Geog 205 Exercise 6

Descriptive Spatial Statistics

Descriptive Spatial Statistics the Old Fashioned Way

1. For the most part, if you are looking to generate descriptive spatial statistics, such as the mean center or the standard distance, you will be using point data for which you have some form of XY coordinates and perhaps some additional attribute data. (If you had areal data, you could work with area centroids, but that often does not make that much sense.)

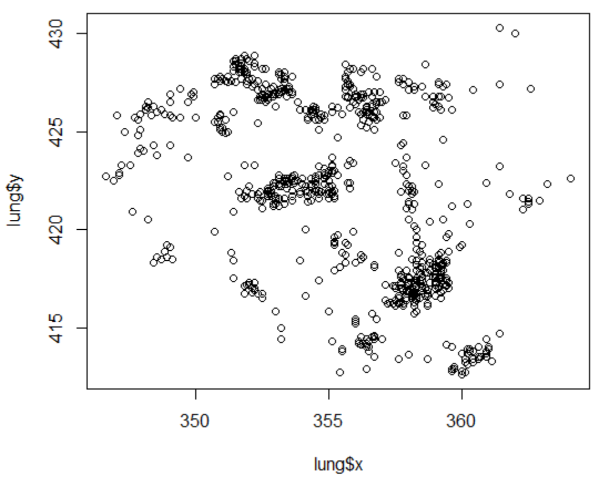
2. There is a data file under the Module 2 datasets folder on the class webpage called chorley_lung.csv. Chorley is a town in NW England where a famous point pattern dataset was developed on patients with lung and larynx cancer. The observations in chorley_lung.csv refer to patients with lung cancer. There are three variables, the first two being the X and Y coordinates of the point pattern.

3. Set your working directory and read the chorley_lung.csv data into R and plot the XY variables to get a sense of what the point pattern looks like

```
> lung <- read.csv("chorley_lung.csv", sep=",", header=T)
```

```
> plot(lung$x,lung$y)
```

And here is the output



4. Let's generate the mean center and the median center for these data and then add those pieces of information to the plot. The median center is going to be the UK version (equal number of points north, south, east and west of the median center).

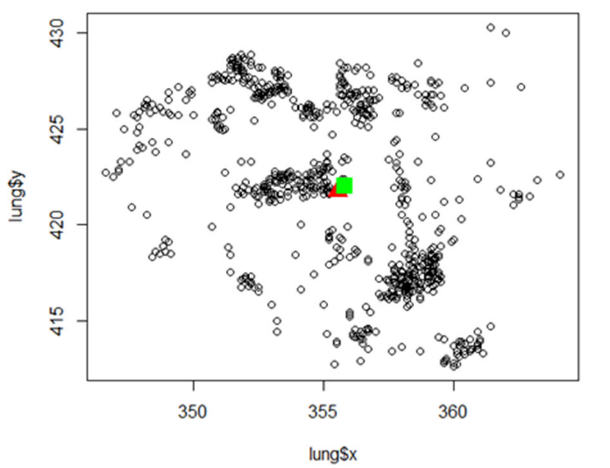
```
> meanx <- mean(lung$x)
```

```
> meany <- mean(lung$y)
```

```
> medx <- median(lung$x)
> medy <- median(lung$y)
```

5. Add points to the plot with the points() function

```
> plot(lung$x,lung$y)
> points(meanx, meany, col="red", cex = 2.0, pch=17)
> points(medx, medy, col="green", cex = 2.0, pch=15)
```



Here you can see that:

col determines the color of the point you add

cex sets the size of the point (try some different sizes)

pch determines the shape of the point that you add (you might try a few other numbers in the range 0-25 for basic symbols….you can Google search for a complete list in R)

The mean center and the median center are almost in the same location. What does that tell you?

6. Now let’s find and plot the standard distance for the lung cancer observations

```
> std_dist <- sqrt(sum((lung$x-meanx)^2 + (lung$y-meany)^2)/length(lung$x))
> std_dist
```

[1] 5.705975

To plot the circle involves a little geometry:

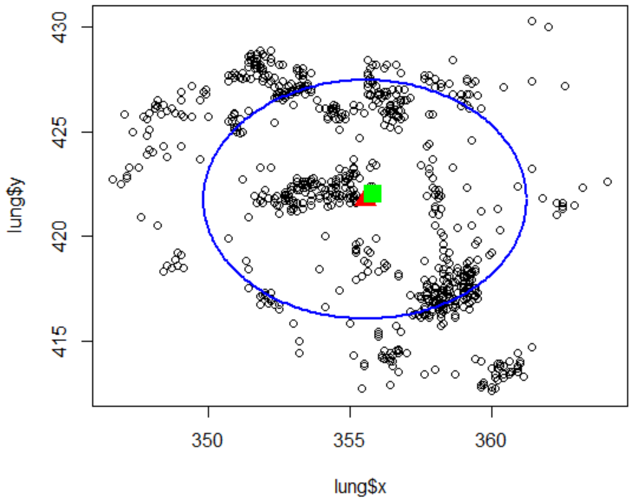
```
> bearing <- 1:360*pi/180
> cx <- meanx +std_dist * cos(bearing)
> cy <- meany + std_dist * sin(bearing)
> circle <- cbind(cx, cy)
```

And then add a line to the plot, where the form of the line is a circle

```
> lines(circle, col="blue", lwd=2.0)
```

Here lwd controls the line weight (try some different values) and lty controls the type of line, solid, dashed etc.

And the plot ...



7. It is easy to add a title with the command

> title(main="scatterplot of lung cancer data with geostats")

You can add text to the plot with the text command, specifying the starting coordinates & the text. You can clearly do a lot more customizing…..

> text(356,421, "geostats are here")

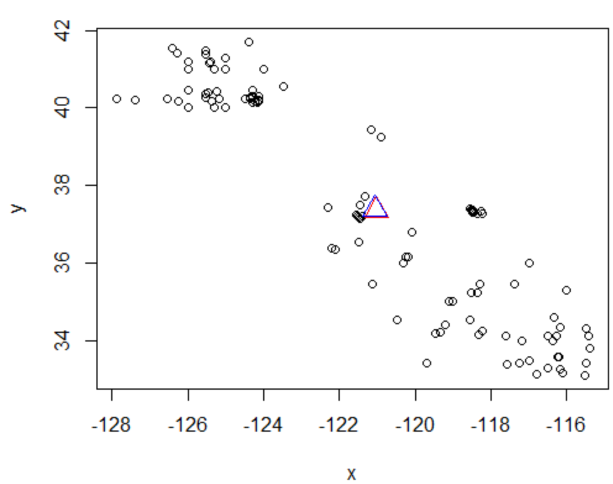
8. What about the weighted mean center? The data file caquakes.csv records the coordinates and the magnitude of most of the largest CA earthquakes from around 1870 to 1990. Read in the data, plot the point pattern and report the location of the mean center and the weighted mean center.

> quakes <- read.csv("caquakes.csv", sep=",", header=T)

```
> str(quakes)
'data.frame':	111 obs. of  3 variables:
 $ longitude: num  -120 -122 -121 -122 -124 ...
 $ latitude : num  36 37.2 37.2 37.4 41 ...
 $ magnitude: num  6.4 5.5 5.5 8.3 6.4 5.3 6 5.8 5.8 6 ...
```

Notice that the coordinates are latitude and longitude. If we had a large geographic distribution (global earthquakes), the positive and negative values of latitude and longitude would cause lots of problems and mapping the points in 2D would be tricky. Let’s assume that we overcome those concerns by focusing on California.

```
> x <- quakes$longitude
> y <- quakes$latitude
> mag <- quakes$magnitude
> meanx <- mean(x)
> meany <- mean(y)
> sumw <- sum(mag)
> meanxw<-sum(mag*x)/sumw
> meanyw <- sum(mag*y)/sumw
> meanxw
[1] -121.0685
> meanyw
[1] 37.37709
> meanx
[1] -121.0186
> meany
[1] 37.35225
> plot(x,y)
> points(meanx, meany, col="red", cex = 2.0, pch=2)
> points(meanxw, meanyw, col="green", cex = 2.0, pch=2)
```



So the two means (weighted and unweighted) are almost on top of one another. You might be tempted to interpret the plot as implying that larger earthquakes are equally likely to occur as earthquakes in general across CA. That interpretation might be problematic because these data only cover larger earthquakes...

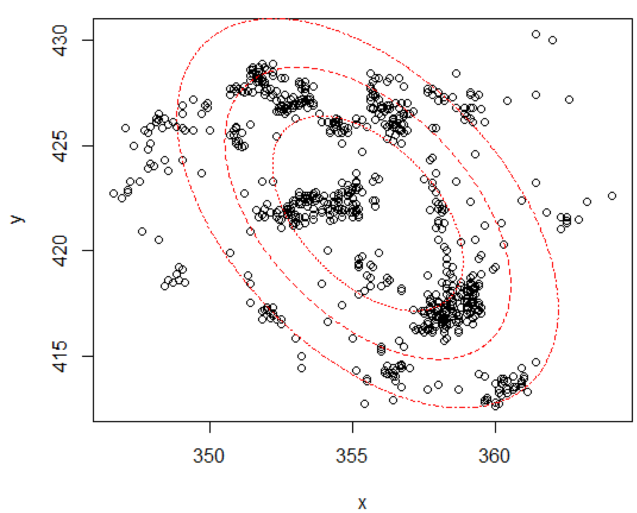
9. There are not many tools for estimating and plotting the standard deviation ellipse in R. An archived package called aspace was good, but not easy to find and install. A simple alternative is the package “phonTools”.

```
>install.packages(“phonTools”)
> library(phonTools)
```

The file input to the function for the ellipse must contain only coordinates and so

```
> x <- lung$x
> y <- lung$y
> coords <- cbind(x,y)
> plot(lung$x,lung$y)
> sdellipse (coords, stdev=1.0, col=”red”, lty=3)
> sdellipse(coords, stdev=1.5, col="red", lty=2)
> sdellipse(coords, stdev=2, col="red", lty=4)
```

From which you get



The long axis of the ellipses shows the angle of most variation in the point data. The ellipses might be read as confidence intervals for the point pattern associated with different probabilities.