Reading Shapefiles into Spatstat for Point Pattern Analysis

We can read point and polygon data into Spatstat as follows.

1. Load the sf and spatstat libraries

```
> library(sf) >
library(spatstat)
```

2.We need to get some point location data and a polygon shape file. A point file showing Walmart store locations and Massachusetts polygon map data are in the file Walmarts.zip. Copy that zip file over to your own working directory and unzip the contents.

3. After setting your working directory, read the points and polygon files into objects that spatstat can work with (there may be different ways of doing this):

#Read in the base map data

```
> map <- st_read("MA.shp")
```

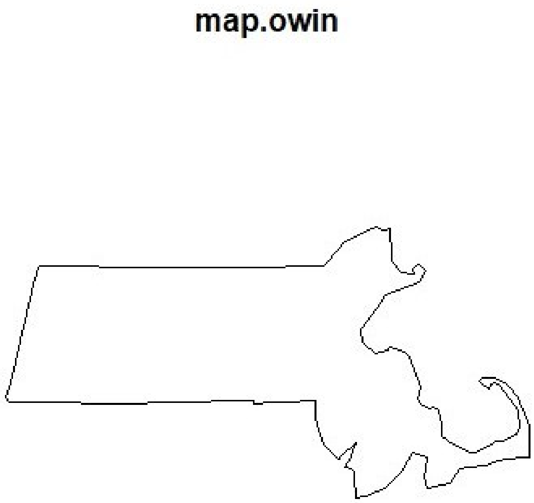
Hint: You may need the command `> sf_use_s2(FALSE)`

4. Now coerce the polygon map data into a plot window (an owin object) for spatstat (this is using the sf library in place of the maptools package that is no longer supported in R)

```
> map.owin <-as.owin(map)
```

And you can plot this polygon map file in spatstat as

```
> plot(map.owin)
```



5. Now let’s read in the point data and coerce into a spatstat points object (ppp)

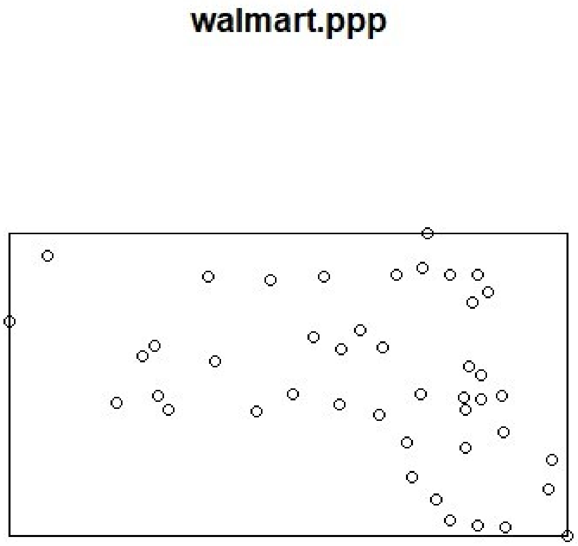
```
> walmart <- st_read("Walmarts.shp")
> walmart.ppp <- as.ppp(walmart)
```

You will get a warning message noting that your ppp object is marked. This means that it contains attribute data along with point locations. This can be useful, but for now we want to remove the marks (attribute information)

```
> walmart.ppp <- unmark(walmart.ppp)
```

6. You can plot the point file using a regular bounding box in spatstat

```
> plot(walmart.ppp)
```



And you can overlay quadrats and perform a quadrat test on these spatial point data.

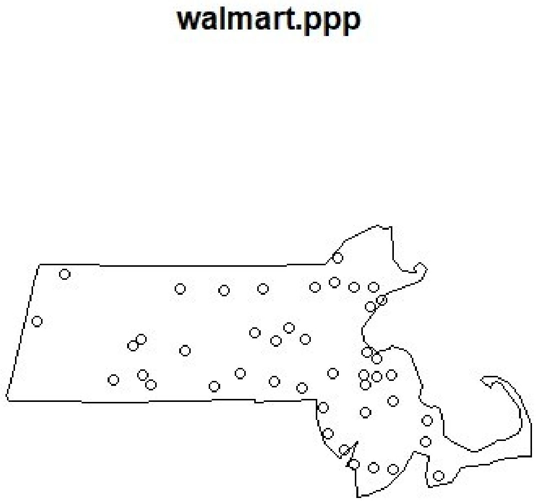
7. When R reads a point layer, it uses the points' extent to define a rectangular window. Since different spatial statistics can be sensitive to the study area's boundaries, we need to ensure that the proper boundary is defined. In this example, we are restricting our study region to the State of Massachusetts. This region can be defined in R by assigning the *map.owin* object (which is the polygon representation of the state of Massachusetts) to the *walmart.ppp* object's as the mapping window environment.

```
> walmart.ppp$window<- map.owin
```

To check that the plot window is properly aligned to the point object, plot walmart.ppp

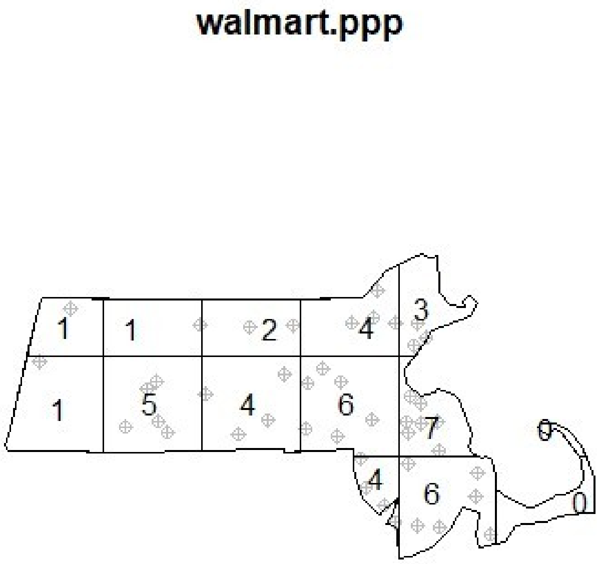
```
> plot(walmart.ppp)
```

You should see the following plot (or something like it)



8. Now you have point and polygon data into spatstat format, you can use many of the spatstat functions discussed in exercises 8-10. For example:

```
> qcount <- quadratcount(walmart.ppp, nx=6, ny=3)
> plot(walmart.ppp, pch=10, cols="grey")
> plot(qcount, add=TRUE)
```

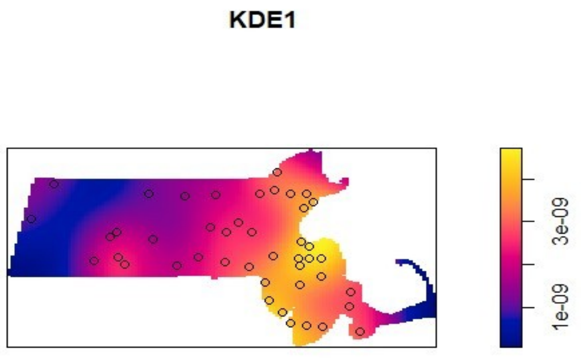


How about a quadrat test?

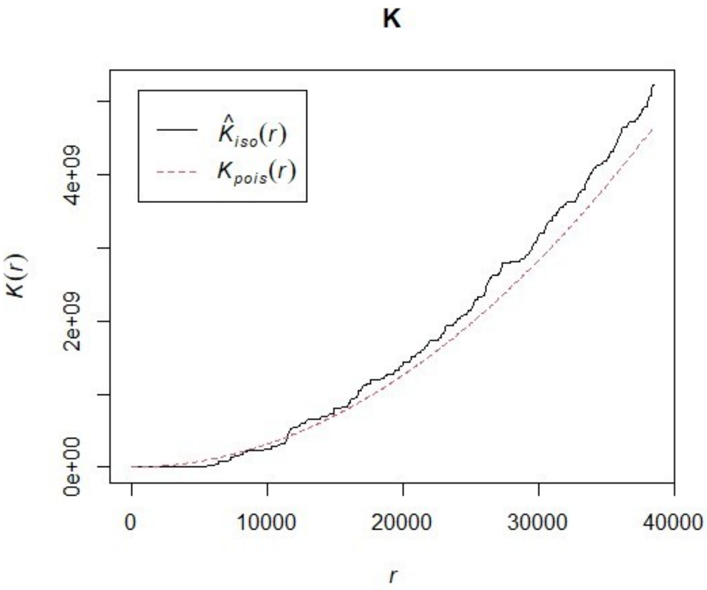
```
> plot(walmart.ppp, cex=0.5, pch="+")
> walmartqtest <-quadrat.test(walmart.ppp, nx=4, ny=3)
> walmartqtest

        Chi-squared test of CSR using quadrat counts

data:  walmart.ppp
X2 = 14.192, df = 9, p-value = 0.2313 alternative
hypothesis: two.sided

Quadrats: 10 tiles (irregular windows)
```

10. And KDE

```
> KDE1 <- density(walmart.ppp)
> plot(KDE1)
> plot(walmart.ppp, add=TRUE)
```



```
> K <- Kest(walmart.ppp, correction="Ripley")
> plot(K)
```



Now you can conquer Question 2 on the assignment!