# Chapter 3: Using statistical measures to analyze data distributions

This chapter provides an overview of different statistical and spatial statistical methods commonly used in summarizing data. Understanding the statistical distribution of a dataset helps a data scientist gain fundamental knowledge to move the analysis forward. The chapter will illustrate basic statistical methods using a number of datasets with non-spatial or spatial characteristics. Continued use and R practice, a statistical software package will deepen our knowledge and understanding of the basic statistical measures.

In this exercise, we will accomplish three specific tasks. For Task 1, we will generate descriptive statistics; for Task 2, we will generate spatial statistics, and for Task 3, we will explore the different probability distributions. At the end of this lesson, learners will be able to generate and interpret descriptive statistics and spatial statistics, as well as apply probability distribution concepts with spatial datasets. The datasets for use in this chapter are located in .. /ChapterData/Chapter3\_Data\_Folder.

## Task I: Generating descriptive statistics

Here we illustrate how to generate descriptive statistics using the eleven sampled tree heights nearby a residential area in a Chicago neighborhood (in Meters) dataset (See Table 3.1). Two of the most commonly used sets of descriptive measures are the measures of center and the measures of dispersion. These measures are illustrated below.

### Step I: Create a vector of the tree height data

As with any project, the starting point is to load the data we’ll be using. For this, let’s create a vector (to be assigned a variable, x) of the eleven sampled tree heights nearby a residential area in a Chicago neighborhood (in meters). A vector is a sequence of values belonging to the same mode or class (Crawley, 2007; Ligges, 2008). A further attribute of the vector is its length (its only dimension). Modes for data are logical, numeric, factor, complex and character.

#Let’s create a vector of the sample tree heights

x <- c(16, 18, 21, 32, 34, 37, 43, 43, 45, 60, 72);x

## [1] 16 18 21 32 34 37 43 43 45 60 72

### Step II: Generating descriptive statistics from the data

There are two commonly used sets of descriptive measures; these are the measures of center and the measures of dispersion. The measures of center include; mode, median, mean whereas the measures of dispersion include; range, standard deviation, variance. A measure of dispersion is a descriptive statistic that quantifies the spread of a set of observations. These two measures are illustrated below.

#Measures of central tendency

temp <- table(as.vector(x));temp

## 16 18 21 32 34 37 43 45 60 72

## 1 1 1 1 1 1 2 1 1 1

*#This returns the names of the values that have the highest count in temp's second row. And since the mode is the value(s) that occur most frequently in a vector, this line returns the mode.*

names(temp)[temp == max(temp)]

#Your mode should look like this

## [1] "43"

median(x, na.rm = TRUE)

#Your median should look like this

## [1] 37

mean(x, na.rm = TRUE)

#Your mean should look like this

## [1] 38.27273

#Measures of dispersion

RANGE <- max(x)- min(x); RANGE *#Returns the range of x*

# Your range should look like this

## [1] 56

sd(x) *#Returns the standard deviation of x*

# Your standard deviation should look like this

## [1] 17.2168

var(x) *#Returns the variance of x*

*# Your variance should look like this*

## [1] 296.4182

summary(x) *#Returns the summary statistics of your data*

# Your summary statistics should look this

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 16.00 26.50 37.00 38.27 44.00 72.00

*#Now let’s visualize the distribution of the data using a histogram*

hist(x, col = "gray", xlab = "Tree heights (in meters)", border = "black")

#Your histogram should like this;

## 

## References

Crawley, M.J. (2007). The R book. John Wiley, Chichester.

Ligges, U. (2008). Programmieren mit R. Springer, Berlin.

## Task 2: Generating spatial statistics

Task 2 will teach students how to generate spatial descriptive statistics from a given spatial dataset within the R environment. In a bid to understand the basic spatial characteristics, we apply spatial descriptive statistics. There are two common types of measures that can be undertaken: (1) one that measures centrality (spatial measures of central tendency) and (2) the other measures dispersions (spatial measures of dispersion) of events over space. These measures provide useful summaries of a spatial distribution.

We will now illustrate spatial descriptive measures using an environmental quality dataset downloaded from the Texas Commission on Environmental Quality website (<http://www.tceq.state.tx.us/>). This example demonstrates on how both types of measures can be generated.

### Step I: Load the data of interest

Setup the environment, define working directory, load the shapefile of interest, and visualize it.

*# Set your workspace directory:*

setwd("../ChapterData/Chapter3\_Data\_Folder")

*# select the Texas air monitoring sites data (*TCEQ\_Air\_Monitoring\_Sites\_PR\_studyregion.shp) *from the workspace directory (…*/K24901\_Data\_Folders/Chapter3\_Data\_Folder*)*

**library**(rgdal)

##Loading required package: sp

##rgdal: version: 1.2-16, (SVN revision 701)

##Geospatial Data Abstraction Library extensions to R successfully loaded

##Loaded GDAL runtime: GDAL 2.2.0, released 2017/04/28

##Path to GDAL shared files: C:/Users/user/Documents/R/win-library/3.4/rgdal/gdal

##GDAL binary built with GEOS: TRUE

##Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ\_VERSION: 493]

##Path to PROJ.4 shared files: C:/Users/user/Documents/R/win-library/3.4/rgdal/proj

##Linking to sp version: 1.2-7

**library**(sp)

*# Read in the Texas dataset*

TCEQ <- readOGR('TCEQ\_Air\_Monitoring\_Sites\_PR\_studyregion.shp', layer=' TCEQ\_Air\_Monitoring\_Sites\_PR\_studyregion')

##OGR data source with driver: ESRI Shapefile

##Source: "TCEQ\_Air\_Monitoring\_Sites\_PR\_studyregion.shp", layer:

##"TCEQ\_Air\_Monitoring\_Sites\_PR\_studyregion"

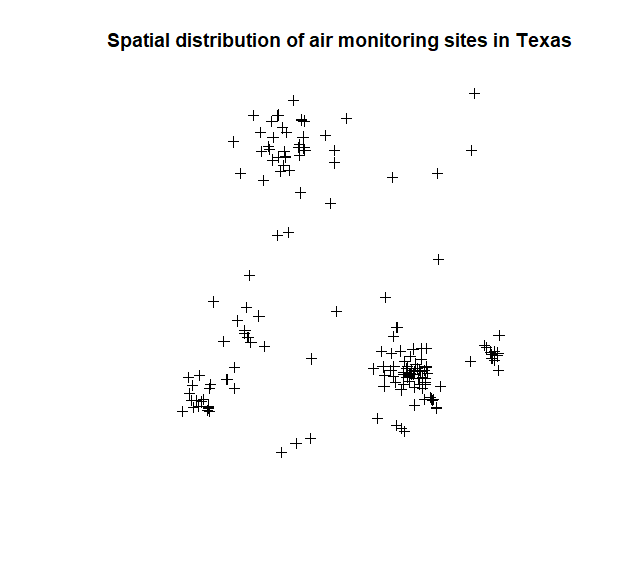
##with 175 features

##It has 37 fields

##Integer64 fields read as strings: OBJECTID SITE\_ID AQS\_SITE\_C ELEVATION ADDR\_ZIP\_C ##AIR\_TOXICS METEOROLOG NITROGEN PM\_25 OZONE CO\_SO2\_H2S PM10\_OTHER LEAD CHROMIUM\_V

plot(TCEQ, main = "Spatial distribution of air monitoring sites in Texas")

#Your outline map should like this;



**Step II: Generate spatial descriptive statistics using the Texas Air Monitoring sites dataset**

Here, we shall explore on how to perform the spatial measures of tendency and dispersion.

*# First let’s extract the coordinates of the data, which has already been read in as* TCEQ

TCEQ\_coords <- as.data.frame(coordinates(TCEQ))

View(TCEQ\_coords)

*#You should be able to view 175 pairs of coordinates*

#Now, compute for the spatial mean and median using the 175 coordinate pairs of longitudes and latitudes

#Spatial Mean

TCEQ\_X\_MEAN <- mean(TCEQ\_coords$coords.x1); TCEQ\_X\_MEAN

*#Your spatial mean of longitudes should be*

## [1] 772058.6

TCEQ\_Y\_MEAN <- mean(TCEQ\_coords$coords.x2); TCEQ\_Y\_MEAN

*#Your spatial mean of latitudes should be*

## [1] 3385088

#Spatial Median

TCEQ\_X\_MED <- median(TCEQ\_coords$coords.x1); TCEQ\_X\_MED

*#Your spatial median of longitudes should be*

## [1] 825458.1

TCEQ\_Y\_MED <- median(TCEQ\_coords$coords.x2); TCEQ\_Y\_MED

*#Your spatial median of latitudes should be*

## [1] 3312929

The dispersion of points in one dimension is easy to calculate and to visualize, but the spread of points in two (or more) dimensions is more complex. Instead of familiar error bars, the spread of a set of points on a Cartesian plane can be described using either the standard distance deviation (SDD) or the standard deviational ellipse (SDE).

For the purpose of geographic visualization, the SDD is typically portrayed as a circle with radius SDD centered on the mean center of a set of point observations. The orthogonal dispersion of a set of points can also be described using the standard deviation of the x- and y-coordinates of a set of point observations. The **aspace** R library provides functionalities for calculating SDD and SDE.

*#install.packages("aspace")*

**library**(aspace)

##Loading required package: splancs

##Spatial Point Pattern Analysis Code in S-Plus

## Version 2 - Spatial and Space-Time analysis

##Attaching package: ‘splancs’

##The following object is masked from ‘package:raster’:

## zoom

##Loading required package: Hmisc

##Loading required package: lattice

##Loading required package: survival

##Loading required package: Formula

##Loading required package: ggplot2

##Attaching package: ‘Hmisc’

##The following object is masked from ‘package:splancs’:

## zoom

##The following objects are masked from ‘package:raster’:

## mask, zoom

##The following objects are masked from ‘package:base’:

## format.pval, units

##Loading required package: shapefiles

##Loading required package: foreign

##Attaching package: ‘shapefiles’

##The following objects are masked from ‘package:foreign’:

## read.dbf, write.dbf

##Warning messages:

##1: package ‘aspace’ was built under R version 3.4.4

##2: package ‘splancs’ was built under R version 3.4.4

##3: package ‘Hmisc’ was built under R version 3.4.4

##4: package ‘Formula’ was built under R version 3.4.4

##5: package ‘ggplot2’ was built under R version 3.4.4

##6: package ‘shapefiles’ was built under R version 3.4.4

*## The Standard Distance is computed for using;*

calc\_sdd(id=1, filename="SDD\_TCEQ\_Output.txt", centre.xy=NULL, calccentre=TRUE,

weighted=FALSE, weights=NULL, points= TCEQ\_coords, verbose=FALSE)

#Your output should look like this; where SDD.radius is the standard distance

## $id

## [1] 1

## $calccentre

## [1] TRUE

## $weighted

## [1] FALSE

## $CENTRE.x

## [1] 772058.6

## $CENTRE.y

## [1] 3385088

## $SDD.radius

## [1] 203464

## $SDD.area

## [1] 130054403314

*## The Standard Distance Deviation is calculated using;*

calc\_sde(id=1,points= TCEQ\_coords)

#Your result should look like this;

## $id

## [1] 1

## $CALCCENTRE

## [1] TRUE

## $weighted

## [1] FALSE

## $CENTRE.x

## [1] 772058.6

## $CENTRE.y

## [1] 3385088

## $Sigma.x

## [1] 170645.5

## $Sigma.y

## [1] 231679.4

## $Major

## [1] "SigmaY"

## $Minor

## [1] "SigmaX"

## $Theta

## [1] 148.9453

## $Eccentricity

## [1] 0.6763734

## $Area.sde

## [1] 1.24203e+11

## $TanTheta

## [1] -0.6021611

## $SinTheta

## [1] 0.5158564

## $CosTheta

## [1] -0.8566751

## $SinThetaCosTheta

## [1] -0.4419213

## $Sin2Theta

## [1] 0.2661078

## $Cos2Theta

## [1] 0.7338922

## $ThetaCorr

## [1] 148.9453

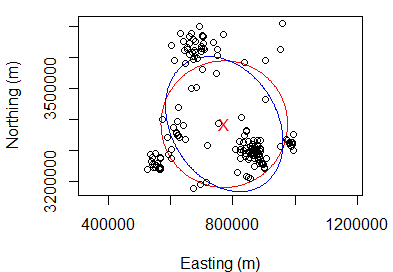
*# Plot the standard distance (red color)*

plot\_sdd(plotnew=FALSE, plotcentre=FALSE, centre.col="red", centre.pch="1", sdd.col="red",sdd.lwd=1,titletxt="", plotpoints=TRUE,points.col="black")

*# Plot the standard deviation ellipse (blue color) on the existing plot*

plot\_sde(plotnew=FALSE, plotcentre=FALSE, centre.col="red", centre.pch="1", sde.col="red",sde.lwd=1,titletxt="", plotpoints=TRUE,points.col="black")

## Your plot should look this this;



Do not forget to replicate this example using your own data for visualization of SDEs.

## References

Bui, R., Buliung, R.N. and Remmel, T.K., 2012. A collection of functions for estimating centrographic statistics and computational geometries for spatial point patterns

Toman, P., 2012. Calculating and visualizing standard deviations in two dimensions. <http://www.pamelatoman.net/blog/tag/standard-deviational-ellipses/>

**Task 3: Exploring the different probability distributions**

Along with the descriptive measures presented in the previous sections, it is equally important for spatial data scientists to be conversant with the probability distributions. Task 3 will therefore teach students how the different probability distributions can be computed for within the R environment with specific reference to examples in Oyana & Margai, (2016).

**Step I: Binomial Distribution**

A binomial distribution describes the sequence of a fixed number of events (x = 1,2, 3, n) in a sample space that can be isolated into two outcomes, where x represents the number of times each event occurs in the experiment, and these events are independent of each other. In R, this can be computed using the function dbinom(); which give the density, distribution function. So let's try this out;

#Experiment I: If a traveler from the city of St. Louis on the way to Chicago

#randomly stops at five convenience stores, find the probability that the traveler

#stops at exactly three stores. Each stop has six possible choices.

*#Using the density function; qbinom(): x = 3, size = 5, & prob = (1/6)*

p\_3 <- dbinom(3,5,(1/6));p\_3

## [1] 0.03215021

#Experiment II: In the last 3 years, the U.S. Transportation Security Administration #found that 40% of the passengers passing through Chicago’s O’Hare International Airp#ort had banned liquids exceeding 100 mL. If 10 passengers are selected randomly, fin#d the probability that at least 6 of them have banned liquids. To find this

#probability, we have to calculate individual probabilities for either 5, 6, 7, 8, 9, #or 10 then add them up to get the answer.

p\_5 <- dbinom(5,10,0.4)

p\_6 <- dbinom(6,10,0.4)

p\_7 <- dbinom(7,10,0.4)

p\_8 <- dbinom(8,10,0.4)

p\_9 <- dbinom(9,10,0.4)

p\_10 <- dbinom(10,10,0.4)

prob <- (p\_5+p\_6+p\_7+p\_8+p\_9+p\_10);prob

## [1] 0.3668967

**Step II: Poisson Distribution**

A Poisson distribution is normally used for an ordered or ranked series of spatial outcomes that are truly the result of random processes. It is used under the following circumstances: (1) when there is a specified interval for an event (equally segregated spatial areas or temporal sequences) and it is possible to count how many events have occurred; (2) when the events occur independently of each other both in space and time; (3) when each of the two outcomes in an event is virtually zero; (4) when there are low-occurrence events, rare events, isolated events, or a low-density pattern, and (5) when the average rate is known for a specified number of occurrences for an event.

The Poisson distribution can be computed using the function dpois(), which gives the density distribution, So let's try it out.

#1. On average, electrical storms occur about 31 days per year in New York. Suppose w#e observe 28 days a year. What will be the probability that we observe electrical st#orms?

*#Using dpois(); x=28, & mean =31 days*

dpois(28,31)

## [1] 0.06469306

#2. On average, electrical storms occur on about 21 days per year in both Paris and R#ome. Suppose we observe 14 days a year. What will be

#the probability that we observe electrical storms?

dpois(14,21)

## [1] 0.02821484

#3. On average, electrical storms occur on about 16 days per year in London. Suppose #we observe 14 days a year. What will be the probability that we observe electrical s#torms?

dpois(14,16)

## [1] 0.09301644

#4. On average, the residents of Kampala, Uganda, hear thunderstorms 240 days per yea#r, one of the highest rates in the world. Suppose we observe 200, 220, or 250 thunder#storm occurrences per year, what will be their probabilities?

dpois(200,240); dpois(220,240); dpois(250,240)

## [1] 0.0008216447

## [1] 0.01140586

## [1] 0.02053754

#5. On average, lightning kills about 100 Americans and inflicts another 500 injuries #per year. Suppose we observe 84 and 95 death occurrences, and/or 486 and 490 injury occurrences. What will be probabilities for these occurrences?

dpois(84,100); dpois(95,100) #Deaths

## [1] 0.01122452

## [1] 0.03601243

dpois(486,500); dpois(490,500) #Injuries

## [1] 0.01484533

## [1] 0.01629357

**Step III: Normal Distribution**

Often times we use binomial and Poisson distributions to describe discrete random variables, but to adequately describe continuous probability of variables we use the normal distribution. The probability of a normal distribution is described using the mean and standard deviation. Within the R environment, this is computed using the function pnorm(); which give the distribution function. Now, let’s try this out.

#1. Suppose that the tree height in samples from Chicago neighborhoods has a mean of 38.3 m and a standard deviation of 17.2 m. What is the probability that trees in a ra#ndomly selected tree sample will be: (a) less than 51 m, (b) more than 51 m, and (c) #between 26 and 66 m?

*#using pnorm(); x= 51; mean=38.3; & standard deviation = 17.2*

pnorm(51,38.3,17.2,lower.tail = TRUE) *#p<51*

## [1] 0.7698558

pnorm(51,38.3,17.2,lower.tail = FALSE) *#p>51*

## [1] 0.2301442

pnorm(66,38.3,17.2,lower.tail = TRUE) - pnorm(26,38.3,17.2,lower.tail = TRUE) *#p(26 < x < 66)*

## [1] 0.7090832

#2. Suppose that the blood lead levels among children from New York City have a mean #of 8.3 μg of lead per deciliter of blood(μg/dL) and a standard deviation of 4.6 μg/dL#What is the probability that blood lead levels among children in a randomly selecte#d blood testing sample will be (a) less than 10 μg/dL, (b) more than 10 μg/dL, and (#c) between 5 and 20 μg/dL?

pnorm(10,8.3,4.6,lower.tail = TRUE) *#p<51*

## [1] 0.6441468

pnorm(10,8.3,4.6,lower.tail = FALSE) *#p>51*

## [1] 0.3558532

pnorm(20,8.3,4.6,lower.tail = TRUE) - pnorm(5,8.3,4.6,lower.tail = TRUE) *#p(5 < x < 20)*

## [1] 0.7579459

**Step IV: Using Z-Score to assess the relative position in data distributions**

In order to be successful in testing hypotheses and comparing different observations, we can derive a set of statistics called the Z-score. The Z-score is a standard normal transformation that offers a better metric for comparing such observations. The Z-score is derived using the mean and standard deviation of a given random variable. In spatial analysis, Z-scores can be used to describe how the distributions of observations fit within the standard normal distribution or compare different normal distributions with similar standard deviations.

To illustrate the applications, let us explore the normal probability distribution for the Illinois corn and soybean production data using Chapter3\_Data\_folder (Data files: Illinios\_cnty\_agricultural\_statistics or agricul\_ILL\_stats3). This data was introduced earlier in Chapter 2.

*#Load the necessary packages*

library(rgdal)

## Loading required package: sp

## rgdal: version: 1.2-16, (SVN revision 701)

## Geospatial Data Abstraction Library extensions to R successfully loaded

## Loaded GDAL runtime: GDAL 2.2.0, released 2017/04/28

## Path to GDAL shared files: C:/Users/user/Documents/R/win-library/3.4/rgdal/gdal

## GDAL binary built with GEOS: TRUE

## Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ\_VERSION: 493]

## Path to PROJ.4 shared files: C:/Users/user/Documents/R/win-library/3.4/rgdal/proj

## Linking to sp version: 1.2-7

library(dplyr)

## Attaching package: ‘dplyr’

## The following objects are masked from ‘package:stats’:

## filter, lag

## The following objects are masked from ‘package:base’:

## intersect, setdiff, setequal, union

## Warning message:

## package ‘dplyr’ was built under R version 3.4.4

*#set the working directory*

setwd("../ChapterData/Chapter3\_Data\_Folder")

*#load in the shape file of interest*

x <- readOGR('agricul\_ILL\_stats3.shp', layer='agricul\_ILL\_stats3')

## OGR data source with driver: ESRI Shapefile

## Source: "agricul\_ILL\_stats3.shp", layer: "agricul\_ILL\_stats3"

## with 102 features

## It has 66 fields

## Integer64 fields read as strings: NO\_FARMS07 AVG\_SIZE07 CROP\_ACR07

*#Extract the data on corn and soybean production and label them as n & m respectively*

n <- x@data$CornProduc *#corn production*

n

## [1] 15126400 12151200 17399100 15752000 64128000 59202500 44160000 23218000 27878400 8131500

## [11] 36727500 22451800 34760000 22300400 60027000 18208500 34713900 62125000 26232500 22833600

## [21] 33536800 69920000 33160000 28911400 31559000 32939400 39474000 24770200 49720000 42112000

## [31] 26591200 20083800 13229600 22212900 17285400 18330000 8116800 33539200 47564000 30139200

## [41] 22808400 28880000 23777000 45451500 17748000 8881500 20077000 32703300 31240600 32106500

## [51] 17601200 16480800 3236100 11050000 11556700 15184000 14586300 12136000 14128000 8347600

## [61] 15174500 7049000 8184400 8580000 8066000 12762900 12167500 11294200 4498200 7276500

## [71] 6884700 13497000 4755300 10320000 6816000 7919200 6555400 4870600 3877600 10475400

## [81] 6058800 1750000 331000 1026800 1955800 1086300 3685200 1152000 2622400 28315000

## [91] 15640800 16274000 13401700 1228200 42596000 27380000 560000 17427600 43848000 893800

## [101] 44435500 49595000

m <- x@data$SoyProduct *#soybean production*

m

## [1] 1569500 2054400 4429800 2335900 268600 5635000 7516600 3539100 4540800 1045000

## [11] 5428500 3264800 5808800 3192600 11773500 2514300 5952800 11397500 5296500 3572400

## [21] 5432400 12316500 5179200 4926600 6360000 5648400 10020500 3180100 11850000 5955000

## [31] 5415000 4037600 2281600 4169700 2865600 2430400 1187700 5217300 5951700 7160000

## [41] 5594700 5231200 3526000 7303400 3432300 1804800 5305000 7374500 6683600 5759200

## [51] 3807000 4600000 576000 3058000 2373500 6647000 4627600 4312000 5425200 4005000

## [61] 5940800 5067600 3751500 4780600 4752000 4995000 5280800 5887600 1717800 2692800

## [71] 3220000 7658700 3748000 5203800 3912000 4783500 3956000 3188000 3147600 2857700

## [81] 2392000 1068600 72700 624200 896800 620100 1396000 1192000 1457400 3087000

## [91] 1747200 1821300 1587600 9326400 3271500 2082500 259600 1922800 4540800 62700

## [101] 3601800 3960600

*#Derive their Z-scores using formula (X-mean)/ (standard deviation).*

*#corn production*

n\_mean <- mean(n);n\_mean *#mean*

## [1] 20892069

n\_sd <- sd(n);n\_sd *#standard deviation*

## [1] 16153562

n\_zscore <- ((n-n\_mean)/n\_sd);n\_zscore *#Z-score*

## [1] -0.35692862 -0.54111091 -0.21623519 -0.31820032 2.67655712 2.37163985 1.44042109

## [8] 0.14398876 0.43249479 -0.78995386 0.98030585 0.09655650 0.85850609 0.08718395

## [15] 2.42268122 -0.16612860 0.85565223 2.55255970 0.33060395 0.12019215 0.78278285

## [22] 3.03511581 0.75945673 0.49644353 0.66034546 0.74580030 1.15033027 0.24007902

## [29] 1.78461762 1.31363791 0.35280958 -0.05003656 -0.47435164 0.08176719 -0.22327389

## [36] -0.15860704 -0.79086387 0.78293143 1.65114861 0.57245154 0.11863212 0.49449969

## [43] 0.17859413 1.52037250 -0.19463624 -0.74352447 -0.05045752 0.73118433 0.64063464

## [50] 0.69423892 -0.20372402 -0.27308334 -1.09300776 -0.60928163 -0.57791394 -0.35336285

## [57] -0.39036397 -0.54205188 -0.41873542 -0.77657600 -0.35395095 -0.85696694 -0.78667904

## [64] -0.76218908 -0.79400869 -0.50324310 -0.54010184 -0.59416422 -1.01487639 -0.84288336

## [71] -0.86713807 -0.45779802 -0.99896039 -0.65447290 -0.87139101 -0.80309647 -0.88752367

## [78] -0.99182265 -1.05329516 -0.64485273 -0.91826612 -1.18500605 -1.27285045 -1.22977636

## [85] -1.17226582 -1.22609296 -1.06520584 -1.22202574 -1.13099938 0.45952288 -0.32508425

## [92] -0.28588547 -0.46369764 -1.21730852 1.34360034 0.40164091 -1.25867401 -0.21447088

## [99] 1.42110646 -1.23800984 1.45747615 1.77687939

*#soybean production*

m\_mean <- mean(m);m\_mean *#mean*

## [1] 4193148

m\_sd <- sd(m);m\_sd *#standard deviation*

[1] 2562285

m\_zscore <- ((m-m\_mean)/m\_sd);m\_zscore *#zscore*

## [1] -1.023948693 -0.834703523 0.092359746 -0.724840632 -1.531659650 0.562721222 1.297065857

## [8] -0.255259709 0.135680459 -1.228648821 0.482129084 -0.362312607 0.630551312 -0.390490585

## [15] 2.958434733 -0.655215269 0.686751157 2.811690694 0.430612560 -0.242263495 0.483651163

## [22] 3.170354980 0.384833103 0.286249209 0.845671828 0.567950930 2.274279681 -0.395369044

## [29] 2.988290900 0.687609766 0.476860349 -0.060706775 -0.746032657 -0.009151223 -0.518111065

## [36] -0.687959484 -1.172956337 0.399702645 0.686321852 1.157893187 0.546993071 0.405127492

## [43] -0.260372334 1.213858865 -0.296941260 -0.932116586 0.433929912 1.241607538 0.971965368

## [50] 0.611193588 -0.150704582 0.158784840 -1.411688593 -0.443021828 -0.710166228 0.957681241

## [57] 0.169556477 0.046385151 0.480841171 -0.073429795 0.682067836 0.341278224 -0.172364938

## [64] 0.229268811 0.218106898 0.312944136 0.424485216 0.661305116 -0.966070659 -0.585550879

## [71] -0.379797003 1.352524176 -0.173730907 0.394433910 -0.109725528 0.230400614 -0.092553354

## [78] -0.392285858 -0.408053036 -0.521194251 -0.702946109 -1.219438291 -1.608114855 -1.392877256

## [85] -1.286487828 -1.394477390 -1.091661700 -1.171278147 -1.067698711 -0.431703804 -0.954596524

## [92] -0.925677021 -1.016884685 2.003388625 -0.359697753 -0.823736747 -1.535172140 -0.886063936

## [99] 0.135680459 -1.612017622 -0.230789360 -0.090758081

Now, let’s add a field of the Z-scores of both corn and soybean production to our shapefile. We can add columns to the data table based on other columns using data.table syntax (which preserves the current order of the rows) from the data.table R package:

library(data.table)

x@data <- copy(data.table(x@data))

*# let's add the Corn & Soybean Production z-scores using the data.table syntax*

x@data[, CornProd\_Zscore:=n\_zscore] #cornproduction

x@data[, SoybeanProd\_Zscore:=m\_zscore] #cornproduction

*#let’s check whether the corn and soybean production z-scores have been added to the #attribute table of x*

head(x@data)

ObjectID NAME STATE\_NAME STATE\_FIPS CNTY\_FIPS FIPS POP2000 POP00\_SQMI POP2010

1: 545 Jo Daviess Illinois 17 085 17085 22289 36.0 22046

2: 621 Rock Island Illinois 17 161 17161 149374 330.8 147257

3: 634 Will Illinois 17 197 17197 502266 591.4 712697

4: 636 Kendall Illinois 17 093 17093 54544 169.2 114126

5: 653 La Salle Illinois 17 099 17099 111509 97.1 113252

6: 666 Bureau Illinois 17 011 17011 35503 40.7 34812

POP10\_SQMI WHITE BLACK AMERI\_ES ASIAN HAWN\_PI OTHER MULT\_RACE HISPANIC MALES FEMALES

1: 35.6 21991 44 23 36 1 75 119 342 11175 11114

2: 326.2 127742 11260 410 1524 45 5612 2781 12791 72545 76829

3: 839.2 411027 52509 1038 11125 162 18219 8186 43768 250832 251434

4: 354.0 50658 718 105 480 12 1842 729 4086 27092 27452

5: 98.7 105896 1723 191 598 26 1908 1167 5791 55167 56342

6: 39.9 34365 116 61 182 10 455 314 1732 17244 18259

AGE\_UNDER5 AGE\_5\_17 AGE\_18\_21 AGE\_22\_29 AGE\_30\_39 AGE\_40\_49 AGE\_50\_64 AGE\_65\_UP MED\_AGE

1: 1246 3916 900 1724 2811 3378 4316 3998 41.6

2: 9486 26038 9357 14702 19963 22802 24462 22564 37.8

3: 42028 108683 24449 49264 88597 80019 67616 41610 33.3

4: 4362 11721 2501 5120 9432 8745 8028 4635 34.1

5: 7033 21019 5468 9793 16000 16942 16962 18292 38.1

6: 2094 6691 1626 2850 4731 5343 5869 6299 39.6

MED\_AGE\_M MED\_AGE\_F HOUSEHOLDS AVE\_HH\_SZ HSEHLD\_1\_M HSEHLD\_1\_F MARHH\_CHD MARHH\_NO\_C MHH\_CHILD

1: 40.2 43.1 9218 2.40 1118 1421 2013 3355 154

2: 36.4 39.2 60712 2.38 7629 10699 11786 18040 1249

3: 32.5 34.1 167542 2.94 13686 16198 59174 49456 3067

4: 33.4 34.8 18798 2.89 1346 1731 6674 6264 320

5: 36.8 39.4 43417 2.49 4994 6914 10438 13728 922

6: 37.9 41.1 14182 2.47 1512 2314 3379 4863 302

FHH\_CHILD FAMILIES AVE\_FAM\_SZ HSE\_UNITS VACANT OWNER\_OCC RENTER\_OCC NO\_FARMS07 AVG\_SIZE07

1: 354 6287 2.92 12003 2785 7129 2089 1016 277

2: 4540 39162 2.97 64489 3777 42303 18409 700 255

3: 9243 130972 3.36 175524 7982 139311 28231 877 252

4: 836 14969 3.27 19519 721 15810 2988 424 394

5: 2398 29840 3.04 46438 3021 32584 10833 1622 397

6: 678 9890 2.99 15331 1149 10775 3407 1189 402

CROP\_ACR07 AVG\_SALE07 SQMI CNTY\_FIP\_1 COUNTY\_NAM Group GroupArea CornAllPur CornAcre

1: 196027 134.50 618.7 085 Jo Daviess 10 Northwest 97000 92800

2: 148749 136.15 451.5 161 Rock Island 10 Northwest 67000 66400

3: 208874 145.49 849.3 197 Will 20 Northeast 99000 98300

4: 160527 244.16 322.4 093 Kendall 20 Northeast 90000 89500

5: 614381 202.83 1148.0 099 LaSalle 20 Northeast 336000 334000

6: 439887 255.14 873.3 011 Bureau 10 Northwest 299000 297500

CornYield CornProduc SoyAllPurp SoyAcre SoyYield SoyProduct WheatAllPu WheatAcre WheatYield

1: 163 15126400 37000 36500 43 1569500 0 0 0

2: 183 12151200 43000 42800 48 2054400 1300 1100 59

3: 177 17399100 97000 96300 46 4429800 13200 12600 69

4: 176 15752000 50000 49700 47 2335900 0 0 0

5: 192 64128000 8000 7900 34 268600 1600 1500 63

6: 199 59202500 116000 115000 49 5635000 4400 4100 74

WheatProdu CornProd\_Zscore SoybeanProd\_Zscore

1: 0 -0.3569286 -1.02394869

2: 64900 -0.5411109 -0.83470352

3: 869400 -0.2162352 0.09235975

4: 0 -0.3182003 -0.72484063

5: 94500 2.6765571 -1.53165965

6: 303400 2.3716399 0.56272122

Now let’s map the z-scores for both corn and soybean production using the standard deviation and natural breaks classification methods. For this, we shall use the tmap package with which the classification is selected by means of the style option in tm\_fill.

library(tmap)

#Warning message:

#package ‘tmap’ was built under R version 3.4.4

library(RColorBrewer)

*#Natural Breaks*

*#A natural breaks map is obtained by specifying the style = “jenks” in tm\_fill*

*#Soybean Production*

tm\_shape(x) +

tm\_fill("SoybeanProd\_Zscore",title="z-Score distribution for Soybean",n=5,style="jenks",palette = "PuOr",midpoint = 0) +

tm\_borders()+ tm\_layout()

*#Corn Production*

tm\_shape(x) +

tm\_fill("CornProd\_Zscore",title="z-Score distribution for Corn",n=5,style="jenks",palette = "PuOr",midpoint = 0) +

tm\_borders()+ tm\_layout()

*#Standard deviation*

*#A standard deviation map is included in tmap as style=“sd”*

*#Soybean Production*

tm\_shape(x) +

tm\_fill("SoybeanProd\_Zscore",title="z-Score distribution for Soybean",n=5,style="sd",palette = "PuOr",

breaks = c(-Inf, -1.5, -0.5, 0.5, 1.5, Inf)) +

tm\_borders()+ tm\_layout(title.size = 0.1)

*#Corn Production*

tm\_shape(x) +

tm\_fill("CornProd\_Zscore",title="z-Score distribution for Corn",n=5,style="sd",palette = "PuOr",

breaks = c(-Inf, -1.5, -0.5, 0.5, 1.5, Inf)) +

tm\_borders()+ tm\_layout(title.size = 0.1)