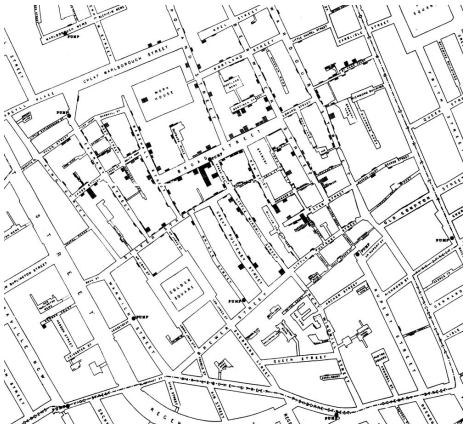
EPI 563: Spatial Epidemiology, Fall 2020

Michael Kramer

Last updated: 2020-10-04

Contents

How to use this eBook



Welcome to Concepts & Applications in Spatial Epidemiology (EPI 563)! This eBook is one of several sources of information and support for your progress through the semester. For an overview of the course, expectations, learning objectives, assignments, and grading, please review the full course syllabus on Canvas. This eBook serves to provide a 'jumping off point' for the content to be covered each week. Specifically, the content herein will introduce key themes, new vocabulary, and provide some additional detail that is complementary to the asynchronous (pre-recorded) video lectures, and foundational to the synchronous (in class) work.

Strategy for using this eBook

There is a separate module or chapter for each week's content. In general, the content within each week's section is divided into two sections focusing on **spatial thinking** and **spatial analysis**. This dichotomy does not always hold, but in broad terms you can expect these sections to be more specific to content in class on Tuesday versus Thursday respectively.

- Spatial thinking for epidemiology: This section introduces vocabulary, concepts, and themes that are important to the incorporation of spatialized or geo-referenced data into epidemiologic work. At a minimum, plan to read this content prior to class Tuesday, although you will likely benefit from reading both sections before Tuesday.
- Spatial analysis for epidemiology: This section is more focused on data management, visualization, spatial statistics, and interpretation. This content is relevant for our work together on Tuesday's, but is essential for successful work in the Thursday lab activities.

Please note that I will be continually updating the eBook throughout the semester, so if you choose to download, please double-check the **Last updated** date to be sure you have the most recent version.

Part I

Getting ready...

Software installation

The information in this module follow on the pre-class video on setting up R and RStudio on your computer.

Installing `R` on your computer

As of August 2020, the most up-to-date version of R is 4.0.2. The R Project for Statistical Computing are continually working to update and improve R, and as a result there are new versions 1-2 times per year.

If you already have R installed, you can open the console and check your current version by doing this: R.Version()$version.string

If you do not have R or have an older version than 4.0 you can install R by going to the R repository: https://www.r-project.org/. Note that there are many 'mirrors' or servers where the software is stored. Generally it is wise to select one that is geographically close to you, although any should work in theory. One mirror that is relatively close to Atlanta is here: http://archive.linux.duke.edu/cran/

Installing RStudio on your computer

The current version of RStudio 1.3.1056. If you do not have RStudio or have a version older than 1.2 please install/update.

TO INSTALL: go to https://www.rstudio.com/products/rstudio/download/

TO UPDATE: Open RStudio and go to Help Menu and choose 'Check for Updates'

Installing packages for this course

While base R has a great deal of essential functionality, most of the power of R comes from the rapidly growing list of user-created and contributed 'packages'. A package is simply a bundle of functions and tools, sometimes also including example datasets, basic documentation, and even tutorial 'vignettes'. You can see all the official R packages by going here: https://cran.r-project.org/web/packages/.

The most common way to install package in R is with the install.packages() command. For instance to install the package ggplot2 you do this:

```
install.packages("ggplot2")
```

Notice that for install.packages() you need quotes around the package name. Remember that you only need to install a package once (although you may have to update packages occasionally – see the green Update button in the Packages tab in R Studio). When you want to actually use a package (for example ggplot2) you call it like this:

```
library(ggplot2)
```

Notice that for the library() function you **do not** need quotes around the package name (unlike the install.packages() above). If your call to library() is working, nothing visible happens. However if you see errors, they might be because your package is out of date (and thus needs to be updated/reinstalled), or because some important dependencies are missing. Dependencies are other packages on which this package depends. Typically these are installed by default, but sometimes something is missing. If so, simply install the missing package and then try calling library(ggplot2) again.

While **most** packages can be installed as mentioned above (e.g. using install.packages()), there are instances where an installation requires additional tools, for instance to install from source or from github. Luckily there is a package for that! It is called Rtools, and you should install that **before** you install the packages below.

As you submit each installation request, note the output. If you get a warning that says installation was not possible because you are missing a package 'namespace', that suggests you are missing a dependency. Try installing the pacakge mentioned in the error. If you have trouble, reach out to the TA's!

Installing `Rtools40`

If your laptop uses a Windows operating system, you may need Rtools40 installed. This is a supplemental package that resides outside of R but is needed to install some packages from source code. However it appears that if you have a MacOS, these tools are already built-in. So you do not need Rtools40 for Mac.

If you are running Windows, navigate to this website: https://cran.r-project.org/bin/windows/Rtools/ and follow the instructions specific to your operating system.

Installing packages used for general data science

For the rest of this page, copy and paste the provided code in order to install packages necessary for this course. Notice if you hover to the right of a code-chunk in the html version of the eBook, you will see a copy icon for quick copying and pasting.

These packages will support some of our general work in R, including working with RMarkdown and R Notebooks, as well as data manipulation tools from the tidyverse. You can learn more about the tidyverse here: https://tidyverse.tidyverse.org/. The tidyverse is actually a collection of data science tools including the visualization/plotting package ggplot2 and the data manipulation package dplyr. For that reason, when you install .packages('tidyverse') below, you are actually installing multiple packages! The packages tinytex, rmarkdown, and knitr are all necessary for creating R Notebooks, which is the format by which many assignments will be submitted.

```
install.packages('tidyverse')
install.packages(c('tinytex', 'rmarkdown', 'knitr'))
tinytex::install_tinytex()
# this function installs the tinytex LaTex on your
#  computer which is necessary for rendering (creating)
    PDF's
```

Installing packages use for geographic data

There are many ways to get the data we want for spatial epidemiology into R. Because we often (but don't always) use census geographies as aggregating units, and census populations as denominators, the following packages will be useful. They are designed to quickly extract both geographic boundary files (e.g. 'shapefiles') as well as attribute data from the US Census website via an API. **NOTE**: For these to work you have to request a free Census API key. Notice the help() function below to get instructions on how to do this.

```
install.packages(c('tidycensus','tigris'))

help('census_api_key','tidycensus')
```

Installing packages used for spatial data manipulation & visualization

This section installs a set of tools specific to our goals of importing, exporting, manipulating, visualizing, and analyzing spatial data. The first line of packages have functions for defining, importing, exporting, and manipulating spatial data. The second line has some tools we will use for visualizing spatial data (e.g. making maps!).

```
install.packages(c('sp', 'sf', 'rgdal', 'rgeos', '
    maptools', 'OpenStreetMap'))
install.packages(c('tmap', 'tmaptools', 'ggmap', 'shinyjs
    ', 'shiny', 'micromap'))
```

Installing packages used for spatial analysis

Finally these are packages specifically to spatial analysis tasks we'll carry out.

```
install.packages(c('spdep', 'CARBayes', 'sparr', '
    spatialreg'))
install.packages(c('GWmodel', 'spgwr') )
```

Part II

Weekly Modules

Chapter 1

Locating Spatial Epidemiology

1.1 Getting Ready

1.1.1 Learning objectives

Table 1.1: Learning objectives by weekly module

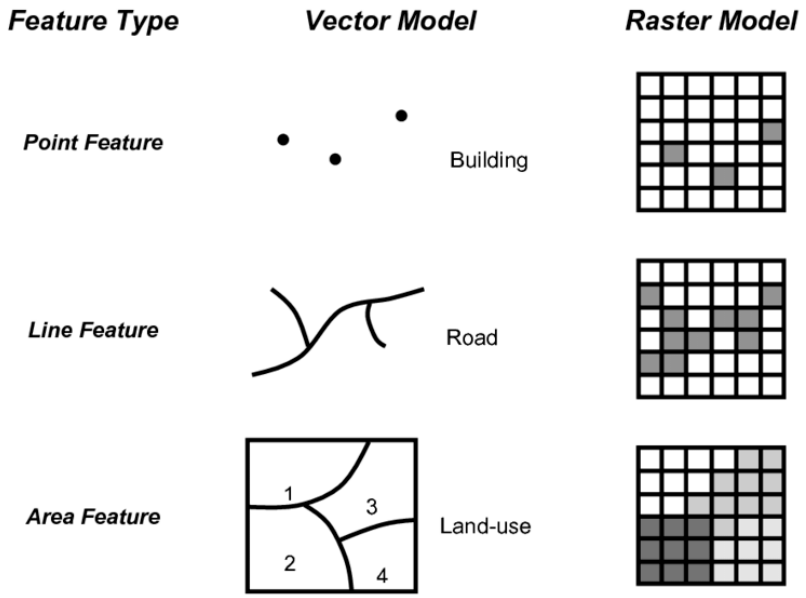
After this module you should be able to...
Explain the potential role of spatial analysis for epidemiologic thinking and practice.
Produce simple thematic maps of epidemiologic data in R.

1.1.2 Additional Resources

- Geocompution with R by Robin Lovelace. This will be a recurring 'additional resource' as it provides lots of useful insight and strategy for working with spatial data in R. I encourage you to browse it quickly now, but return often when you have qusetiona about how to handle geogrpahic data (especially of class sf) in R.
- A basic introduction to the ggplot2 package. This is just one of dozens of great online resources introducing the grammar of graphics approach to plotting in R.
- A basic introduction to the tmap package This is also only one of many introductions to the tmap mapping package. tmap builds on the grammar

of graphics philosophy of ggplot2, but brings a lot of tools useful for thematic mapping!

1.1.3 Important Vocabulary



1.2 Spatial Thinking in Epidemiology

When first learning epidemiology, it can be difficult to distinguish between the concepts, theories, and purpose of epidemiology versus the skills, tools, and methods that we use to implement epidemiology. But these distinctions are foundational to our collective professional identity, and to the way we go about doing our work. For instance do you think of epidemiologists as data analysts, scientists, data scientists, technicians or something else? These questions are bigger than we can address in this class, but their importance becomes especially apparent when learning an area such as spatial epidemiology. This is because there is a tendency for discourse in spatial epidemiology to focus primarily on the data and the methods without understanding how each of those relate to the scientific questions and health of population for which we are ultimately responsible. Distinguishing these threads is an overarching goal of this course, even as we learn the data science and spatial analytic tools.

One quite simplistic but important example of how our questions and methods are inter-related is apparent when we think of **data**. Data is central to quantitative analysis, including epidemiologic analysis. So how is data different in

spatial epidemiology? The first thing that might come to mind is that we have explicitly geographic or spatial measures contained within our data. That the content of the spatial data is distinct: the addition of geographic or spatial location may illuminate otherwise aspatial attributes. But even more fundamental than the content is thinking about the unit of analysis.

It is likely that many other examples in your epidemiology coursework the explicit (or sometimes implicit) unit of analysis has been the individual person. Spatial epidemiology can definitely align with individual-level analysis. But as we'll see, common units we observe and measure in spatial epidemiology – and therefore the units that compose much of our **data** – are not individuals but instead are geographic units (e.g. census tract, county, state, etc) and by extension the collection or aggregation of all the individuals therein. This distinction in unit of analysis has important implications for other epidemiologic concerns including precision, bias, and ultimately for inference (e.g. the meaning we can make from our analysis), as we'll discuss throughout the semester.

One concrete implication of the above discussion is that you should always be able to answer a basic question about any dataset you wish to analyze: "what does one row of data represent?" A row of data is one way to think of the unit of analysis, and often (but not always) in spatial epidemiology a row of data is a summary of the population contained by a geographic unit. Said another way it is an ecologic summary of the population. As stated above, this is only the most simplistic example of how and why it is important to not only learn the spatial statistics and methods, but to also maintain the perspective of epidemiology as a population health science. To advance public health we need good methods but we also need critical understanding of the populations we support, the data we analyze, and the conclusions we can reliably draw from our work.

As we move through the semester, I encourage you to dig deep into how methods work, but also to step back and ask questions like "Why would I choose this method?" or "What question in epidemiology is this useful for?"

1.3 Spatial Analysis in Epidemiology

1.3.1 Spatial data storage foramts

If you have worked with spatial or GIS data using ESRI's ArcMap, you will be familiar with what are called shapefiles. This is one very common format for storing geographic data on computers. ESRI shapefiles are not actually a single file, but are anywhere from four to eight different files all with the same file name but different extensions (e.g. .shp, .prj, .shx, etc). Each different file (corresponding to an extension) contains a different portion of the data ranging from the geometry data, the attribute data, the projection data, an index connecting it all together, etc.

What you may not know is that shapefiles are not the only (and in my opinion **definitely not the best**) way to store geographic data. In this class I recommend storing data in a format called geopackages indicated by the .gpkg extension. Geopackages are an open source format that were developed to be functional on mobile devices. They are useful when we are storing individual files in an efficient and compact way. To be clear, there are many other formats and I make no claim that geopackages are the ultimate format; they just happen to meet the needs for this course, and for much of the work of spatial epidemiologists. It is worth noting that many GIS programs including ArcMap and QGIS can both read and write the geopackage format; so there is no constraint or limitation in terms of software when data are stored in .gpkg format.

1.3.2 Representing spatial data in `R`

The work in this course assumes that you are a basic R user; you do not need to be expert, but I assume that you understand data objects (e.g. data.frame, list , vector), and basic operations including subsetting by index (e.g. using square brackets to extract or modify information: []), base-R plotting, and simple modeling. If you **are not familiar with R**, you will need to do some quick self-directed learning. The instructor and TA's can point you to resources.

Just as our conceptualization of, or thinking about data in spatial epidemiology requires some reflection, the actual storage and representation of that data with a computer tool such as R also requires some attention. Specifically spatial data in R is not exactly like the conventional aspatial epidemiologic data which may exist in R as a rectangular data.frame for example, but it is also need not be as complex as spatial data in software platforms like ESRI's ArcMap.

First, it may be obvious, but spatial data is more complex than simple rectangular attribute data (e.g. data tables where a row is an observation and a column is a variable). To be spatial, a dataset must have a representation of geography, spatial location, or spatial relatedness, and that is most commonly done with either a vector or raster data model (see description above in vocabulary). Those spatial or geographic representations must be stored on your computer and/or held in memory, hopefully with a means for relating or associating the individual locations with their corresponding attributes. For example we want to know the attribute (e.g. the count of deaths for a given place), and the location of that place, and ideally we want the two connected together.

Over the past 10+ years, R has increasingly been used to analyze and visualize spatial data. Early on, investigators tackling the complexities of spatial data analysis in R developed a number of ad hoc, one-off approaches to these data. This worked in the short term for specific applications, but it created new problems as users needed to generalize a method to a new situation, or chain together steps. In those settings it was not uncommon to convert a dataset to multiple different formats to accomplish all tasks; this resulted in convoluted and error-prone coding, and lack of transparency in analysis.

An eventual response to this early tumult was a thoughtful and systematic approach to defining a class of data that tackled the unique challenges of spatial data in R. Roger Bivand, Edzer Pebesma and others developed the sp package which defined spatial data classes, and provided functional tools to interact with them. The sp package defined specific data classes to hold points, lines, and polygons, as well as raster/grid data; each of these data classes can contain geometry only (these have names like SpatialPoints or SpatialPolygons) or could contain geometry plus related data attributes (these have names like SPatialPointsDataFrame or SpatialPolygonsDataFrame). Each spatial object can contain all the information spatial data might include: the spatial extent (min/max x, y values), the coordinate system or spatial projection, the geometry information, the attribute information, etc.

Because of the flexibility and power of the sp∗ class of objects, they became a standard up until the last few years. Interestingly, it was perhaps the sophistication of the sp∗ class that began to undermine it. sp∗ class data was well-designed from a programming point of view, but was still a little cumbersome (and frankly confusing) for more applied analysts and new users. Analysis in spatial epidemiology is not primarily about computer programming, but about producing transparent and reliable data pipelines to conduct valid, reliable, and reproducible analysis. Thus epidemiologists, and other data scientists, desired spatial tools that could be incorporated into the growing toolbox of data science tools in R.

These calls for a more user-friendly and intuitive approach to spatial data led the same team (e.g. Bivand, Pebesma, others) to develop the Simple Features set of spatial data classes for R. Loaded with the sf package, this data format has quickly become the standard for handling spatial data in R. The power of the sf class, as discussed below, is that it makes spatial data behave like rectangular data and thus makes it amenable to manipulation using any tool that works on data.frame or tibble objects. Recognizing that many users and functions prefer the older sp∗ objects, the sf package includes a number of utility functions for easily converting back and forth.

In this class we will use sf∗ class objects as the preferred data class, but because some of the tools we'll learn require sp∗ we will occasionally go back and forth.

sf∗ data classes are designed to hold all the essential spatial information (projection, extent, geometry), but do so with an easy to evaluate data.frame format that integrates the attribute information and the geometry information together. The result is more intuitive sorting, selecting, aggregating, and visualizing.

1.3.3 Benefits of `sf` data classes

As Robin Lovelace writes in his online eBook, Gecomputation in R, sf data classes offer an approach to spatial data that is compatible with QGIS and

PostGIS, important non-ESRI open source GIS platforms, and sf functionality compared to sp provides:

1. Fast reading and writing of data
2. Enhanced plotting performance
3. sf objects can be treated as data frames in most operations
4. sf functions can be combined using %>% pipe operator and works well with the tidyverse collection of R packages (see Tips for using dplyr for examples)
5. sf function names are relatively consistent and intuitive (all begin with st_)

1.3.4 Working with spatial data in R

Here and in lab, one example dataset we will use, called ga.mvc quantifies the counts and rates of death from motor vehicle crashes in each of Georgia's $n = 159$ counties. The dataset is vector in that it represents counties as polygons with associated attributes (e.g. the mortality information, county names, etc).

1.3.4.1 Importing spatial data into R

It is important to distinguish between two kinds of data formats. There is a way that data is stored on a computer hard drive, and then there is a way that data is organized and managed inside a program like R. The shapefiles (.shp) popularized by ESRI/ArcMap is an example of a format for storing spatial data on a hard drive. In contrast, the discussion above about the sf* and sp* data classes refer to how data is organized inside R. Luckily, regardless of how data is stored on your computer, it is possible to import almost any format into R, and once inside R it is possible to make it into either the sp* or sf* data class. That means if you receive data as a .shp shapefile, as a .gpkg geopackage, or as a .tif raster file, each can be easily imported.

All sf functions that act on spatial objects begin with the prefix st_. Therefore to import (read) data we will use st_read(). This function determines **how** to import the data based on the extension of the file name you specify. Look at the help documentation for st_read(). Notice that the first argument dsn=, might be a complete file name (e.g. myData.shp), or it might be a folder name (e.g. mygeodatabase.gdb). So if you had a the motor vehicle crash data saved as both a shapefile (mvc.shp, which is actually six different files on your computer), and as a geopackage (mvc.gpkg) you can read them in like this:

```
# this is the shapefile
mvc.a <- st_read('GA_MVC/ga_mvc.shp')

# this is the geopackage
mvc.b <- st_read('GA_MVC/ga_mvc.gpkg')
```

We can take a look at the defined data class of the imported objects within R:

```
class(mvc.a)
```

```
## [1] "sf"         "data.frame"
```

```
class(mvc.b)
```

```
## [1] "sf"         "data.frame"
```

First, note that when we use the st_read() function, the data class (e.g. the way the data are defined and organized within R) is the same for both mvc.a (which started as a .shp file) and mvc.b (which started as a .gpkg file). That is because st_read() automatically classifies spatial data using sf classes when it imports.

You will also notice that when we examined the class() of each object, they are classified as **both** sf **and** data.frame class. That is incredibly important, and it speaks to an elegant simplicity of the sf* data classes! That it is classified as sf is perhaps obvious; but the fact that each object is also classified as data.frame means that we can treat the object for the purposes of data management, manipulation and analysis as a relatively simple-seeming object: a rectangular data.frame. How does that work? We will explore this more in lab but essentially each dataset has rows (observations) and columns (variables). We can see the variable/column names like this:

```
names(mvc.a)
```

```
## [1] "GEOID"      "NAME"       "MVCRATE_17" "geometry"
```

```
names(mvc.b)
```

```
## [1] "GEOID"      "NAME"       "MVCRATE_17" "geom"
```

We can see that each dataset has the same attribute variables (e.g. GEOID , NAME, MVCRATE_17), and then a final column called geometry in one and called geom in another. These geometry columns are unique in that they don't hold a single value like the other columns; each 'cell' in those columns actually contains an embedded list of x, y coordinates defining the vertices of the polygons for each of Georgia's counties.

Combining these two observations, we now know that we can work with a wide range of spatial data formats, and that once imported we can conceive of (and manipulate!) these data almost as if they were simple rectangular datasets. This has implications for subsetting, recoding, merging, and aggregating data as we'll learn in the coming weeks.

1.3.4.2 Exporting spatial data from R

While importing is often the primary challenge with spatial data and R, it is not uncommon that you might modify or alter a spatial dataset and wish to save it for future use, or to write it out to disk to share with a colleague. Luckily the sf package has the same functionality to write an sf spatial object to disk in a wide variety of formats including shapefiles (.shp) and geopackages (.gpkg). Again, R uses the extension you specify in the filename to determine the target format.

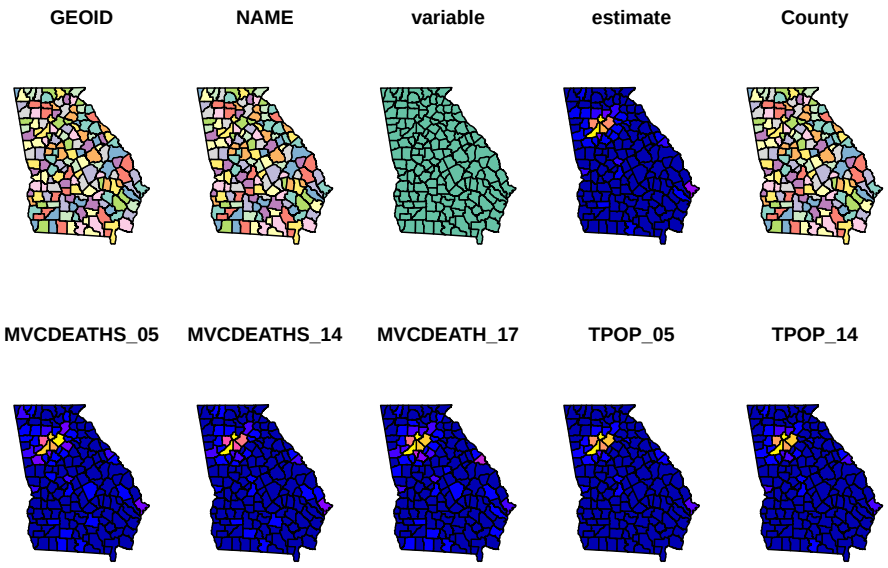
```
# Write the file mvc to disk as a shapefile format
st_write(mvc, 'GA_MVC/ga_mvc_v2.shp')

# Write the file mvc to disk as a geopackage format
st_write(mvc, 'GA_MVC/ga_mvc_v2.gpkg')
```

1.3.5 Basic visual inspection/plots

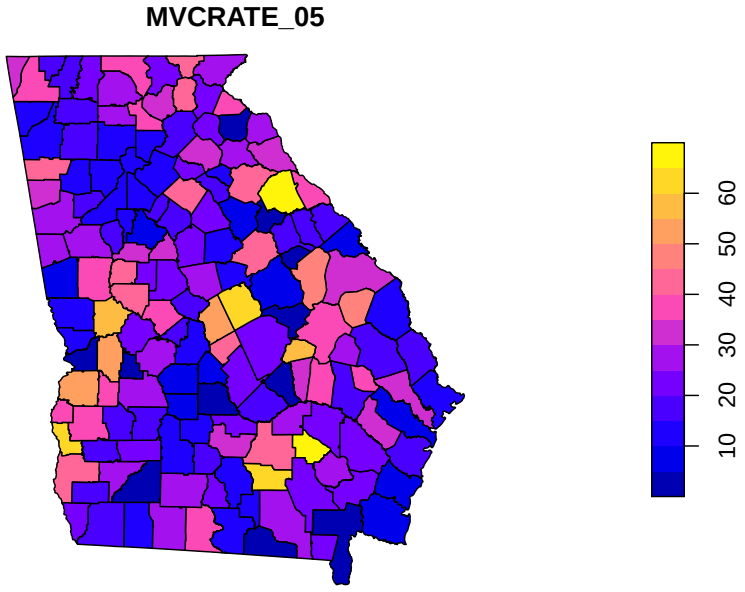
The base-R plot() function is extended by the sf package. That means that if you call plot() on a spatial object **without having loaded** sf, the results will be different than if plot() called **after loading** sf.

When you plot() with sf, by default it will try to make a map **for every variable in the data frame**! Try it once. If this is not what you want, you can force it to only plot some variables by providing a vector of variable names.

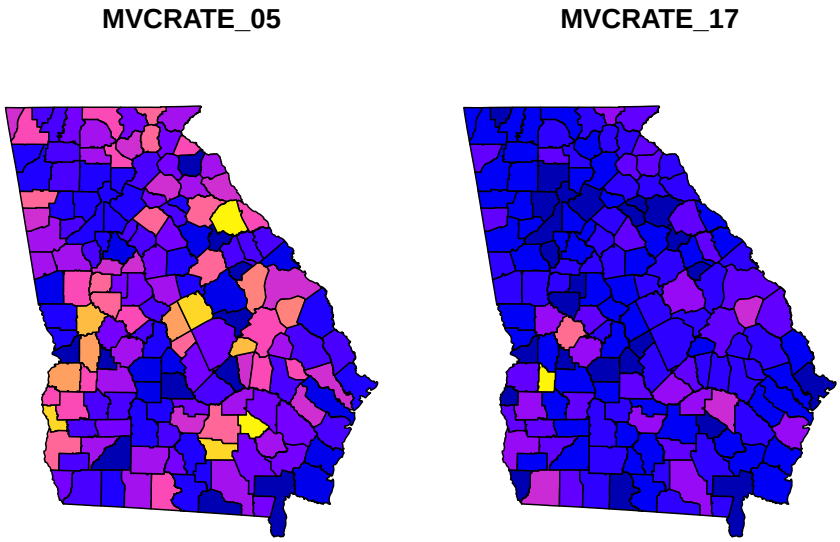
```
plot(mvc) # this plots a panel for every column - or
    actually the first 10 columns
```




```
plot(mvc['MVCRATE_05']) # this plots only a single
    variable, the MVC mortality rate for 2005
```



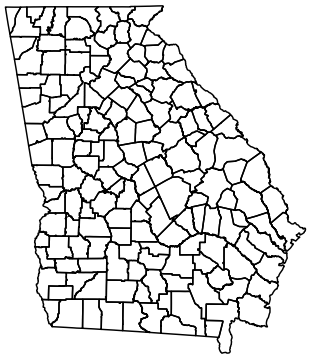

```
plot(mvc[c('MVCRATE_05', 'MVCRATE_17')]) # this plots two
    variables: MVC rate in 2005 & 2017
```



You might only want to see the geometry of the spatial object (e.g. not attributes) if you are checking its extent, the scale, or otherwise confirming something about the spatial aspects of the object. Here are two approaches to quickly plot the geometry:

```
plot(st_geometry(mvc)) # st_geometry() returns the geom
    information to plot
```

```
plot(mvc$geom)  # this is an alternative approach...
    directly plot the 'geom' column
```



1.3.6 Working with CRS and projection

If CRS (coordinate reference system) and projection information was contained in the original file you imported, it will be maintained. **If there is NO CRS information imported it is critical that you find out the CRS information from the data source!** The most unambiguous way to describe a projection is by using the **EPSG** code, which stands for European Petroleum Survey Group. This consortium has standardized hundreds of projection definitions in a manner adopted by several R packages including rgdal and sf.

This course is not a GIS course, and learning about the theory and application of coordinate reference systems and projections is not our primary purpose. However some basic knowledge is necessary for successfully working with spatial epidemiologic data. Here are several resources you should peruse to learn more about CRS, projections, and EPSG codes:

- A useful overview/review of coordinate reference systems in R
- Robin Lovelace's Geocompuation in R on projections with sf
- EPSG website: This link is to a searchable database of valid ESPG codes
- Here are some useful EPSG codes

The choice of CRS and/or projection has a substantial impact on how the produced map looks, as is evident in the figure above (source of image).

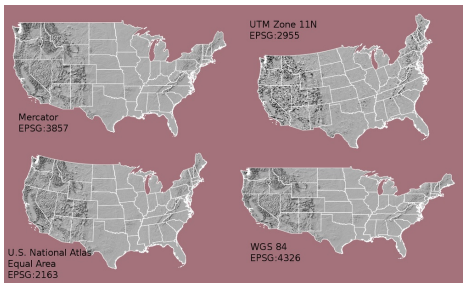


Figure 1.1: Comparing CRS

We already saw the CRS/projection information of the mvc object when we used the head() function above; it was at the top and read WGS 84. Recall there are two main types of CRS: purely **geographic** which is to say coordinate locations are represented as latitude and longitude degrees; and **projected** which means the coordinate values have been transformed for representation of the spherical geoid onto a planar (Euclidean) coordinate system. WGS 84 is a ubiquitous geographic coordinate system common to boundary files retrieved from the U.S. Census bureau.

An important question when you work with a spatial dataset is to understand whether it is primarily a geographic or projected CRS, and if so which one.

```
st_is_longlat(mvc)
```

```
## [1] TRUE
```

This quick logical test returns TRUE or FALSE to answer the question "Is the sf object simply a longitude/latitude geographic CRS?". The answer in this case is TRUE because WGS 84 is a geographic (longlat) coordinate system. But what if it were FALSE or we wanted to know more about the CRS/projection?

```
st_crs(mvc)
```

```
## Coordinate Reference System:
##   User input: WGS 84
##   wkt:
## GEOGCRS["WGS 84",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
```

```
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["unknown"],
##         AREA["World"],
##         BBOX[-90,-180,90,180]],
##     ID["EPSG",4326]]
```

This somewhat complicated looking output is a summary of the CRS stored with the spatial object. There are two things to note about this output:

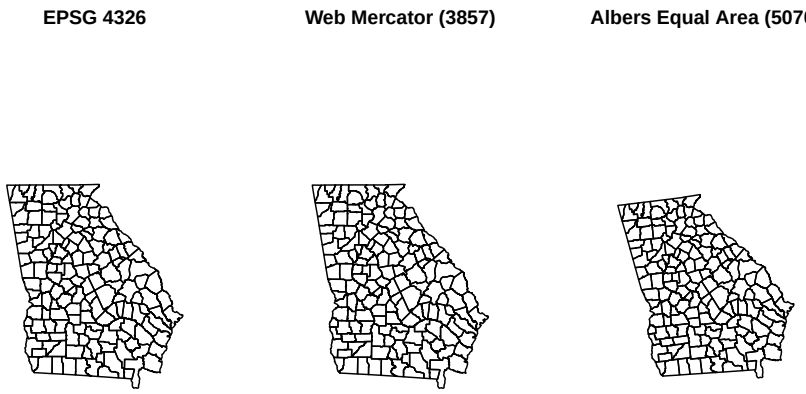
- At the top, the User input is WGS 84
- At the bottom of the section labeled GEOGCRS it says ID["EPSG",4326"]

While there are literally hundreds of distinct EPSG codes describing different geographic and projected coordinate systems, for this semester there are three worth remembering:

- **EPSG: 4326** is a common geographic (unprojected or long-lat) CRS
- **EPSG: 3857** is also called WGS 84/Web Mercator, and is the dominant CRS used by Google Maps
- **EPSG: 5070** is the code for a projected CRS called Albers Equal Area which has the benefit of representing the visual area of maps in an equal manner.

Once the CRS/projection is clearly defined, you may choose to transform or project the data to a different system. The sf package has another handy function called st_transform() that takes in a spatial object (dtaaset) with one CRS and outputs that object transformed to a new CRS.

```
# This uses the Albers equal area USA,
mvc.aea <- st_transform(mvc, 5070)

# This uses the Web Mercator CRS (EPSG 3857) which is
    just barely different from EPSG 4326
mvc.wm <- st_transform(mvc, 3857)

# Now let's look at them side-by-side
plot(st_geometry(mvc), main = 'EPSG 4326')
plot(st_geometry(mvc.wm), main = 'Web Mercator (3857)')
plot(st_geometry(mvc.aea), main = 'Albers Equal Area
    (5070)')
```



Do you see the difference between the three? Because EPSG 4326 and 3857 are both unprojected (e.g. they are long/lat), they appear quite similar but are not identical. Albers Equal Area, on the other hand, is more distinct. In general we will prefer to use 'projected' rather than 'unprojected' (long/lat only) data for both visualization and analysis. That means that whenever you bring in a new dataset you will need to check the CRS and project or transform as desired.

Important: It is important to distinguish between defining the current projection of data and the act of projecting or transforming data from one known system to a new CRS/projection. **We cannot transform data until we correctly define its current or original CRS/projection status.** The above function tells us what the current status is. In some cases data do not have associated CRS information and this might be completely blank (for instance if you read in numerical x, y points from a geocoding or GPS process). In those cases you can **set** the underlying CRS using st_set_crs() to define it, but this assumes you **know** what it is. There are two arguments to this function: the first is x = objectName, and the second is value = xxx where 'xxx' is a valid EPSG code.

Table 1.2: Vocabulary for Week 1

Term	Definition
Data, attribute	Nonspatial information about a geographic feature in a GIS, usually stored in a table and linked to the feature by a unique identifier. For example, attributes of a county might include the population size, density, and birth rate for the resident population
Data, geometry	Spatial information about a geogrpahic feature. This could include the x, y coordinates for points or for vertices of lines or polygons, or the cell coordinates for raster data
Datum	The reference specifications of a measurement system, usually a system of coordinate positions on a surface (a horizontal datum) or heights above or below a surface (a vertical datum)
Geographic coordinate system	A reference system that uses latitude and longitude to define the locations of points on the surface of a sphere or spheroid. A geographic coordinate system definition includes a datum, prime meridian, and angular unit
	A method by which the curved surface of the earth is portrayed on a flat surface. This generally requires a systematic mathematical transformation of the earth's graticule of lines of longitude and latitude onto a plane. Some projections can be visualized as a transparent globe with a light bulb at its center (though not all projections emanate from the globe's center) casting lines of latitude and longitude onto a sheet of paper. Generally, the paper is either flat and placed tangent to the globe (a planar or azimuthal projection) or formed into a cone or cylinder and placed over the globe (cylindrical and conical projections). Every map projection distorts distance, area, shape, direction, or some

Chapter 2

Cartography for Epidemiology I

2.1 Getting Ready

2.1.1 Learning objectives, w2

Table 2.1: Learning objectives by weekly module

After this module you should be able to...
Design a cartographic representation of epidemiologic data that is consistent with best practices in public health data visualization.
Apply data processing functions to accomplish foundational data management and preparation for spatial epidemiology (e.g. summarize, aggregate, combine, recode, etc)

2.1.2 Additional Resources, w2

- CDC Guidance for Cartography of Public Health Data (complements required reading)
- When Maps Lie
- Color Brewer Website for color guidance in choropleth maps

2.1.3 Important Vocabulary, w2

2.2 Spatial Thinking in Epidemiology, w2

Making pretty maps is not the full extent of spatial epidemiology. However, epidemiologic cartography can sometimes be the beginning and end of spatial epidemiology for a given purpose. And even when an epidemiologic analysis goes well beyond mapping (perhaps to incorporate aspatial analysis, or to incorporate more sophisticated spatial analysis), the ability to produce a clear, concise, and interpretable map is an important skill.

As Robb, et al[1] write:

> Disease mapping can be used to provide visual cues about disease etiology, particularly as it relates to environmental exposures….Mapping where things are allows visualization of a baseline pattern or spatial structure of disease, potential detection of disease clusters, and the initial investigation of an exposure-disease relationship.

There are aspects of cartography and map design that are general to most thematic maps of quantitative data. But there are some issues that seem especially pertinent to us as epidemiologists or quantitative population health scientists. These include the decisions we make about color choice and the process of categorizing numerical data for visual representation in a map.

Why are these especially important for epidemiology? A primary purpose of a map is to visually represent something meaningful about the spatial or geographic variation in health or a health-relevant feature (e.g. an exposure or resource). Communicating what is meaningful and representing variation that matters is not solely a technical GIS task; it requires epidemiologic insight. For instance our approach to representing ratio measures such as an odds ratio or risk ratio should be different from how we represent risk or rate data, because we understand that the scale and units are distinct in each case. Similarly, we understand that understanding variation or heterogeneity in a normal or Gaussian (bell-shaped curve) distribution is different from a uniform or a highly skewed distribution with a long right tail. This insight into how scales and values are differently interpretted epidemiologically muts be translated into sensible choices in mapping.

[1] Robb SW, Bauer SE, Vena JE. Integration of Different Epidemiological Perspectives and Applications to Spatial Epidemiology. Chapter 1 in Handbook of Spatial Epidemiology. 2016. CRC Press, Boca Raton, FL.

2.2.1 Color choices

For most thematic maps, color is the most flexible and important tools for communication. Color, hue, and contrast can accentuate map elements or themes and minimize others. The result is that you can completely change the story your map tells with seemingly small changes to how you use color. This means you should be clear and explicit about why you choose a given color or sequence of colors, and beware of unintentionally misrepresenting your data by your color choices.

In producing choropleth maps, we often talk about collections of colors as color ramps or color palettes, because a single color by itself is not very interesting. A quick scan of either the tmaptools::palette_explorer() utility, or the Color Brewer website will demonstrate that there are many colors to choose from, so is it just a matter of preference? Perhaps, but there are some guidelines to keep in mind.

2.2.1.1 Sequential palettes

All color palettes use the color hue, value, or saturation to represent or symbolize the values of the underlying statistical parameter of interest. When a parameter or statistic is naturally ordered, sequential and monotonic, then it makes sense to choose colors that range from light to dark. Conventionally lighter or more neutral tones represent lower or smaller numbers and darker colors and more intense tones represent higher or larger numbers. The dark colors jump out at the viewer more readily, so occasionally the inverse is used to emphasize small values, but this should be done with caution as it can be counterintuitive.

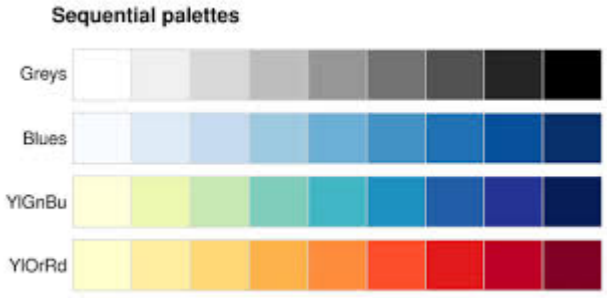


Figure 2.1: Sequential color palettes

Sequential palettes are useful for epidemiologic parameters such as prevalence, risk, or rates, or continuous exposure values where the emphasis is on distinguishing higher values from lower values.

2.2.1.2 Diverging palettes

A less common choice, but one that is especially important for some epidemiologic parameters, is the diverging palette. In this pattern, the neutral color is in the center of the sequence, with two different color hues become darker and more intense as they go out from the center.

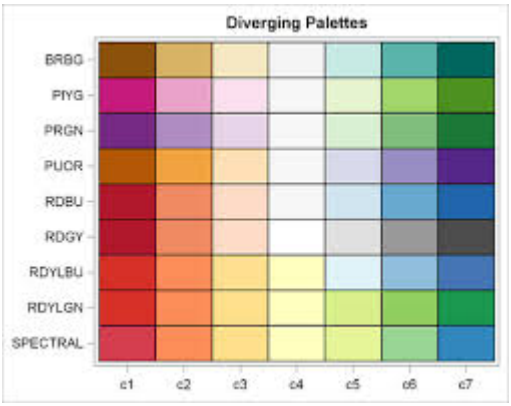


Figure 2.2: Diverging color palettes

You might choose this color sequence for one of two reasons:

1. You wish to show how units vary around the overall mean or median, highlighting those that are larger than versus smaller than the overall mean/median. For instance diverging palettes might emphasize areas with particularly high burden of disease (and therefore in need of additional attention), as well as those with unexpectedly low burden of disease (and therefore worthy of understanding about protective factors).
2. You are mapping any epidemiologic parameter on the ratio scale where there are values both above and below the null ratio of 1.0. For example if you map Standardized Mortality/Morbidity Ratios, risk or odds ratios, or prevalence ratios, you potentially have diverging data. The exception would be if all of the ratio values were on the same side of the null (e.g. all were $>> 1$ or $<< 1$).

In the map above, the SMR (a ratio of the county-specific prevalence of very low birth weight infants to the overall statewide live birth prevalence) varies from 0.13 to 2.30. But this range is not sequential in the same way as a risk or prevalence. Instead the neutral color is assigned to counties in the range of $0.90 - 1.10$, around the null. This is a way of indicating these counties are average or typical. In contrast, counties with increasing excess morbidity have darker green, and substantially lower morbidity are darker purple.

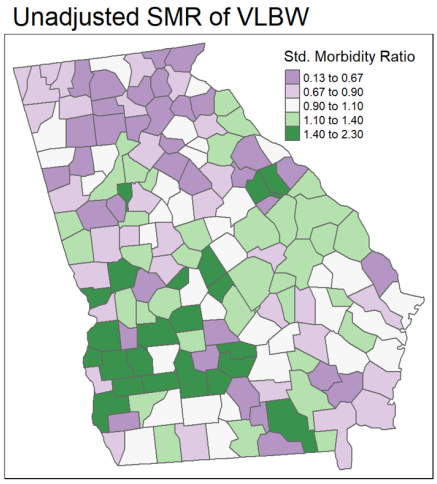


Figure 2.3: Mapping ratio measure with divergent palette

2.2.1.3 Qualitative palettes

Qualitative refers to categories that are not naturally ordered or sequential. For instance if counties were assigned values for the single leading cause of death in the county, we might choose a qualitative palette, as a sequential or diverging palette might mislead the viewer into thinking there is some natural ordering to which causes should be more or less intense in their color.

2.2.2 Choropleth binning styles

A second topic relevant to the intersection of cartography and epidemiologic thinking is the means by which we choose cutpoints for visualizing data. In other words for a map to visually represent some underlying statistical value, we have to assign or map numerical values to colors. How you do that depends greatly on the intended message or story your map needs to tell. Are you interested in distinguish units that rank higher or lower in values? Or are you primarily focused on finding extreme outliers, with variation in the 'middle' of the distribution of less interest? These distinct purposes give rise to different decisions about how to assign colors to numerical values in your data.

As discussed in the lecture, there are numerous methods or styles for categorizing continuous data for choropleth mapping (e.g. identical data is summarized under four different styles in figure above). Cynthia Brewer (of ColorBrewer fame) and Linda Pickle (**?**) sought to evaluate which styles are most effective for communicating the spatial patterns of epidemiologic data.

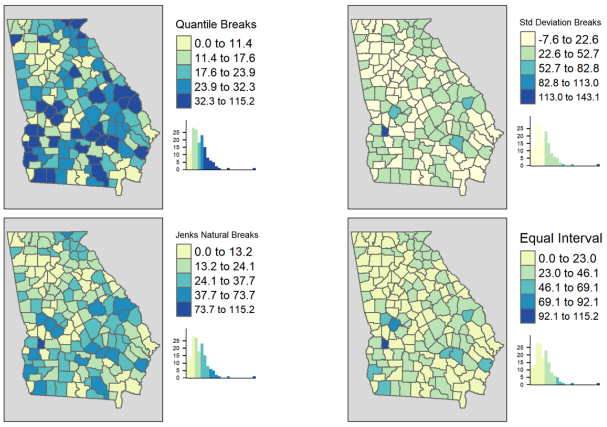


Figure 2.4: Comparing binning styles with same data

As cartographers, Brewer & Pickle were critical of the epidemiologists' over-reliance on quantile cutpoints, given many other strategies that seemed to have cartographic advantages. However, after randomizing map 'readers' to interpret maps of the same underlying epidemiologic data using seven different styles, they determined that readers could most accurately and reliably interpret the disease patterns in maps using quantile cutpoints. While there are benefits of the other styles for some purposes, for the common use of communicating which spatial areas rank higher or lower in terms of disease burden, quantiles are most straightforward.

2.2.2.1 Mapping time series

It is common in spatial epidemiology that we want to map the spatial patterns of disease for several different snapshots in time as a series to observe the evolution of disease burden over time. But changing patterns over time raises additional questions about how to make cuts to the data. There are several options for determining the cutpoints when you have a time series:

1. Pool all of the years data together before calculating the cutpoints (e.g. using quantiles). Use the pooled cutpoints for all years.
2. Create custom year-specific cutpoints that reflect the distribution of data for each year separately.
3. Create cutpoints based on a single year and apply them to all other years.

The map above of Georgia motor vehicle crash mortality data in three different years (2005, 2014, 2017), was created in tmap using the tm_facet() option where the the by = was year. As a result, the quantile cutpoints represent the breaks

Figure 2.5: Georgia MVC deaths by year with a common scale

pooling all observations across the three years. In other words the cutpoints come from 159 counties times three years: 477 values.

By having a common legend that applies to all three maps, this strategy is useful for comparing differences in absolute rates across years.

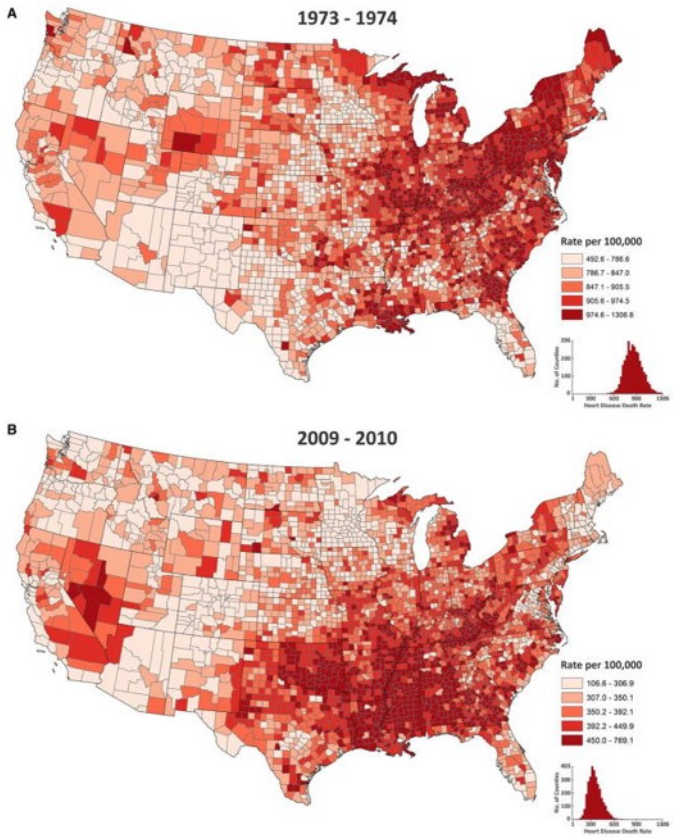


Figure 2.6: U.S. heart disease mortality with a year-specific scales

The map above of heart disease mortality rates by county in two years (1973-4; 2009-10) uses quantile breaks calculated separately for each time period. This was done in part because the heart disease mortality rate declined so much between these years that a scale that distinguished highs from lows on one map would not distinguish anything on the other map. In this case what is being compared is not the absolute rates but the relative ranking of counties in the two years.

2.3 Spatial Analysis in Epidemiology, w2

Every spatial epidemiology project must include attention to data acquisition, cleaning, integration, and visualization. The specific workflow is driven largely by the overarching epidemiologic question, purpose, or goal. In this section we use a specific question to illustrate key steps to data preparation for epidemiologic cartography.

> **Case Example Objective**: Create a choropleth map visualizing geographic variation in the all-cause mortality rate for U.S. counties in 2016-2018. Compare this to a choropleth map of % uninsured in U.S. counties.

This objective will be directly relevant for the lab this week as well as for the Visualizing US Mortality, Visual Portfolio, an assignment due later in the semester.

Although this specific question dictates specific data needs, these four types of data are frequently needed to produce a map of a health outcome or state:

1. Numerator data, in this case representing the count of deaths per county in the target year
2. Denominator data, in this case representing the population at risk for death in each county in the target year
3. Contextual or covariate data, in this case the prevalence uninsured for each U.S. county
4. Geometric data representing the shapes and boundaries of U.S. counties

2.3.1 Obtaining and preparing numerator data

The event of interest (e.g. the numerator in a risk, rate, or prevalence) can come from many sources. If you are conducting primary data collection, it arises from your study design and measurement. When using secondary data, it is common to use surveillance data (e.g. vital records, notifiable diseases, registries, etc) or administrative data as a source of health events.

When using secondary data sources owned or managed by another entity, one challenge that can occur is suppression of data to protect privacy. For example the National Center for Health Statistics mortality data available from CDC WONDER suppresses the count of deaths, as well as the crude mortality rate, whenever the numerator count is less than ten events. There can also be instances when a local or state public health agency fails to report data to NCHS, producing missing values.

Depending on the data format, it is possible that either **missing** or **suppressed** data could be inadvertently imported into R as zero-count rather than missing. It is therefore critically important to understand the data source and guidelines. The decision about how to manage zero, missing, and suppressed data is an epidemiologic choice, but one that must be addressed before creating a map.

How to deal with data suppression. There are many reasons your target data may fall below thresholds for suppressions. Perhaps the outcome event is quite rare, or you are stratifying by multiple demographic factors, or perhaps you are counting at a very small geographic unit. If suppression is problematic for mapping, consider pooling over multiple years, reducing demographic stratification, or using larger geographic areas to increase event count and reduce the number of suppressed cells.

For this example, we have downloaded all-cause mortality counts by county from CDC WONDER for 2016-2018 (pooling over three years to reduce suppression). In Lab we will discuss the procedure for acquiring data from the web. After importing the data this is how it appears.

```
head(death)
```

2.3.2 Obtaining and preparing denominator or contextual data

The mortality data accessed from CDC included both numerator (count of deaths) and denominator (population at risk). However there are instances where you may have one dataset that provides the health event data (numerator), but you need to link it to a population denominator in order to calculate risk, rate, or prevalence. The U.S. Census Bureau maintains the most reliable population count data for the U.S., and it is available in aggregates from Census Block Group, Census Tract, Zip code tabluation area, City or Place, County, State, and Region.

Census data can be aggregated as total population or stratified by age, gender, race/ethnicity, and many other variables. The census data also contains measures of social, economic, and housing attributes which may be relevant has context or exposures in spatial epidemiologic analyses. There are two broad types of data demographic and socioeconomic data released by the Census Bureau.

- **Decennial Census** tables which (theoretically) count 100% of the population every 10 years. These can be cross-classified by age, race/ethnicity, sex, and householder status (e.g. whether head of house owns or rents and how many people live in house)

- **American Community Survey (ACS)** tables which provide a much larger number of measures but are based on samples rather than complete counts. The ACS began in the early 2000's and is a continually sampled survey. Despite being collected every year, for many small areas (e.g. census tracts or even counties) there are not enough responses in a single year to make reliable estimates. Therefore ACS data pooled into 5-year moving-window datasets. For instance the 2014-2018 ACS (the most recent release) reports estimates for all responses collected during that time period, and these are available from the Census Block Group up. The next release will probably come in late 2020, and will be for 2015-2019.

You may have accessed Census or ACS data directly from the Census Bureau website for other classes or tasks in the past. In the interest of reproducibility and efficiency, we introduce the tidycensus package in R. It is an excellent tool for acquiring either Decennial Census or ACS data directly within R. The advantage of doing so is twofold: first it can be quicker once you learn how to do it; second, it makes your data acquisition fully reproducible without any unrecorded steps happening in web browsers.

We will practice the code in the next few sections in lab. It is included here as a primer. In these sections I walk through **one way** to download and prepare data to quantify the county-level prevalence of the population who are uninsured, as this might be a covariate of interest when examining spatial variation in mortality. I selected the code below because it is relatively efficient, although you may find some of it complex or confusing. I include it for those who would like to explore other data-manipulation functions in R. Please note that you do not need learn all of the functions in this Census data acquisitions section below for this course, although you might find these or related approaches useful. Note also that there are many ways to accomplish anything in R, and you could achieve the same ends with different strategies.

2.3.2.1 Setting up Census API

To access any Census products (e.g. attribute tables or geographic boundary files) using the tidycensus package, you need to register yourself by declaring your API key. If you haven't already done so, go here to register for the key.

```
# Only do this if you haven't already done it; it should
    only need to be done once.

tidycensus::census_api_key('YourKeyHere', install = T)
```

2.3.2.2 Choosing Variables

By far the biggest challenge of requesting data from the Census Bureau is knowing what you want, and where it is stored. Census data are distributed as aggregated counts contained in specific tables (each has a unique ID), and made up of specific variables (also a unique ID composed of table ID plus a unique ID). There are two ways to find variables:

- You could go to the Census website and browse around. For instance the Census Data Explorer website is one way to browse the topics and variables
- You could download all of the variables for a given year into R, and use filters to search it.

This code queries the Census website (assuming you have internet connection) and requests a list of all variables for the ACS 5-year pooled dataset (e.g. acs5) and for the window of time ending in 2018 (e.g. 2014-2018). I also specify cache = T which just means to save the results for quicker loading if I ask again in the future.

```
library(tidycensus)

all_vars <- load_variables(year = 2018, dataset = 'acs5',
    cache = T)

head(all_vars)
```

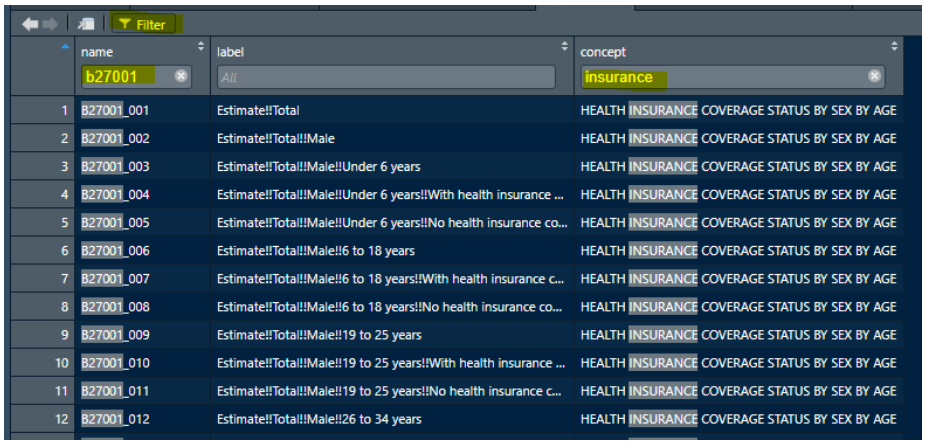
It may be easiest to look at the dataset using the View() function. When you do so, you see the three variables, and you have the option to click the **Filter** button (upper left of View pane; looks like a funnel). The Filter option is one way to search key words in either the label or concept column.

We are interested in capturing the prevalence of uninsured in each county. Try this:

- Go to View mode of variables (e.g. View(all_vars))
- Click the Filter button
- Type insurance in the concept field
- Type B27001 in the name field

What we want is a list of the specific tables and variable ID's to extract from the Census. In lab we will use some more detailed code to accomplish this goal.

You may have noticed that the full list of ACS variables has nearly 27,000 variables! In the code below I use some tricks to filter the huge list of all variables to get only the names I want. It relies on the tidyverse package stringr which



	name	label	concept
	b27001	All	insurance
1	B27001_001	Estimate!!Total	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
2	B27001_002	Estimate!!Total!!Male	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
3	B27001_003	Estimate!!Total!!Male!!Under 6 years	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
4	B27001_004	Estimate!!Total!!Male!!Under 6 years!!With health insurance ...	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
5	B27001_005	Estimate!!Total!!Male!!Under 6 years!!No health insurance co...	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
6	B27001_006	Estimate!!Total!!Male!!6 to 18 years	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
7	B27001_007	Estimate!!Total!!Male!!6 to 18 years!!With health insurance c...	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
8	B27001_008	Estimate!!Total!!Male!!6 to 18 years!!No health insurance co...	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
9	B27001_009	Estimate!!Total!!Male!!19 to 25 years	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
10	B27001_010	Estimate!!Total!!Male!!19 to 25 years!!With health insurance ...	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
11	B27001_011	Estimate!!Total!!Male!!19 to 25 years!!No health insurance c...	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE
12	B27001_012	Estimate!!Total!!Male!!26 to 34 years	HEALTH INSURANCE COVERAGE STATUS BY SEX BY AGE

Figure 2.7: Screenshot of RStudio View() of ACS variables

is great for manipulating character variables (this is great for many data science tasks; read more about stringr here). In this case I am using it to filter down to just the table I want (e.g. B27001), and then to get the names of the variables that contain the string 'No health insurance'.

Here is the list of variables we want to acquire; each one represents a count of uninsured at each of multiple age groups. We will sum them up to get a total population uninsured prevalence.

```
##  [1] "B27001_001" "B27001_005" "B27001_008" "
    B27001_011" "B27001_014"
##  [6] "B27001_017" "B27001_020" "B27001_023" "
    B27001_026" "B27001_029"
## [11] "B27001_033" "B27001_036" "B27001_039" "
    B27001_042" "B27001_045"
## [16] "B27001_048" "B27001_051" "B27001_054" "
    B27001_057"
```

2.3.2.3 Retrieving data from Census

To actually retrieve data from the Census we use the function get_acs() (or if you were getting decennial data the function would be get_decennial()). When you request data you must specify the geography (e.g. do you want counts for states, counties, census tracts, census block groups?), the variables, the year, and the dataset. Look at ?get_acs to read more about options.

The following code chunks use the dplyr and tidyvrse verbs and the %>% (pipe) to connect data steps together. This is complex at first, but it is worth carefully

examining how each step works. If you are not familiar with this syntax, it would probably be helpful to review the Appendix section on dplyr.

```
# First, request the data from ACS
insure_tidy <- get_acs(geography = 'county',
                       variables = myVars,
                       year = 2018,
                       survey = 'acs5') %>%
  select(-moe)

# Look at the resulting object
head(insure_tidy)
```

Looking at the first few rows of the data object insure_tidy above, you might be surprised that there is a column labeled variable, and the cells within that column are actually what we thought were the variable names! That is because these data are structured in a tidy format, which happens to be long not wide. Read more about transposing data here. In the following steps we will reshape this data to be more useful.

What this code does:

- define the geography = as county.
- Specify the vector (previously created and named myVars) of variables to download
- Specify the year of interest. Note that 2018 references the 2014-2018 5-year window
- specify the survey, which will most often be acs5

```
# Now I pull out the denominator
insure_denom <- insure_tidy %>%
  filter(variable == 'B27001_001') %>%
  rename(TOTPOP = estimate) %>%
  select(-variable)

# Look at the resulting object
head(insure_denom)
```

The code above was necessary because most of the variables were age-specific counts of the number of uninsured people. But one variable, B27001_001 is the count of all included in the table. In other words, it is the denominator for calculating the prevalence of uninsured. Therefore I did the following in the code above:

- filter () restricts to only the rows of data where the variable is the denominator count (B27001_001). Filter is like where in SAS

- rename() is a way to rename variables to my own liking
- select () drops the variable called variable

```
# Now I sum up all the variables for the numerator
insure_num <- insure_tidy %>%
  filter(variable != 'B27001_001') %>%
  group_by(GEOID) %>%
  summarise(no_insure = sum(estimate))

head(insure_num)
```

The code above addresses an issue common to census tables: they may not be constructed in the way you want them. As discussed above, in this case the values are counts for each age group, but we only want a single count for the entire population of each county. Therefore, it is necessary to sum across or aggregate the counts over all age groups to get a single count (the numerator number of uninsured) for each county. The strategy used above was specific to the data being in long format, which happens to be tidy data in this case. Read about changing between long and wide here.

The code above achieves this through steps:

- filter () using the != mean "is **not** equal to"; this simply removes the denominator variable, so that we are only summing over numerator counts
- group_by() is a very useful dplyr verb; it is similar to using class in SAS, and tells R to do something separately for each group (e.g. each GEOID or county in this case)
- summarise() is a verb that works hand-in-hand with group_by(). The grouping declares which groups, but the summarise() tells what to do. In this case we just want to count up all of those uninsured across all age gruops.

```
# Finally, merge the numerator and denominator in order
    to calculate prevalence
uninsured <- insure_denom %>%
  left_join(insure_num, by = 'GEOID') %>%
  mutate(uninsured = no_insure / TOTPOP) %>%
  select(GEOID, uninsured)

# Take a look at the resulting object
head(uninsured)
```

This was a simple merge, but it is worth mentioning a few of the steps:

- left_join () is one of a famiy of merging verbs. The left in left_join () simply means start with the first table (the one on the left) and merge

with the second table. The implications are with whether all rows or only rows in the left or the right (first or second) table are retained. In this case the left of first table is insure_denom and the right or second table is insure_num)
- mutate() calculates the uninsured prevalence
- select () excludes unnecessary variables

The code process above was complex. While it was specific to this exact scenario, each scenario might require different steps. The challenge for you, the new spatial analyst, is to think through in your mind how the data looks at the beginning and how you want it to look at the end. Then create a sequence of steps that progressess from beginning to end. It takes practice, but is worthwhile for spatial epidemiology, but also for data science and processing more generally.

2.3.3 Obtaining and preparing geographic data

The final type of data needed is the geographic or geometry data. Again, the source for geometry data varies by the study specifics: you may need polygons (e.g. political or administrative boundaries), lines (e.g. transportation networks), or points (e.g. hospitals, food stores, toxic waste sites, etc). On the other hand you may need or have data that are in raster format, including weather or air pollution surfaces. There are open-access versions of many types of geographic data online.

For choropleth mapping, area units including administrative and political boundaries are commonly used. In the U.S. context, the Census geographies are frequently used, including blocks, block groups, tracts, zip-code tabluation areas, counties, cities & places, metropolitan areas, tribal areas, states, and regions. In this section I provide a brief introduction to downloading census boundary files directly into R.

2.3.3.1 Obtain geometry data from `tidycensus`

The first option is a very minor modification to the code in the previous section acquiring census count data. The get_acs() function has an argument geometry = that is FALSE by default. However, if you change it to geometry = TRUE, you will automatically retrieve the data as an sf object including a geometry column!

```
insure_tidy <- get_acs(geography = 'county',
                       variables = myVars,
                       year = 2018,
                       geometry = TRUE,    # added geometry
                          = T
                       survey = 'acs5')
```

One other argument to get_acs() not demonstrated here is shift_geo. It is FALSE by default, but if set to shift_geo = TRUE, it will return boundaries that have been projected to Albers Equal Area, and where the states of Hawaii and Alaska are artificially shifted to fit on a thematic map of the U.S.

2.3.3.2 Obtain geometry data from `tigris`

The tidycensus package actually requests the geometry by depending on another package called tigris (the Census geography files are called TIGER files). If you are obtaining both attributes (e.g. population counts) and geometries at the same time, the tidycensus package makes the most sense. However, sometimes you only need the geometry, perhaps because the other data come from sources other than the Census Bureau.

If you want to directly obtain areal boundary units, coastline data, road or rail networks, voting districts, or other spatial data maintained by the Census Bureau, consider using the tigris package. Try looking at the help documentation (e.g. ? tigris , then click the Index link at the bottom to see all of the options).

Here I demonstrate by retrieving the U.S. county boundaries:

```
library(tigris)
options(tigris_use_cache = TRUE)
us <- counties(cb = TRUE,
                           resolution = '5m',
                           year = 2018,
                           class = 'sf')
```

Here is what the code above does:

- The counties() function is one of dozens in tigris for downloading specific kinds of boundary data
- cb = TRUE adjusts the level of detail or resolution of the boundaries. By default cb = FALSE returns the most detailed data, which is quite large. Setting cb = TRUE defaults to a generalized (1:500k scale) shape.
- resolution = '5m' is a further specification that I want an even more generalized boundary file. The 1:5 million scale is more coarse in terms of resolution of curves in county boundaries, but it is also a smaller file. You must decide the balance between file size and resolution for a specific need.
- year = 2018 specifies which vintage of boundary files. Tracts, counties, cities, etc all change boundaries from year to year.
- class = 'sf' results in the object returned being a sf object, rather than sp class data (the default).

```
summary(us)
```

```
##    STATEFP            COUNTYFP           COUNTYNS
          AFFGEOID
## Length:3233        Length:3233        Length:3233
       Length:3233
## Class :character   Class :character   Class :
   character   Class :character
## Mode  :character   Mode  :character   Mode  :
   character   Mode  :character
##
##
##
##     GEOID                NAME               LSAD
                ALAND
## Length:3233        Length:3233        Length:3233
       Min.   :8.209e+04
## Class :character   Class :character   Class :
   character   1st Qu.:1.079e+09
## Mode  :character   Mode  :character   Mode  :
   character   Median :1.563e+09
##

   Mean   :2.833e+09
##

   3rd Qu.:2.367e+09
##

   Max.   :3.770e+11
##      AWATER                  geometry
## Min.   :0.000e+00   MULTIPOLYGON :3233
## 1st Qu.:7.038e+06   epsg:4269    :   0
## Median :1.950e+07   +proj=long...:   0
## Mean   :2.161e+08
## 3rd Qu.:6.159e+07
## Max.   :2.599e+10
```

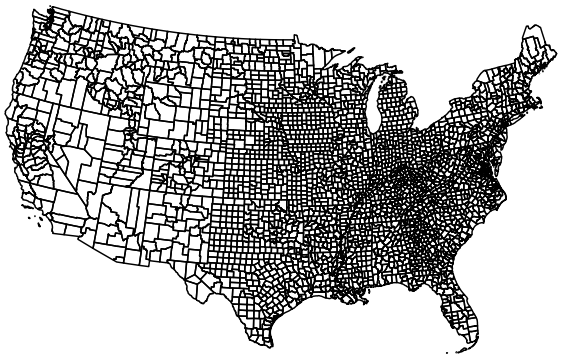
We can see from the summary that the data has a CRS/projection EPSG code of 4269 (it is unprojected).

What does this boundary file look like?

```
plot(st_geometry(us))
```

The Census boundaries include information for all U.S. counties and territories! Therefore the map looks this way because Guam, American Samoa, Puerto Rico, as well as Hawaii and Alaska are included. If you were only interested in mapping the "*lower 48" or contiguous states, you could exclude these. In the code below, I also transform or project the data to Albers Equal Area using EPSG code

```
us <- us %>%
  filter(!(STATEFP %in% c('02', '15', '66', '60', '78', '
      72', '69'))) %>%
  select(GEOID, STATEFP, COUNTYFP, NAME) %>%
  st_transform(5070)

plot(st_geometry(us))
```



2.3.4 Merging Attributes and Geography

A final step in data preparation is bringing together the attribute data and the geometry data, assuming it has not already been incorporated. Assuming the attributes are a data.frame (or perhaps a tibble, which is a tidyverse data table object), and the geometry is a sf object (which also has class data.frame), the merge is straightforward. Here is what is needed for merging or joining data:

- Unique key or ID variable in the attribute data that matches with the ID in the geometry data
- Unique key or ID variable in the geometry data that matches with the ID in the attribute data
- Matching ID's **does not require same variable name** but **does require same variable type**.

If you are merging several datasets, and one of them is an sf object, put that dataset first in the sequence, as that will insure that the final object remains of class sf. If you cannot put the sf first, you may need to re-define the object as sf at the end. See the Appendix on st_as_sf() for more detail.

```
us2 <- us %>%
  left_join(death, by = c('GEOID' = 'FIPS')) %>%
  left_join(uninsured, by = 'GEOID')
```

2.3.5 Mapping Mortality & Uninsured

```
library(tmap)

t1 <- tm_shape(us2) +
  tm_fill('crude',
          style = 'quantile',
          palette = 'BuPu',
          title = 'Rate per 100,000 py') +
  tm_borders(alpha = 0.2) +
tm_credits('Source: CDC Wonder',
           position = c('RIGHT', 'BOTTOM')) +
tm_layout(main.title = 'All-cause mortality rate, 
    2016-2018',
          bg.color = 'grey85')

t2 <- tm_shape(us2) +
  tm_fill('uninsured',
          style = 'quantile',
          palette = 'YlOrRd',
          title = '% Uninsured',
          legend.format = list(fun=function(x) paste0(
              formatC(x * 100,
                                                      digits
                                                        =1,

                                                      format
                                                        =
                                                        "
                                                        f
                                                        "
                                                        )
                                                        ,

                                                        "
                                                        %
                                                        "
                                                        )
                                                        )
                                                        )

                                                        +

  tm_borders(alpha = 0.2) +
tm_credits('Source: American Community Survey',
           position = c('RIGHT', 'BOTTOM')) +
```

```
tm_layout(main.title = 'Uninsured Prevalence, 2014-2018',
          bg.color = 'grey85')

tmap_arrange(t1, t2, ncol = 1)
```

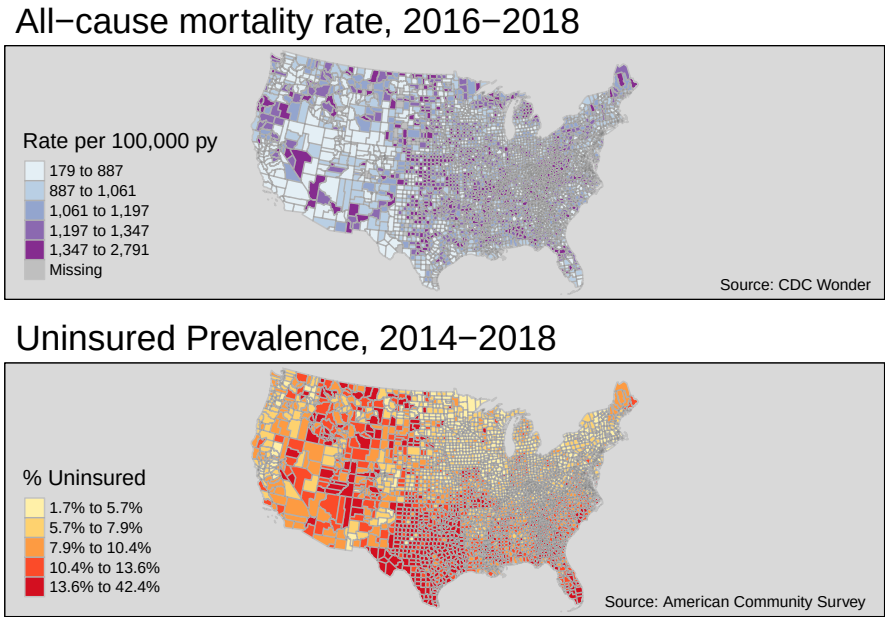


Table 2.2: Vocabulary for Week 2

Term	Definition
Cartography	The production of maps, including construction of projections, design, compilation, drafting, and reproduction
Choropleth map	A type of thematic map in which areas are shaded or patterned in proportion to a statistical variable that represents an aggregate summary of a geographic characteristic within each area, such as population density, disease risk, or standardized mortality ratio
Color palette: diverging	Diverging schemes allow the emphasis of a quantitative data display to be progressions outward from a critical midpoint of the data range. A typical diverging scheme pairs sequential schemes based on two different hues so that they diverge from a shared light color, for the critical midpoint, toward dark colors of different hues at each extreme
Color palette: qualitative	Qualitative schemes use differences in hue to represent nominal differences, or differences in kind. The lightness of the hues used for qualitative categories should be similar but not equal.
Color palette: sequential	Sequential data classes are logically arranged from high to low, and this stepped sequence of categories should be represented by sequential lightness steps. Low data values are usually represented by light colors and high values represented by dark colors. Transitions between hues may be used in a sequential scheme, but the light-to-dark progression should dominate the scheme.
	A type of thematic map that uses contour lines or colors to indicate areas with similar regional aspects. It typically symbolizes the underlying statistic as varying continuously in space, in contrast to the discrete unit-specific variation of

Table 2.3

FIPS	County	Deaths	Population	crude
01001	Autauga County, AL	536	55601	964
01003	Baldwin County, AL	2357	218022	1.08e+03
01005	Barbour County, AL	312	24881	1.25e+03
01007	Bibb County, AL	276	22400	1.23e+03
01009	Blount County, AL	689	57840	1.19e+03
01011	Bullock County, AL	112	10138	1.1e+03

Table 2.4

GEOID	NAME	variable	estimate
01001	Autauga County, Alabama	B27001_001	5.43e+04
01001	Autauga County, Alabama	B27001_005	36
01001	Autauga County, Alabama	B27001_008	157
01001	Autauga County, Alabama	B27001_011	397
01001	Autauga County, Alabama	B27001_014	354
01001	Autauga County, Alabama	B27001_017	500

Table 2.5

GEOID	NAME	TOTPOP
01001	Autauga County, Alabama	5.43e+04
01003	Baldwin County, Alabama	2.05e+05
01005	Barbour County, Alabama	2.29e+04
01007	Bibb County, Alabama	2.05e+04
01009	Blount County, Alabama	5.72e+04
01011	Bullock County, Alabama	9.98e+03

Table 2.6

GEOID	no_insure
01001	3.88e+03
01003	2.09e+04
01005	2.56e+03
01007	1.62e+03
01009	6.3e+03
01011	1.08e+03

Table 2.7

GEOID	uninsured
01001	0.0714
01003	0.102
01005	0.112
01007	0.0791
01009	0.11
01011	0.108

Chapter 3

Cartography for Epidemiology II

3.1 Learning objectives, w3

Table 3.1: Learning objectives by weekly module

After this module you should be able to...
Describe potential threats to privacy and research ethics that arise when population health data is represented geographically
Critique spatial epidemiologic literature based on consistency with ethical principles of privacy, avoidance of harm through stigmatization, and balance of benefit and risk

3.2 Additional resources, w3

- Report on confidentiality issues and policies related to geospatial data for public health applicatins

3.3 Important Vocabulary, w3

3.4 Spatial Thinking in Epidemiology, w3

> "Progress in achieving health for all depends upon effectively collecting, integrating, and utilizing medical, public health, socioeconomic, environmental, and physical science data."

> "Although new technological advances can empower individuals and neighborhoods seeking resources for better health care, they have also heightened concerns about individual privacy and confidentiality."

> – Confidentiality Issues and Policies Related to the Utilization and Dissemination of Geospatial Data for Public Health Applications

Ethical concern for justice, beneficence, and respect for persons ground guidelines and practices in responsible conduct of public health research. When we work with geospatial data these concerns are not lessened but instead often are heightened, because of the power of locational information as a means for discerning private information and the risk for intended or unintended breaches of confidentiality and even the transmission of stigma to groups by highlighting health status in marginalized populations.

3.4.1 Risks of privacy breaches in collection of geospatial information

Geographic identifiers below the scale of the state (e.g. county, city, census tract, address) are considered Protected Health Information under HIPAA if they are connected to individual health information. Surveillance and research activities routinely collect geospatial information for contact or notification purposes, or for reporting, although many consent forms do not explicitly explain the intended purpose or use of the geospatial information.

While any individual should expect protection of privacy not only of individual PHI such as date of birth or name, it is not always explicit that information such as address can be uniquely identifiable and is linkable to other data. Privacy is breached when app-based geocodes are captured without consent, or when geospatial information is collected without express consent (e.g. if a research respondent is asked to report the address for someone in their social network without that persons consent).

While respect for personal autonomy dictates that individuals should be permitted control of private information, there can also be risks beyond breach of privacy. In some instances, disclosed geospatial information could result in harms to the participant or others. For example collected address information could inadvertently be released to someone seeking to commit violence (e.g as in the case of intimate partner violence). Similarly, studies collecting geospatial information can (and have) been requested by force of law to aid in the investigation or prosecution of suspected crimes. Thus the collection of geospatial information must be well reasoned with respect to risk and benefit to the participant, with appropriate notification and consenting process, and protections in place to maintain confidentiality.

3.4.2 Risk of confidentiality breaches through unintentional de-identification

Once private geospatial data has been collected, there is a responsibility for data owners (e.g. public health agencies, researchers) to protect the confidentiality of that disclosed private information. Confidentiality protection refers to both the secure control of confidential data as well as the avoidance of the unintended re-identification of data deemed 'de-identified' through data linkages.

Maintaining data security is critical for all public health research and surveillance activities, but sometimes geospatial data is ignored as a unique identifier. In one instance I submitted a data request to a public health agency to obtain surveillance data on abortion incidence. The data was delivered as an Excel sheet where individual identifiers such as name and date of birth were removed, but the field for address of residence was included. An address is an incredibly powerful unique identifier, particularly when combined with other fields including age or sex.

Geospatial data can be stored separately from other research attributes, maintaining only a key for linkage in the instances when the spatial data are needed. When they are not needed, there is less risk of accidental disclosure of these fields.

Another risk that drives many public health agencies restrictive guidelines around data suppression and reporting, is the concern for re-identification of individuals from aggregated data because of small cell size and the ability to discern identity from quasi-identifiers. For example, age, race, ethnicity, or health outcome could each be quasi-identifiers in some instances when cross-tabulation make individuals unique or nearly so.

In a study of the 1990 decennial census, researchers found that 87% of the U.S. population could be uniquely identified with only three variables: exact date of birth, zip code, and gender! This is due in part to the combined granularity or specificity of two variables: date of birth and zip code. In most instances, reporting health events at the zip code level without respect to age, or perhaps

with age categorized in coarse groups would eliminate the risk. But the take home message is that the stratification of data we prefer for better scientific understanding can quickly lead to at least some sub-groups being individually or nearly individually identifiable.

3.4.3 Risk of stigmatization of place

A final ethical concern that is particularly relevant for disease mapping activities is concern for unintentional harm of persons or populations through the stigmatization of place. This can happen when a map identifies locations where marginalized populations spend time, and serves to either further stigmatize that group, or stigmatize others unassociated with the group, but sharing the same location. Such stigmatization can lead to psychosocial harms, but also can alter behavior by other institutional forces including social services, law enforcement, and health services.

Examples of stigmatization of place include the identification of venues where men who have sex with men seek partners, or the mapping of concentrations of commercial sex workers or injection drug users. But the concern for stigmatization of place has also been raised from the point of view of social epidemiology, when predominantly Black and brown neighborhoods are repeatedly characterized as 'unhealthy'. The potential harm perpetrated by these maps could arise from the (presumably well-intended) desire to highlight unjust burdens, but the failure to similarly highlight resilience in the face of burdens.

Relatedly, many spatial representations of economic and racial disparities fail to name the factors that give rise to the inequities, including the role of sociohistorical and structural discrimination. By failing to name structural racism or policies that serve to concentrate affluence separately from concentrated poverty, the maps contribute to a narrative that the communities are in some way to blame for their health outcomes.

Table 3.2: Vocabulary for Week 3

Term	Definition
Confidentiality	The duty of anyone entrusted with health information to keep that information private
Ethical principles: beneficence	Two general rules have been formulated as complementary expressions of beneficent actions in this sense: (1) do not harm (e.g. non-maleficence) and (2) maximize possible benefits and minimize possible harms
Ethical principles: justice	Ethical principle that the burdens and benefits of research and public health practice should be justly distributed, including attention to need, effort, contribution, and merit
Ethical principles: respect for persons	Defined by two ethical convictions: a) individuals should be treated as autonomous agents; b) persons with diminished autonomy are entitled to protection
Geomask	A class of methods for changing the geographic location of an individual in an unpredictable way to protect confidentiality, while trying to preserve the relationship between geocoded locations and disease occurrence (Sherman and Fetters 2007, Wiggins 2002)
Privacy	The right of an individual to keep his or her information (health related or otherwise) private

Chapter 4

Disease Mapping I

4.1 Getting ready, w4

4.1.1 Learning objectives, w4

Table 4.1: Learning objectives by weekly module

After this module you should be able to...
Determine and defend appropriate disease mapping strategies consistent with basic epidemiologic concepts (e.g. study design, sampling strategy, measurement error, and systematic bias)
Create statistically smoothed, age-adjusted disease maps of epidemiologic parameters including SMR, disease risk or rate, and measures of estimate precision/stability
Describe the modifiable areal unit problem and discuss strategies for evaluating bias arising from MAUP

4.1.2 Additional Resources, w4

- Arianna Planey blog on spatial thinking and MAUP
- Waller L, Gotway C. Applied Spatial Statistics for Public Health Data. Hoboken, NJ: John Wiley & Sons, Inc; 2004.
- Clayton D, Kaldor J. Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. Biometrics. 1987 Sep;43(3):671–81.

4.1.3 Important Vocabulary, w4

4.2 Spatial Thinking in Epidemiology, w4

Disease mapping is located at the intersection of statistics, geography, and epidemiology. Whereas the out-of-the-box GIS approach to making maps of health statistics (e.g. what I've been referring to as epidemiologic cartography) takes raw data and simply shows it on a map, disease mapping typically implies that we are interested in going beyond just making pretty maps. Instead we are driven by core epidemiologic questions and concerned about fundamental epidemiologic and statistical issues.

4.2.1 Why do we need disease mapping?

The defining driver or purpose of epidemiology is an interest in characterizing and estimating the distribution and determinants of health in populations. Disease mapping is primarily focused on the former (distribution of health), providing novel insight into the geographic patterns of health and disease. The latter (determinants of health) can begin to be addressed by Modules 3 and 4 of this course focusing on Clustering and Spatial Regression.

To spatially describe the distribution of disease, epidemiologists are primarily interested in one over-arching question:

> **Is the intensity of disease or health spatially heterogeneous or spatially homogenous?**

Spatial heterogeneity simply implies that the global parameter (e.g. rate, risk, prevalence, etc) for an entire study area is not identical to the local parameter in every sub-region of that study area. In contrast, spatial homogeneity means that if you know the overall, global parameter, you basically know every local parameter, plus or minus random variation. Looking for heterogeneity is the whole reason for mapping. If the occurrence of disease were the same everywhere, a map would not tell us much! In previous weeks we mapped disease, but our epidemiologic cartography efforts to date fall short because we did not attend to the following three challenges:

1. Parameter estimate instability due to sparse data/rare events;
2. Spurious heterogeneity arising from 'confounding' by nuisance covariates;
3. Biased description of spatial heterogeneity arising from the modifiable areal unit problem (MAUP), a form of the ecologic fallacy

4.2.1.1 The problem and approach to data sparsity

Reliable and precise estimation of any parameter presumes we have sufficient data to produce a summary (e.g. a measure of central tendency like a mean, prevalence, risk, etc). When either a disease is quite rare – resulting in a small numerator – or the population at risk is quite sparse – resulting in a small denominator – the estimate of disease burden is inherently unstable. That means that adding just one or two more events or persons at risk has a notable impact on the estimate. For instance imagine a county with 10 people. In one year, perhaps none die, in the next year one dies, and in the third year three die. The mortality rate is estimated at 0%, 10% and 30%, when none of those seems very plausible as an average expected mortality. The problem is the estimate of mortality rate is derived from too little data.

In practice, public health agencies often suppress data when counts are small, both out of concern for confidentiality, but also because the resulting estimates are so unstable as to be potentially misleading. We have already discussed two approaches to address data sparsity and the resulting parameter instability or imprecision:

- Aggregate over more time to increase the opportunity for events, or extend the amount of person-time
- Aggregate over geographic units to pool together larger populations. For example data for mortality may be too sparse at the census tract level but might be stable after pooling all tracts to their respective county level.

We will spend the next several weeks exploring a range of methods that together constitute a third option: statistical smoothing or stabilization. These tools use the amount of information (as a function of sample size) to smooth extreme highs and extreme lows in an effort to recover a plausible 'true' amount of spatial heterogeneity. A critical goal of disease rate stabilization is that we do not smooth any more than is necessary, so that true highs and lows persist, but spurious or unstable values are adjusted.

This week we will use aspatial or global Empirical Bayes estimators as our first approach to parameter stabilization. In future weeks we will explore spatial Empirical Bayes, kernel density estimators, and fully Bayesian estimators as additional strategies for producing maps that highlights the signal of spatial heterogeneity net of the noise from random error.

4.2.1.2 The problem and approach to confounding

Confounding in epidemiology refers to a specific causal structure, wherein the association between a putative exposure and a target disease outcome is spuriously biased because of a backdoor path through one or more confounders. In disease mapping we do not have a formal 'exposure', with place perhaps being

a stand-in for unmeasured attributes that vary through space. Therefore we probably should not call this confounding in the strictest sense of the word.

Instead you can imagine that there are covariates that are simply a nuisance. That means they explain some amount of spatial heterogeneity, but you as the epidemiologist are not particularly interested in their role as an explanation; instead you wish to know if there is still heterogeneity above and beyond those covariates. For example consider comparison of mortality rates by state:

State	Crude mortality rate (per 100,000)	Age-adjusted mortality rate (per 100,000)
Florida	957.3	666.6
Alaska	605.7	745.6

Using the crude mortality rate, it is clear that Florida has a mortality rate perhaps 30% higher than Alaska, suggesting something really awful is going on in Florida! However once we adjust or standardize by age, it is actually Alaska that has a slightly higher mortality rate. Depending on your purpose both numbers are useful, but if mapping mortality across states, you might think that differences in age-structure (e.g. many more retirees in Florida than Alaska) is a nuisance to accomplishing the goal; so for disease mapping an age-adjusted estimate may be more useful.

The strategies in spatial epidemiology for addressing confounding (e.g. removing the effects of nuisance variables) is similar to those in non-spatial epidemiology. Standardization, stratification, and regression control are conventional tools. In disease mapping it is quite common to use standardization as a tool to balance or condition on one or more covariates, such as age. However there are methods including the fully Bayesian models and later spatial regression models, where it is possible to control for multiple covariates.

4.2.1.3 The problem and approach to MAUP

In this interesting article about the Flint water-lead crisis, a geographer, Richard Sadler, describes mapping some lead-level data from Flint early in the process. There had been some alarms raised about high levels of lead in Flint, but state-based reporting did not identify or detect anomalies. As the geographer points out, this was likely because state-based reporting was based on (aggregated to) zip codes. While zip codes are not ideal geographic units for any disease mapping, it may not be apparent exactly why zip codes could have led public health officials astray in Flint. Until you look at the map of zip code boundaries overlaid city boundaries.

As you can see, there are seven zip codes in the Flint area, but only two of them are fully contained within the city limits. The others seem evenly split between

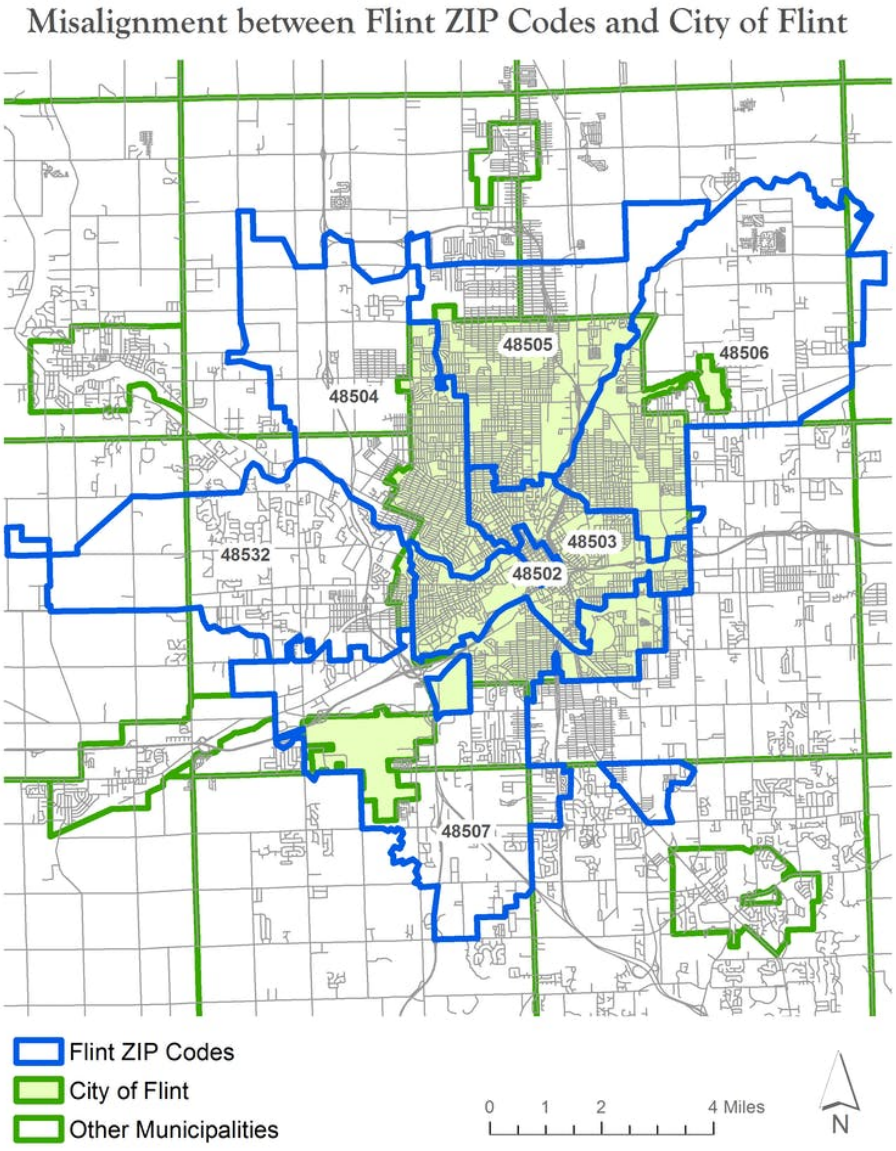


Figure 4.1: Zip code boundaries in Flint, Michigan

areas inside the city limits and outside the city. This became important because the water system issues that produced excess lead exposure were constrained to households inside the city limits. The net result was that aggregation of events (high blood lead levels) and population at risk within each zip code area contained a mix of truly exposed and unexposed households. The zip code reporting masked or obscured the true elevations, diluting the early warnings of a problem.

This is a powerful example of the concern referred to by geographers as the modifiable areal unit problem (MAUP). Epidemiologists may be familiar with a related idea: the ecologic fallacy or ecologic bias. The problem is not inherently about aggregation. Instead the problem arises when the way data are aggregated results in a mixing of different types of people, producing a kind of cross-level confounding. In Flint this meant diluting the population with people exposed to clean water, but it could also result from enriching a specific region with people with confounding risk factors, producing a spurious estimate of the true experience of health within the area.

There are two ways that the MAUP can occur:

1. **Arbitrary zoning** or boundaries to create aggregates. This is the case in Flint, where one (arbitrary) zoning system (zip codes) was applied to a different zoning system (e.g. municipal city boundaries). The result is a mis-alignment between what is actually happening and the way we count it up.
2. **Arbitrary scale** or level of aggregation. This occurs when we aggregate to a level or scale that is different from the level or scale at which population health is generated. There is no single 'right' scale. It depends on the process of interest. The 'correct' scale for understanding the effect of Medicaid expansion under the ACA is likely different from the 'correct' or best scale for understanding the role of healthy commercial food retailers on obesity.

One key take away from the above discussion is that the bias from the MAUP arises when the way we carry out an analysis does not align with the way that health occurs. In other words, not all aggregation or zoning are similarly harmful. The work for the spatial epidemiologist is to consider how aligned (or mis-aligned) the available aggregation is with respect to the hypothetical process. Sometimes it is possible to explore sensitivity of results to choice of scale or zoning but repeating analyses with alternative boundaries or scales.

4.2.2 Using statistics and probability models to improve disease mapping

In epidemiology, we spend a lot of time trying to disentangle 'noise' from 'signal' in collections of data and their relationships. This is evident in our focus on

two broad buckets of error: random error that comes from chance and is related to sample size; and systematic error that comes from structural bias (e.g. confounding, selection, misclassification/measurement error) that is not driven by sample size and is therefore not fixed by increasing sample size).

To make inference (make meaning or decisions) from data that take account of random error we adopt statistical probability models that describe the role of chance alone in generating values. For instance many statistics operate under assumptions related to Gaussian or normal distributions. We also rely on Poisson and binomial distributions to evaluate variation and differences for count and binary data respectively.

4.2.2.1 How are statistics different in space?

Spatial statistics is a huge field, well broader than what we will cover this week, or this entire semester. However it is worth introducing a few key ideas to motivate the statistics we will be using.

Health events typically occur at the level of the individual, and individuals can be referenced with respect to their location in space. Consider, for example a study region represented by the blue square in the image below. There is a population distributed across the region, occupying any particular x, y location. In this population defined by geographic bounds, there may be some individuals experiencing a health event. The set of points observed at a point in time represents a specific realization of a spatial point process. In other words we can imagine each individual having some random chance of experiencing the event, and the set of events indexed by their location is one realization or version of the random process.

To describe or quantify what is observed we could describe the spatial disease intensity of the event as a spatially continuous surface. In other words for every location, the intensity is the amount of disease per unit-area. To calculate a single, global, measure of spatial intensity for the figure above we divide events by area:

$\frac{events}{Area} = \frac{14}{4km^2} = \frac{3.5}{km^2}$

In this simplistic case we assumed the population at risk was evenly distributed across the study region. More realistically, we can normalize events to the spatially-varying population at risk to quantify the spatial intensity of disease.

Because we often do not have the exact x, y location of every person at risk and every health event, we cannot observe the full spatial point process and thus cannot estimate the continuous spatial intensity surface. However, we can approximate the spatial intensity by aggregating health events and population and summarizing the ratio (e.g. as risk, rate, prevalence) per areal unit. In the figure above, each rectangle contains $n = 100$ person-years at risk, producing the following disease rates estimating the spatial intensity of disease:

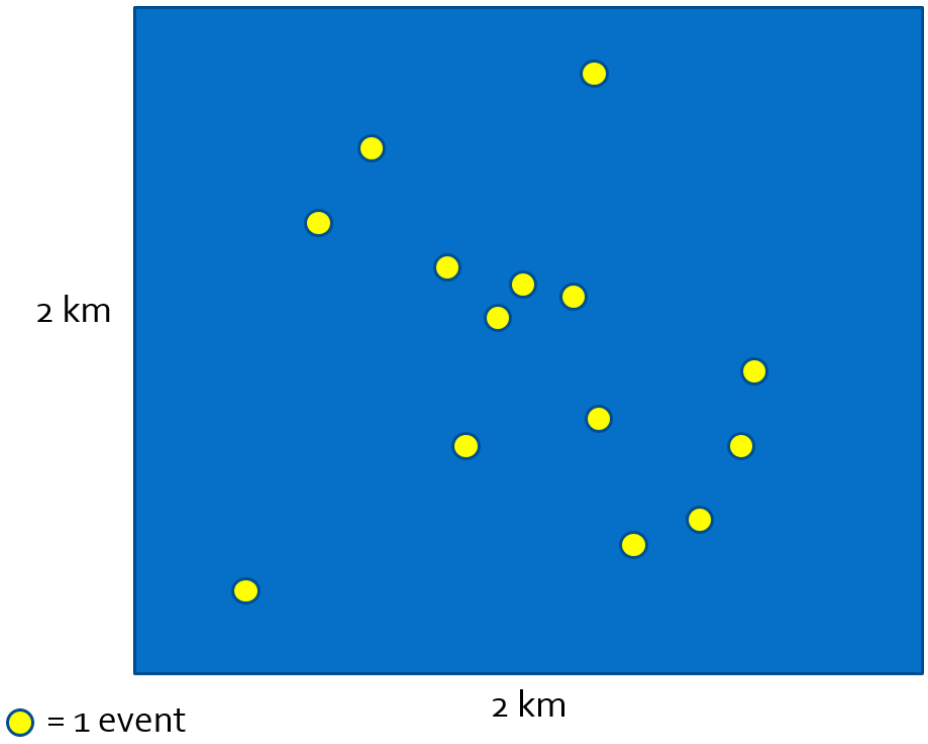


Figure 4.2: Spatial point process

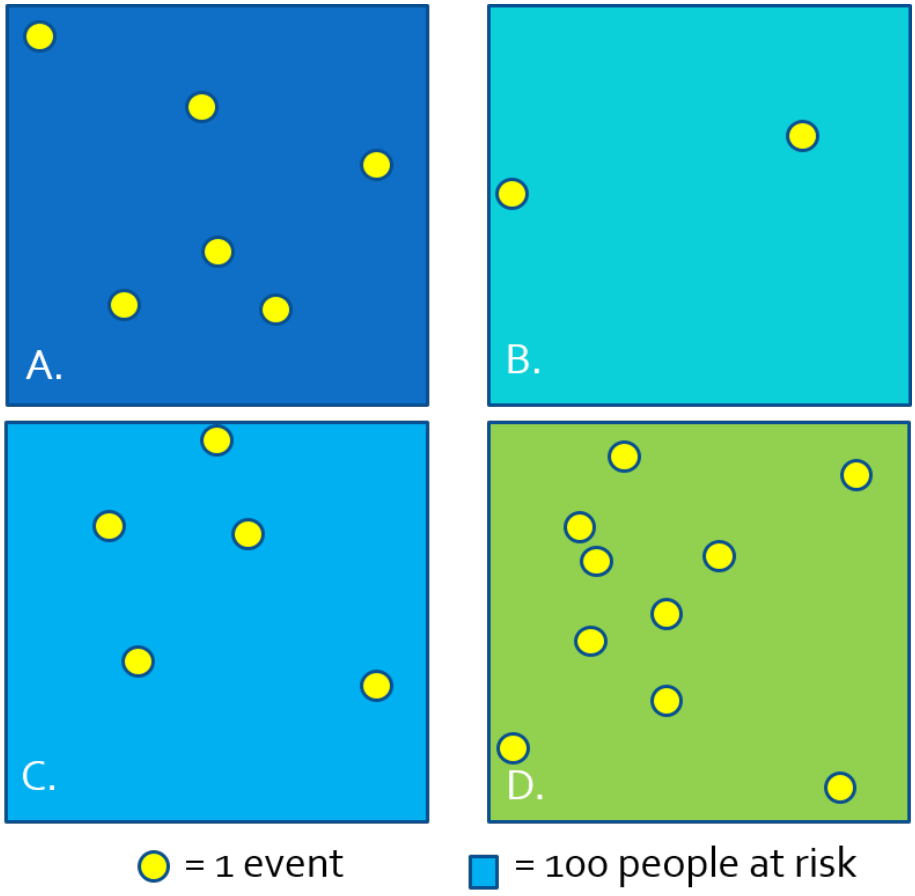


Figure 4.3: Approximating intensity with areal aggregates

Region	$\frac{events}{population}$	Estimate
A	$\frac{6}{100}$	6%
B	$\frac{2}{100}$	2%
C	$\frac{5}{100}$	5%
D	$\frac{10}{100}$	10%

When we have data in this form (e.g. counts of events and counts of population), we can use one of several parametric statistical probability distributions common in epidemiology including Poisson, binomial, and negative binomial.

Why are probability distributions useful?

Because they provide a model for describing what to expect fro data due to random chance alone. Specifically, relating the count of disease events to a probability distribution permits the calculation of standard errors or confidence intervals expressing the precision or certainty in an estimate. Alternatively we could calculate a p-value as a means to test evidence for consistency with a null hypothesis.

Here is a brief summary of probability distributions common to disease mapping:

Distribution	Paramaterization	MLE and comments
Binomial	$Y_i \vert r_i \sim Bin(N_i, r_i)$	$\hat{r_i} = \frac{Y_i}{N_i}$
Poisson	$Y_i \vert \theta_i \sim Poisson(E_i\theta_i)$	$\theta_i = \frac{Y_i}{E_i}$
Poisson-gamma mixture (a.k.a negative binomial)	$Y_i \vert \theta_i \sim Poisson(E_i\theta_i)$, $\theta_i \sim gamma(\alpha, \beta)$	Note the gamma distribution explains how much the θ_i varies. In Bayesian framework the gamma is a prior for θ.

In the formulas above:

- Y_i is the count of health events in the i_{th} areal unit
- N_i is the count of population at risk in the i_{th} areal unit
- r_i is the risk in the i_{th} areal unit
- E_i is the expected count, which is calculated as $N_i \times r$, where r is an overall reference level of risk. So expected simply means the burden of disease in the i_{th} areal unit if they experienced the reference risk.
- θ_i is the relative risk in the i_{th} areal unit; this is essentially the relative deviation of this region from the expected.

Don't panic looking at these formulas. Here are some take away points:

- **Poisson distribution** is a classic distribution to use for evaluating counts of events possibly offsetting by the time-at-risk or person-years.
  - Poisson assumes that the mean of the distribution is the same as the variance of the distribution.
  - Poisson distribution only approximates the disease intensity rate well for rare disease processes. Therefore Poisson is not a good choice if the outcome is not rare.
- **Binomial distribution** is useful for characterizing disease occurrence for non-rare or common disease processes.
- **Poisson-gamma Mixture** may be the most foreign. However, you may have heard of the Negative Binomial distribution for count data? Poisson-gamma mixture is essentially a negative binomial model. It is a probability distribution like the Poisson, except without the expectation that the mean is equal to the variance. In other words it is robust to what is called over-dispersion, when the variation in the count is greater than expected under the Poisson.
  - Over-dispersion is quite common in spatial epidemiology because there often are unobserved factors driving the occurrence of disease in each area, and those unobserved differences produce event intensity that is not strictly Poisson in nature. We will use Poisson-gamma for this reason.

If you want to learn more about Poisson point processes or probability distributions for spatial epidemiology, I highly recommend Lance Waller's text, Applied Spatial Statistics for Public Health Data (Waller & Gotway, 2004). It is available electronically via Woodruff Library.

4.3 Spatial Analysis in Epidemiology, w4

As an example dataset, for the next four weeks of disease mapping we will aim to estimate the spatial heterogeneity at the county level of the occurrence of very low birthweight (VLBW; weight at birth < 1500 grams) babies in 2018-2019. These data were derived from the Georgia OASIS website. This outcome is of public health importance because of the high morbidity and mortality associated with being born so early or so small. However, with an overall rate of VLBW of only 1.8%, it is a rare outcome, and there will likely be sparse data for many rural counties.