# Geospatial Problem Solving

## Geospatial Data

Image from NASA: https://commons.wikimedia.org/wiki/File:Earth_Western_Hemisphere_transparent_background.png#filelinks

WV VIEW

AmericaView
Empowering Earth Observation Education
AmericaView.org - Est. 2003

Welcome to another lecture. Our goal here is to discuss the variety of spatial data types that are available. We will discuss both vector and raster data models and a variety of data formats.

We will begin with a brief review of datums and projections.

## Module Topics

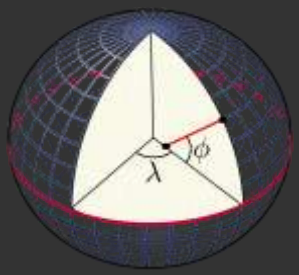
1. Levels of measurement
2. Vector data model
3. Topology
4. Georelational vs. object-based vector data file formats
5. Open-source vector data file formats
6. Raster data model
7. Pixel types and bit depth
8. Multiband raster grids
9. Cloud optimized and multidimensional raster data
10. Vector-to-raster and raster-to-vector conversion

2

These are the topics covered in this module.

## Coordinates on a Sphere

- Latitude and Longitude
- Specified as angles or degrees




https://commons.wikimedia.org/wiki/File:Latitude_and_longitude_graticule_on_an_ellipsoid.svg

Image from ArcGIS API for JavaScript

3

In order to use data within a map space, it must be referenced to the correct location on the surface of the Earth.

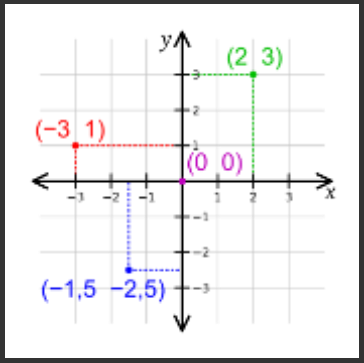
One option is to reference your data against the globe. In this case, coordinates will be defined as latitude and longitude coordinates on the spherical surface of the Earth. Further, these coordinates will be referenced against a specific ellipsoid/oblate spheroid model of the Earth or datum (for example, WGS84 or NAD83).

Models of the globe are known as Geographic Coordinate Systems. These systems are commonly used for global scale data or small scale maps.

GPS coordinates are also often referenced using latitude and longitude relative to the WGS84 datum.

## Coordinates on a Plane

- X, Y
- For projected coordinate systems
- Length or distance units relative to an origin




https://commons.wikimedia.org/wiki/File:Cartesian_coordinate_system_(comma).svg

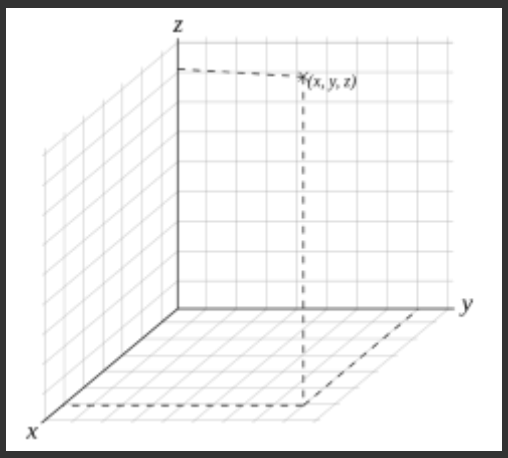
4

The surface of the Earth can also be projected to a flat map space as a map projection. These systems are known as Projected Coordinate Systems. Examples include: UTM Zones, State Plane Coordinate Systems, Web Mercator, North America Albers Equal Area Conic, and North America Lambert Conformal Conic.

When using projected coordinate systems, coordinates are no longer specified using angles. Instead, coordinates are defined using length units, such as meters, relative to an origin location. This is similar to defining coordinates on a graph as an X and Y coordinate pair since we are now making measurements in flat or Cartesian map space. Projected coordinate systems are commonly used for 2D map surfaces and especially for large scale maps.

In this course, we will commonly work within projected coordinate systems as opposed to geographic coordinate systems.

## Coordinates in 3-D Space




https://commons.wikimedia.org/wiki/File:Cartesian_with_grid.svg

- Can expand to 3D

5

Cartesian coordinate systems can also be projected into 3D space by adding a Z coordinate. This allows us to define a location in a 3D space or volume as opposed to a 2D space or plane. We will generally not make use of z-values in this course.

## Geographic Coordinates vs. Projected Coordinates

Image from ArcGIS Maps SDK for JavaScript

6

The data presented here were made available from the ArcGIS Maps SDK for JavaScript.

The image on the left represents the Earth as a globe. This is an example of a geographic coordinate system, where measures of location are made using angles as latitude and longitude.

The image on the right is an example of a projected coordinate system, where the surface of the Earth has been projected to a flat surface. Specifically, this is the Web Mercator Projection that is commonly used for web maps. Now, measurements are made in distance or length units relative to an origin.

If you have difficulties with the concepts of datums and projections, I recommend reviewing the Datums and Projections module.

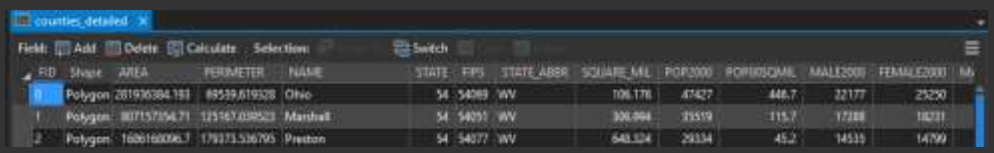
# Levels of Measurement

## Non-Spatial Data in GIS

Attributes = non-spatial characteristics associated with spatial data

Categorical
- Nominal = A unique identifier or unique types
- Ordinal = Ranked data

Numeric
- Interval = Difference between numbers is significant, but no fixed non-arbitrary zero point
- Ratio = Difference between numbers is significant, and there is a fixed non-arbitrary zero value




8

We will now discuss how data are classified. This is an important consideration, since different types of data are often visualized and analyzed differently. Data can be broken into two broad categories: categorical and numeric.

Categorical data represent discrete categories that are either unordered (nominal) or ordered (ordinal). Examples of nominal data include land cover types, soil types, or vegetation types while an example of ordinal data would be rankings such as 3rd place, 2nd place, and 1st place.

Numeric data represent continuous numbers or measures and can be further subdivided into interval and ratio types. For both types the difference between subsequent numbers is fixed; for example, the length difference between 2 meters and 3 meters is the same as the difference between 7 meters and 8 meters. However, they differ in regards to how the zero value is defined. For interval data, the zero value is arbitrary whereas this value is non-arbitrary for ratio data. Temperature in Fahrenheit or Celsius units is an example of interval data as zero temperature on these scales do not mean “no temperature.” Further, ratios cannot be calculated for interval data because, without a fixed zero value, measures that are twice as large do not indicate twice the magnitude. For example, 40° Celsius is not twice as hot as 20° Celsius. Examples of ratio data include lengths, areas, and percentages since the zero

value does have meaning or is non-arbitrary. For example, 0% unemployment indicates that no proportion of the working population is unemployed.

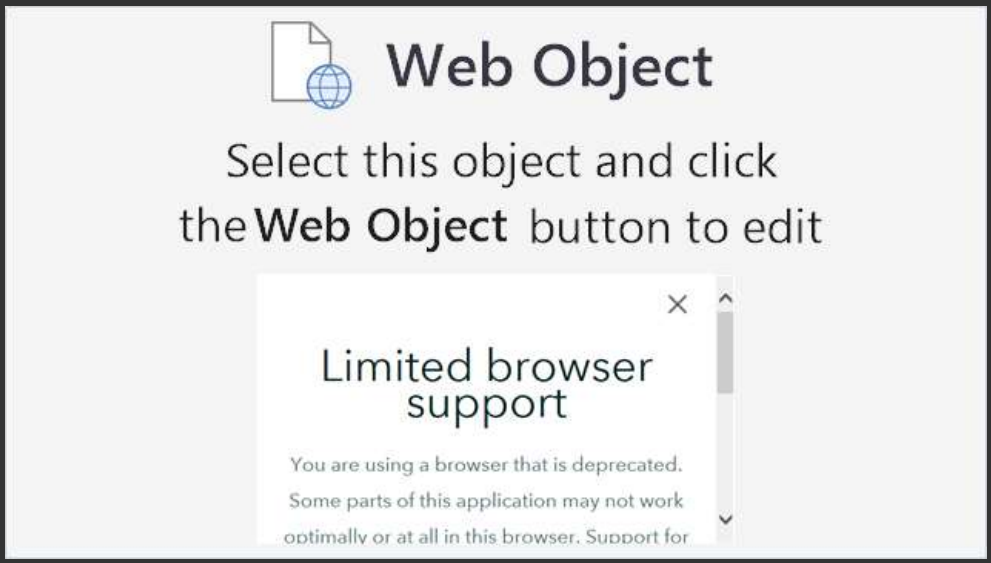
## Nominal Data

9

This slide provides some examples of nominal data as county names, county FIPS codes, and land cover categories. These are all unique categories with no implied order.

Note that it is possible to use numbers to represent categories. For example, FIPS, or Federal Information Processing Standard, codes are numbers. However, in this case they are not being used as numbers. Instead, they are being used to represent a specific feature. Another similar example would be ZIP codes.

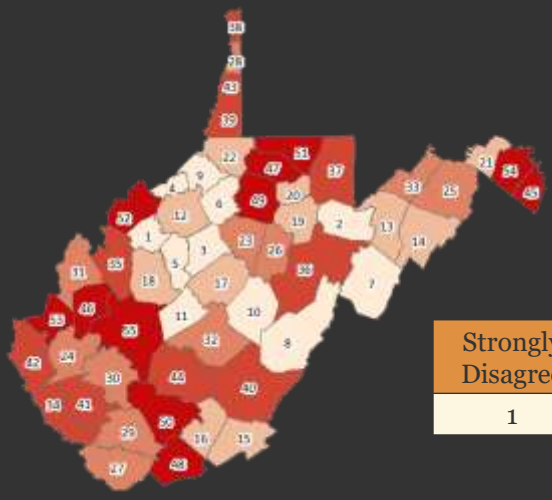
## Nominal Data




10

This web app shows the ZIP code boundaries within West Virginia. This is an example of numeric values that should be treated as nominal data.

## Ordinal Data



| Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

11

Ordinal data represent categories with an applied order, such as ranks.

Survey questions, such as Strongly Disagree to Strongly Agree (the Likert scale), are also ordinal data since there is an implied order in the groupings.

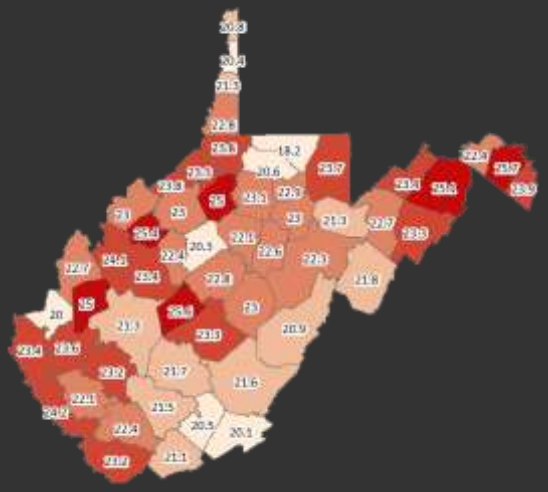
# Interval Data

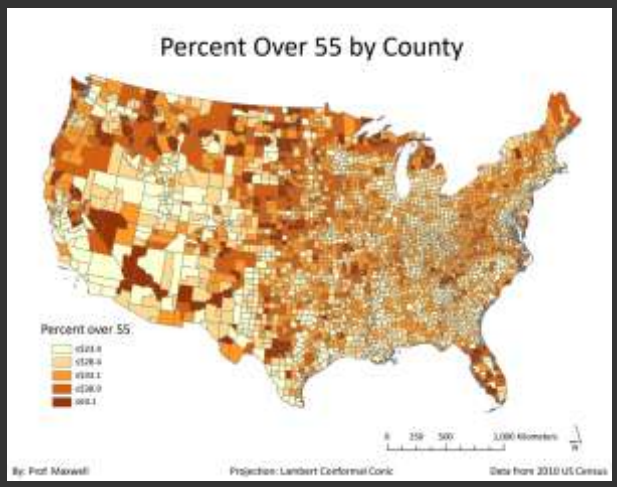

Image from the National Weather Service

12

Temperature in Fahrenheit or Celsius units and time are examples of interval data. The zero value is arbitrary and meaningful ratios cannot be calculated.

# Ratio Data

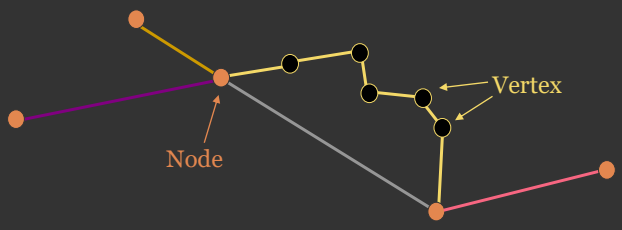


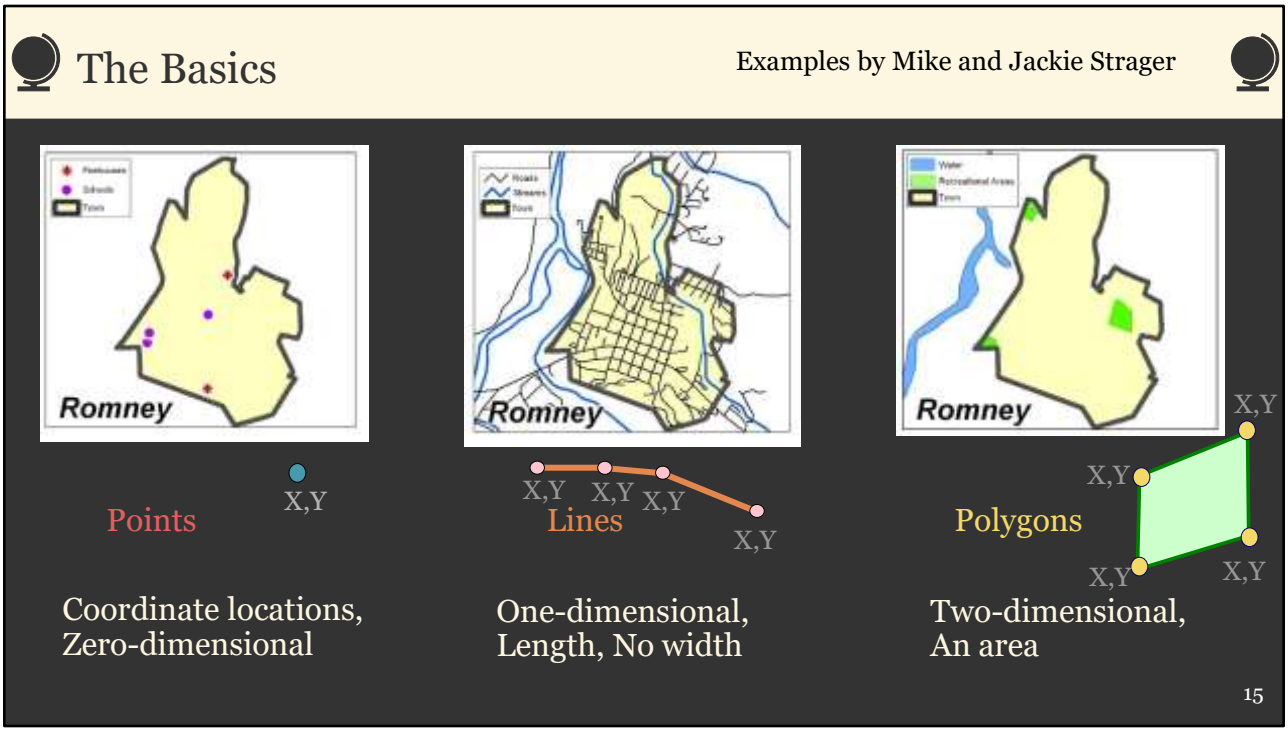



13

Percentages are examples of ratio data. The zero value is non-arbitrary and meaningful ratios can be calculated.

# Vector Data






Vector data can be represented as points, lines, or polygons.

Point data are simply coordinate pairs with associated attributes. They have no spatial dimensions, since they do not have length or width.

Lines are a series of coordinate pairs connected to form a line. They have one spatial dimension, since they have length but not width.

Polygons are a series of coordinates in which the first and last coordinate are connected to form a closed feature or area. They have both length and width, so have two spatial dimensions. Since they have length and width, they have an area.

## Geometric considerations

- Points
  - Only a coordinate
  - No length
  - No area
- Lines
  - Multiple coordinates (vertices)
  - Linked as a linear feature
  - Length
  - No area
- Polygon
  - A closed feature (area)
  - Has length (perimeter)
  - Has area

All types can store attribute information

| Parcel ID | Owner | Acres | Zoning | Building Value | Year Built |
|---|---|---|---|---|---|
| 08-001 | West, Peggy | 0.45 | R | $64,000 | 1954 |
| 08-002 | School District | 5.12 | Ex | - | 1972 |
| 08-003 | North, Tim | 1.33 | R | $72,000 | 1956 |
| 08-004 | 7-Eleven | 2.03 | C | $92,000 | 1995 |

Examples by Mike and Jackie Strager

16

This slide highlights some of the key attributes of the different types of vector data.

Again, points are just coordinates and have zero spatial dimensions.

Lines are a series of connected coordinates and have one spatial dimension (length).

Polygons represent areal features and have two spatial dimensions (length and width).

You can also calculate perimeter for polygon features. Perimeter is the length around the polygon, or the sum of the length of all the lines that make it up.

So, vector objects are made up of coordinates. Coordinates can be related to form more complex features as lines or polygons.

The coordinates are defined relative to a geographic or projected coordinate system so that the object draws at the correct location on a map.

One powerful aspect of a GIS and geospatial data is the ability to relate spatial and tabulated data. Vector data are not just map coordinates. These features

are also associated with tabulated information. For example, the parcel's boundaries shown on this slide are attributed with a set of information. This allows us to relate spatial and tabulated data.

## Need to Generalize

### Polygon Inclusions

17

The real world is infinitely complex. We can't map every blade of grass or every atom in the universe. So, all geospatial data represent generalizations of reality or real-world features. Different techniques are available to make these generalizations.

For example, polygon inclusion allows us to group a variety of features into a single object. In the example above, the rectangle represents a portion of the National Mall that includes a variety of additional features, such as buildings, roads, sidewalks, and trees. However, we are generalizing all of these features into a single object.

## Need to Generalize

### Boundary Generalization

18

Boundary generalization is another way to simplify real world features.

It is generally not possible to map or draw every knock and cranny or the full complexity of a geographic boundary. Instead, we must generalize the feature as a polygon made up of a set of line segments.

In the example, this island has a fairly complex boundary that is being generalized as a polygon features that describes the general shape of the island.

## Minimal Mapping Unit (MMU)

- Smallest area/feature to be mapped

Images from Google Earth

19

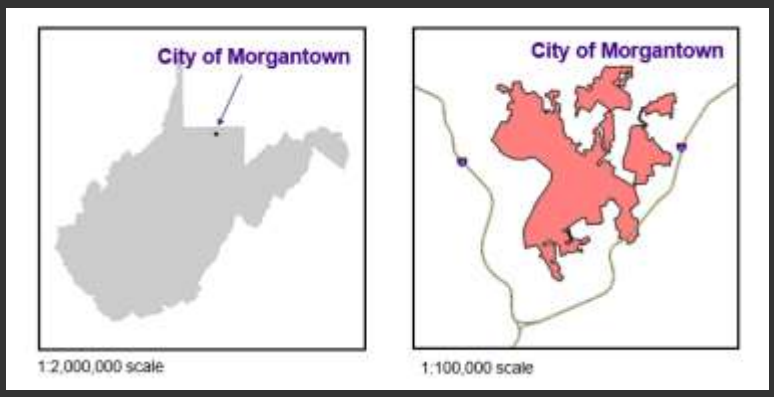
One concept relating to generalization is the minimal mapping unit or MMU. This is the smallest object or feature to be mapped. It can be defined based on a minimal area. For example, we won't map stands of trees that are smaller than 2 hectares. Alternatively, it can be defined more qualitatively. For example, we will not map any structures smaller than a single-family home. So, dog houses or storage sheds won't be mapped.

When undertaking a mapping project, it is generally a good idea to define a minimal mapping unit. This will allow you to avoid mapping feature that are deemed unimportant or too small to consider. Additionally, defining an MMU can make it clear to the client what will be mapped and what type of result will be produced.

In short, I highly recommend defining an MMU in a mapping project. This can make the product more clear and well defined.

In the image to the left, you could define a qualitative MMU as nothing smaller than a single-family home. In the second image, you could define a size cut-off for lakes to digitize or map.

## Point, Line, or polygon?




Example by Mike and Jackie Strager

20

How a geographic feature is represented depends on the nature of the features itself, the scale of the map, and the level of generalization desired.

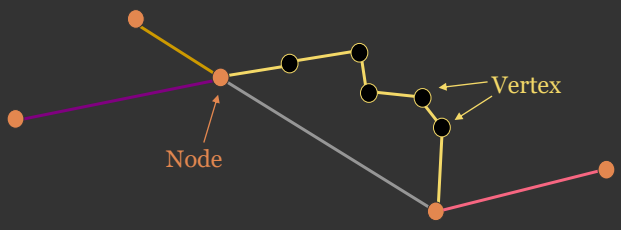
For example, certain features make sense to represent as points, such as telephone poles or fire hydrants. Some features make sense to map as lines, such as roads or rivers.

The optimal feature type partially depends on scale. For example, in the example provided here, Morgantown is represented as a point when zoomed out or on a smaller scale, statewide map. On a larger-scale map or when zoomed in, it makes more sense to represent the city as a polygon extent.

So, you should think about the best way to represent your data based on the nature of the feature itself, the scale of the map, and the level of generalization required.

## Vector Data Structure

- Vector data are made up of connected vertices.
- End vertices are called nodes.
- Vertices can be connected as lines and polygons.




A computer uses relationships between features to build geometric features that are spatially referenced.

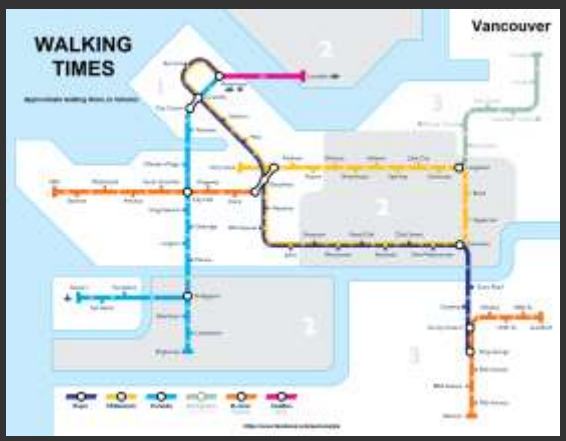
21

As mentioned above, all vector data are made up of coordinates. Points are individual coordinates while lines and polygons are made up of interrelated coordinates.

These individual coordinates are called vertices. Vertices that occur at or define the end of a line segment are called nodes.

Again, all vector data are made up of coordinate points.

## Topology




http://www.insidevancouver.ca/2016/04/04/vancouver-map-showing-walking-times-between-skytrain-stations-and-b-line-stops/

*Study of properties of geometric objects that remain invariant under certain transformations such as bending or stretching* (Massey, 1967)

Examples:

- Adjacent to
- Incident with

Topology and Topography are different!

22

An important concept associated with vector data is the concept of topology. This is not to be confused with topography.

Topology is the study of properties of geometric objects, including map features, that remain invariant, or don't change, under certain transformations such as bending or stretching.

Example topological relationships include adjacent to and incident with. For example, if the US state boundaries could be bent or stretched, the adjacency between states would be maintained. In other words, adjacent states would still be next to each other and share boundaries. If map features were bent or stretched, features that occur within or are incident with other features would maintain this relationship with those other features. For example, cities would still fall within the correct state or country.

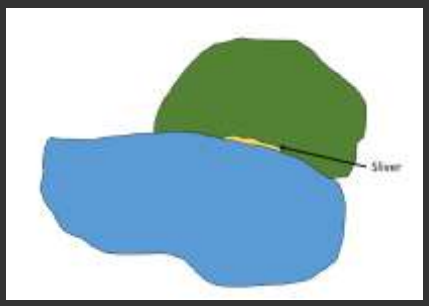
A good example of topology is subway or public transit maps. On this slide, I have included the subway map for the city of Vancouver in British Columbia, Canada. The subway lines and other map features have been generalized such that the full complexity is not represented. However, important topological relationships have been maintained. For example, you know which stops occur along or are incident with which lines. You also know where different lines cross, or where you can transfer between lines. So, even though these data have

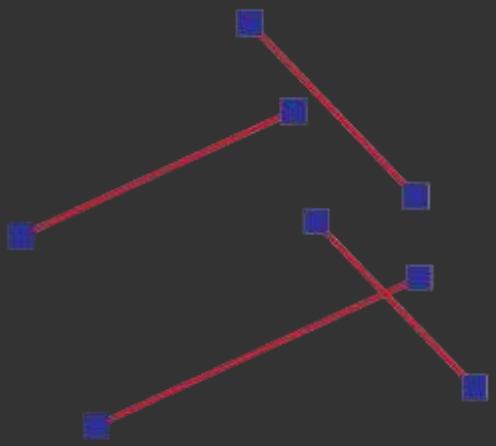
been generalized, important topological rules have been maintained so that you can use the map to navigate the subway system.

Topology is important because it allows us to solve problems that require an understanding of the spatial relationship between map features.

## Topological Issues

1. Dangles
2. Overshoots
3. Slivers



23

When creating new data, issues can arise that cause topological relationships to not be maintained.

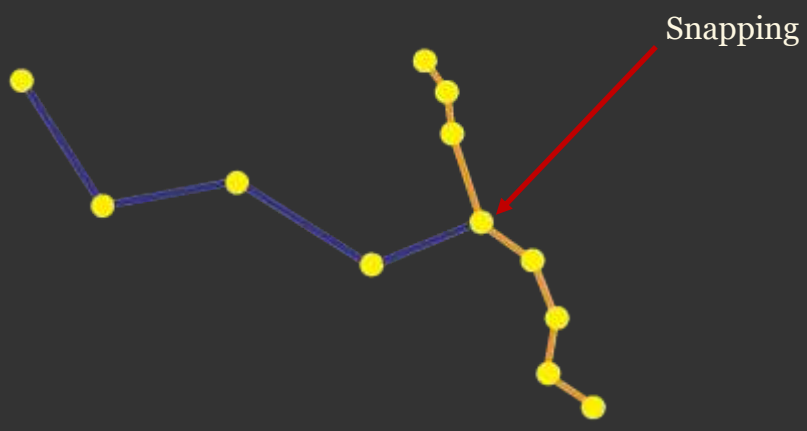
Dangles occur when 2 line features are supposed to connect, such as at a road intersection, but they fall short. If you were trying to model movement along a road network, this would be an issue.

Overshoots occur when one line segment is supposed to end when it intersects another segment, but it continues or overshoots.

Slivers represent areas of non-overlap between features that could represent real patterns or result for digitization errors or scale difference in the input data.

## Snapping

- Causes/forces vertices to match up




24

When drawing or digitizing new data you can use snapping to enforce topological rules or relationships.

When snapping is enabled, if a feature comes within a defined distance or tolerance of another feature, it will automatically be connected to this feature.

## Types of Topology

- Spaghetti Model = Early model for lines that did not model topology
- Planar Topology = All features exist on a plane. If features cross, they must share the same space or intersect
- Non-Planar Topology = Allow for overlap. It is not assumed that features that cross share the same space

25

As our ability to model spatial data within a computer has advanced, our ability to model topological relationships has evolved.

For example, the spaghetti model was an early vector model for lines that allowed computers to represent linear features. However, it did not allow for topological relationships to be modeled. So, if two lines crossed there was no way to model this intersection. So, these models were of limited value in some spatial analyses, such as finding directions along road networks.

Planar topology generally allows for topology to be modeled, but it is assumed that all features occur on the same plane, or that there is no third dimension or elevation. So, features that cross must intersect or share a space. This can be limiting, as this is not always true in the real world. For example, 2 roads may overlap at a location, but that doesn't mean that they are connected. Examples include a bridge passing over a road or a tunnel passing under a road.

This issue is alleviated by using non-planar topology. With non-planar topology it is not assumed that all features exist on the same plane. In other words, there is a vertical component.

## Topology for Points, Lines, and Polygons

- **Points**
  - Simple, independent features
- **Lines**
  - Substantial structure, start node, end node, direction
- **Polygons**
  - Nodes and vertices that make up the line, connected lines, start and end nodes connect

26

Points are generally topologically simple, as they are independent features.

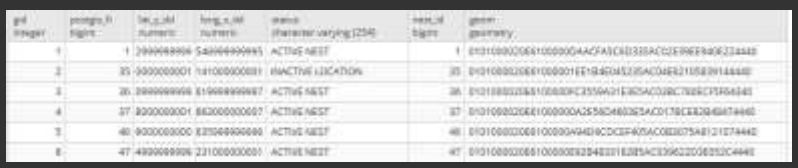
Lines are more complex, as vertices are interconnected to form a single feature. There is generally a start node, end node, and internal structure defined by additional vertices. It is also possible to calculate a direction, as a compass bearing, for the line.

Polygons are also complex, as they are made up of interconnected lines and the start and end node of the polygon must connect to close the feature.

Note that it is also possible to model topological relationships between vector features.

## Vector Data Models

- Georelational = stores geometries and attributes separately
  - Coverage
  - Shapefile
- Object-Based Data Model = treats geospatial data as objects
  - Geodatabase



27

Vector data models can be divided into two broad types: georelational and object-based.

A georelational model stores the geometry or spatial information in a separate file from the aspatial or attribute information. Examples include coverages and shapefiles.

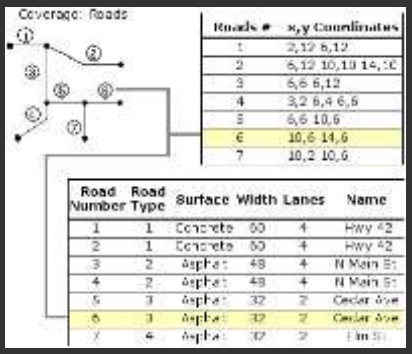
In contrast, object-based models store the geometry or spatial information in the same file as the attribute information. Generally, the spatial or geometric information is stored in a special field in the table along with the attributes. Geodatabases are an example.

We will now explore some example file structures.

# ESRI Vector File Formats

## Coverage Files

- Early vector data model by ESRI (1980s)
- Can model topology
- Connectivity = Arcs connect at nodes
- Area Definition = Area is defined by a series of connected arcs
- Contiguity = Arcs have directions and left and right polygons




http://desktop.arcgis.com/en/arcmap/10.3/manage-data/coverages/what-is-a-coverage.htm

29

Coverage files are georelational, were introduced by ESRI in the 1980s, and provide a means to store vector features and topological relationships.

Arcs, or lines, are defined by a series of vertices. The end vertices are called nodes.

Areas, or polygons, are defined by a series of connected arcs, or lines.

Contiguity was defined by recording the direction of arcs and polygons that are to the left and right of each arc.

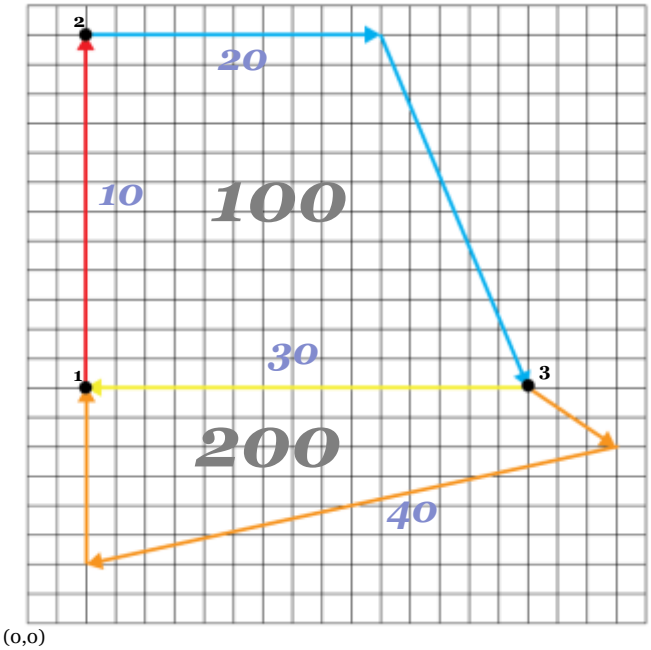
This information is stored in a table structure.

Coverage files are now antiquated and are rarely used.

| Arc # | F-Node | T-Node |
|---|---|---|
| 10 | 1 | 2 |
| 20 | 2 | 3 |
| 30 | 3 | 1 |
| 40 | 3 | 1 |

| Arc # | X, Y Coordinates |
|---|---|
| 10 | (2, 8) (2, 20) |
| 20 | (2, 20) (12, 20) (17, 8) |
| 30 | (17, 8) (2,8) |
| 40 | (17, 8) (20, 6) (2, 2) (2, 8) |

| Polygon # | Arc # |
|---|---|
| 100 | 10, 20, 30 |
| 200 | 30, 40 |




30

This slide describes the structure of a coverage file.

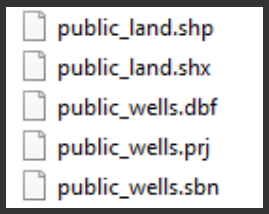
Arcs are defined by an F or From node and a T or To node, as shown in the top table The internal structure of each arc is defined by vertices, as shown in the middle table.

Polygons are defined base on the arcs that make them up, as shown in the bottom table.

A table is also generated to note what polygon is to the left and right of each arc. That table is not shown here.

## The Shapefile

- Nontopological
- No files describe the spatial relationships among geographic objects




.shp = geometry data

.prj = spatial reference/projection information

.dbf = attribute table/aspatial data

.shx = index file for geometry (helps with speed)

.sbn and .sbx = index file for features

.xml = metadata

31

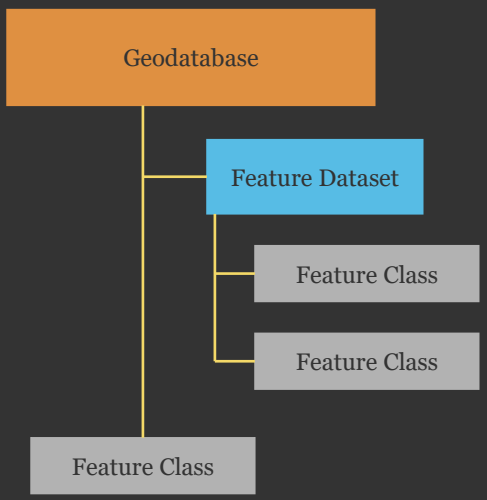
A shapefile is a vector geospatial data format that uses a georelational structure and is still commonly used. It was originally developed by Esri and is now openly documented, allowing it to be used across many GIS software packages.

A shapefile is a nontopological format, meaning that it stores feature geometries but does not explicitly store spatial relationships such as adjacency, connectivity, or shared boundaries. Because of this relatively simple structure, shapefiles can be efficient for display and data exchange. Although topology is not stored in the shapefile itself, GIS software can calculate spatial relationships on-the-fly to support analysis.

A shapefile is actually made up of multiple associated files. The .shp file stores the feature geometry, the .shx file stores an index of the geometry, and the .dbf file stores the attribute table. Other supporting files, such as .prj, may also be included to define the coordinate reference system.

## The Geodatabase

- Stores data in a relational database structure
- Object-based vector model
- Stores geometries and attributes in a single system
- Geometry is stored in a BLOB (binary large object) field
- Vector data are stored in a geodatabase as a feature class
- Can model topological rules on-the-fly
- Allows for topological relationships between feature classes in a feature dataset




32

Geodatabases are Esri's modern data storage model for managing spatial and nonspatial data. In a geodatabase, vector data are stored as feature classes. A feature class is a collection of geographic features that share the same geometry type, such as points, lines, or polygons, and the same attribute structure.

Feature classes can be stored on their own in a geodatabase or grouped within a feature dataset. A feature dataset can be thought of as a container inside a geodatabase that stores related feature classes with the same coordinate system. Feature datasets are often used when multiple feature classes need to participate in shared spatial behavior, such as topology rules, network datasets, or other geodatabase relationships.

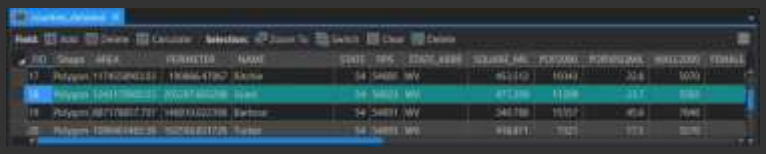
The geodatabase uses an object-based data model. In a feature class, each feature is stored as a row in a table. Attribute values are stored in fields, and the feature geometry is stored in a special geometry field. This differs from the shapefile format, where geometry, attributes, and indexes are stored in separate associated files.

Geodatabases can support more advanced spatial behavior than shapefiles. For example, topology rules can be created to model and validate spatial relationships, such as requiring polygons not to overlap or lines to connect

properly. GIS software can also calculate spatial relationships on-the-fly during analysis, but geodatabase topology allows some of these relationships and rules to be formally defined and checked within the dataset.

Since this is a more modern and flexible method for storing spatial data, we will use geodatabases frequently in this course and in the lab exercises.

## Attribute Tables



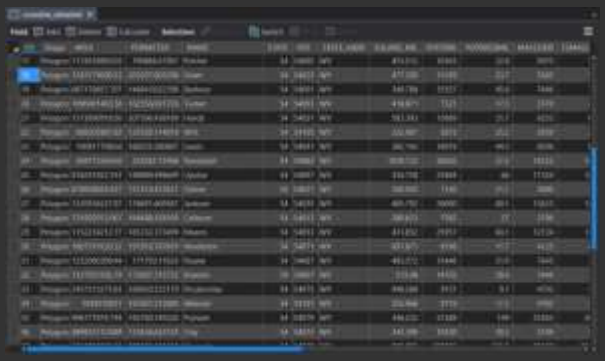
- Commonly one-to-one
- One row entry for each geometric feature (point, line, or polygon)
- Stored in .dbf component of the shapefile

33

As mentioned above, all vector objects have geometric or spatial information and aspatial or tabulated information. The tabulated information is associated with the spatial feature as a one-to-one relationship: each geographic feature has a single row entry associated with it.

As an example, in a shapefile, one row in the .dbf file is associated with a single geographic feature. In the provided example, one record is provided for each county.

## Attribute Tables



- Columns = fields or attributes
- Info stored for each record
- Rows = Records
- One record for each vector feature

- All shapefiles and feature classes have an attribute table
- Some categorical raster data can also have attribute tables

34

Attribute or aspatial information is stored in the attribute table. All shapefiles and feature classes have attribute tables. It is also common for categorical raster grids to have an attribute table. We will discuss this later in this module.

In GIS, columns are commonly referred to as fields or attributes while rows are called records. Each row or record in the table stores all the aspatial data associated with one map feature, such as one county. Each column stores a specific piece of information for each feature, such as the county population.

You will learn to work with, create, and query attribute information in later modules.

## Joining Aspatial Data and Spatial Data

### Join

- Non-spatial data can be joined to spatial data as long as there is a common field between the data.
- Later, the joined fields can be used for analysis or to symbolize the features.

### Relate

- New fields are not added to table. However, a relationship is established.


35

It is possible to associate additional tabulated data with spatial features using a join or relate.

When performing a join, additional attributes are associated with geographic features and are added to the attribute table as a temporary association. This requires that a common field be available between the two datasets.

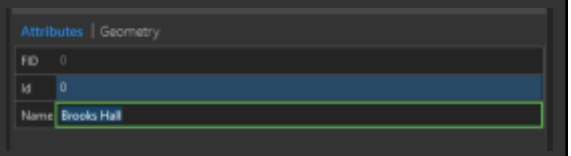
For example, additional information for each county could be joined to the county boundaries if both features have a common county name or FIPS code attribute.

If you would like to permanently associate the new attributes, you can export a copy of the data layer.

A relate is similar to a join, except that the related data are not temporarily added to the attribute table. Instead, the data are related but maintained in separate tables.

## Adding Fields

- Make sure to define the correct type




| Data Type | Description |
|---|---|
| Short Integer | Integer values between -32,768 and 32,767 |
| Long Integer | Integer values between -2,147,483,648 and 2,147,483,647 |
| Float | Decimal values with 1-6 decimal places |
| Double | Decimal values with more the 6 decimal places |
| Text | Text strings |
| Date | Data info (mm/dd/yyyy) |

36

When creating a new field, you will need to define the field type.

Short and long integer fields can only store integer or whole number values. Long integers can store a wider range of numbers than short integers.

If you need to maintain decimal values, you cannot use a short or long integer. Instead, you must use a float or double type. Float fields can store up to 6 decimal places while double can store more than 6.

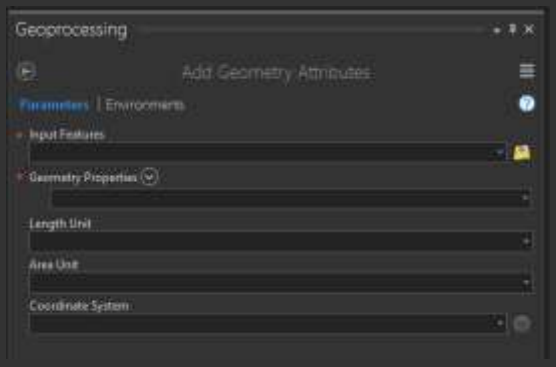
If you would like to store text or string data, you will do so in a text field. Note that numbers stored in text fields will be treated like strings. They will not be treated like numbers. For example, you will not be able to perform mathematical operations on these data.

Date fields store dates and will treat the information as dates.

## Add Geometric Fields

- Use Add Geometry Attributes Tool

| Points | Lines | Polygons |
|---|---|---|
| X-coordinate | Start-X | Area |
| Y-coordinate | Start-Y | Perimeter |
| Z-coordinate (if z-enabled) | End-X | Centroid-X |
| M-coordinate (if m-enabled) | End-Y | Centroid-Y |
| | Mid-X | |
| | Mid-Y | |
| | Length | |
| | Bearing | |




37

Different geometric measures can be calculated and stored in the attribute table for points, lines, and polygon data.

For point data, it is possible to store the x and y coordinate. If the feature is Z-enabled, or contains elevation or height information, this value can also be calculated. If it is M-enable, the M coordinate can be calculated. M-coordinates generally relate to positions along a network.

For lines, you can calculate the x and y coordinates for the start, end, and middle of the line segment. You can also calculate the length and bearing. The bearing is the compass direction of the line and is defined as the direction of travel from the start to the end node.

For polygons, you can calculate the area and perimeter. The perimeter is the length of all the lines that make up the polygon. You can also calculate the centroid, or geometric center, coordinates. Note that centroid coordinates can sometimes be misleading. For example, for irregular or elongated shapes, the centroid may not actually occur within the polygon. For a “C” shaped feature, the centroid would occur somewhere in the middle of the “C” as opposed to within the area that makes it up. So, be careful when using centroid measures.

## Calculating Area of a Polygon

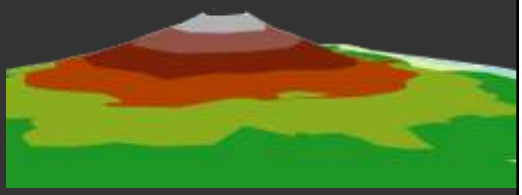
- Area is generally calculated in 2-D space
- This does not take into account topography and the 3D lay of the land.
- This can be misleading in complex terrains
- However, it is computationally much simpler and requires less data

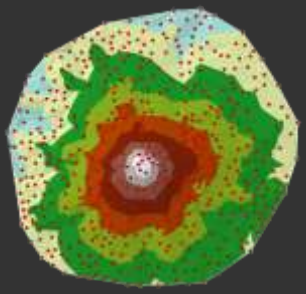
38

As a note, the 3D nature of the topography is generally not considered when calculating areas for polygons. Instead, area is calculated on a plane. So, area measurements can be misleading in complex or variable terrain. Although this can be misleading, it is generally computationally simpler. There are more complex means to calculate area that take into account the 3D nature of the terrain.

## Vector Data, Not for Discrete Features

- TINs (Triangulated Irregular Network)
- Good for representing terrain data
- A space filling model



39

Vector data are generally used to map discrete objects as opposed to continuous objects or surfaces. One exception are TINs or Triangulated Irregular Networks.

TINs provide a means to represent continuous phenomena using a vector model. They are often used to represent terrain or elevation surfaces. However, other continuous variables can be modeled.

## TINs

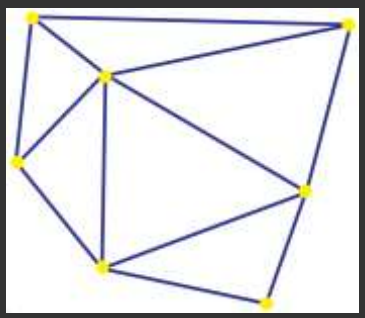
Connect measurement points to form a triangle (face)

Can also include contours and/or breaklines

Lines connecting points = edges

Can be used to represent a terrain (or other continuous surfaces)

Pros and cons compared to raster data



40

TINs are generally produced from mass points or discrete measurements.

Near points are connected to form triangular faces or facets. The lines connecting the points are called edges.

The triangles then hold a measure derived from the connected points.

If the phenomenon of interest changes abruptly, breaklines can also be included. Breaklines represent breaks in the surface. For example, road cuts could be modeled as breaklines in a terrain surface.

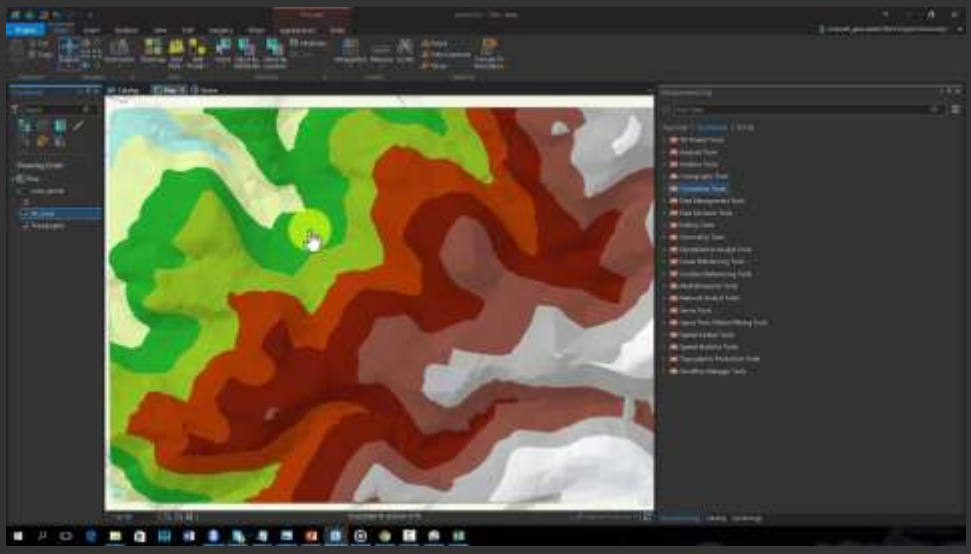
There are pros and cons when representing a continuous surface as a TIN vs. a raster.

TINs are inherently multiresolution. Smaller triangles or more detailed surfaces will be produced in areas with more point measurements whereas larger triangles or less detailed surfaces will be produced in areas with less point measurements. In contrast, a raster-based surface most have the same cell size across the entire extent. TINs can also be smaller files in comparison to raster grids, though this depends on many factors.

Many of the tools available for performing terrain analysis or overlay are

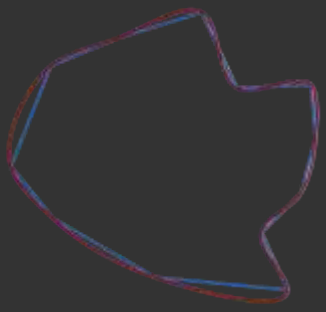
designed for raster data. So, it might be necessary to convert the TIN to a raster to undertake these analyses.

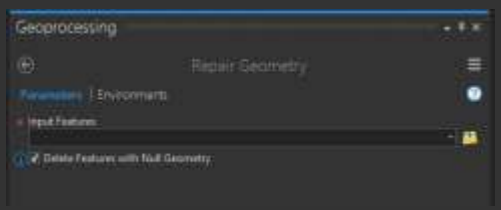
# Video: TINs



## Important geometric operations for vector data

- Repairing geometry
- Smoothing/generalization
- Merging/splitting
- Etc.






42

Many operations are available to alter the geometry of vector data.

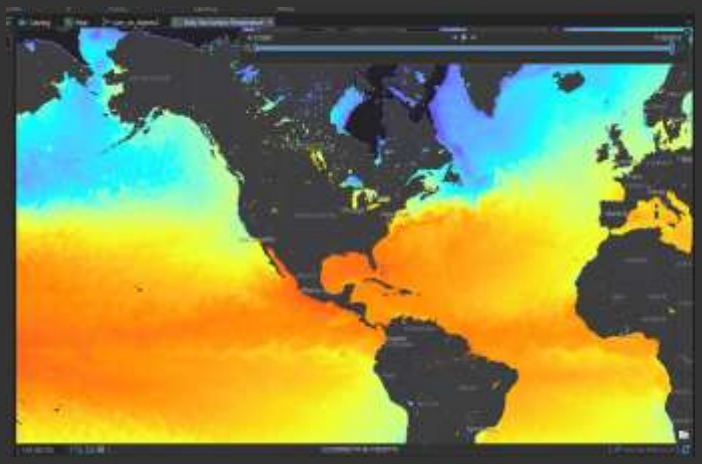
Sometimes the geometry may be invalid due to digitizing errors or other issues. These errors can be flagged and corrected using Repair Geometry.

There are methods available to generalize or smooth vector data. These are often used to enhance or improve cartographic display.

It is also possible to merge and split data.

## Time Enabled Data

- If data are time enabled, you can use time information to analyze, display, and query the data



Data from Naval Oceanographic Office and made available in the ESRI Living Atlas

43

Time enabled data includes date or time information.

With time enable data, it is possible to update the display relative to a certain time period.

It is also possible to perform analyses that require multi-temporal data and query the features based on time.

# Other Vector Data Models

## GeoJSON

- JSON = JavaScript Object Notion
- Used to store and exchange data
- Text written with JavaScript object notation
- GeoJSON = allows you to encode geographic structures
- Used in web applications
- One file can store multiple geometry types


http://geojson.org/

45

In this section we will discuss some open-source formats that can be used to store vector data.

JSON stands for JavaScript Object Notation. JavaScript is capable of interacting with data provided in this format. Since JavaScript is the primary client-side web programming language, JSON is often used to represent data on the web or transfer data between the client and server.

GeoJSON expands JSON to include geographic features and spatial reference information. This is one means to work with spatial data on the web. In contrast to the ESRI models discussed above, a GeoJSON file can store multiple geometry types (points, lines, and polygons) in the same file.

The next set of slides will provide some examples.

## GeoJSON: Point


```
{
"type": "FeatureCollection",
"name": "json_point2",
"crs": { "type": "name", "properties": {
"name": "urn:ogc:def:crs:OGC:1.3:CRS84" }
},
"features": [
{ "type": "Feature", "properties": { "id": 1,
"attribute1": 1, "attribute2": "point1",
"attribute3": 0.1111 }, "geometry": { "type":
"Point", "coordinates": [ -96.437, 37.283 ] } }
]
}
```


46

The text on this page represents the GeoJSON code for the point shown on the map.

This file contains a feature type, a file name, a coordinate reference system, or crn, definition, a list of attribute names and values, and the x and y coordinate representing the point.

This structure makes use of objects and arrays to defines this geographic feature as text.

## GeoJSON: Point


```
{
"type": "FeatureCollection",
"name": "example_point",
"crs": { "type": "name", "properties": { "name":
"urn:ogc:def:crs:OGC:1.3:CRS84" } },
"features": [
{ "type": "Feature", "properties": { "id": 1, "attribute1": 1,
"attribute2": "point1", "attribute3": 0.1111 }, "geometry": { "type":
"Point", "coordinates": [ -96.437, 37.283 ] } },
{ "type": "Feature", "properties": { "id": 2, "attribute1": 2,
"attribute2": "point2", "attribute3": 0.22222 }, "geometry": {
"type": "Point", "coordinates": [ -83.006, 39.913 ] } },
{ "type": "Feature", "properties": { "id": 3, "attribute1": 3,
"attribute2": "point3", "attribute3": 0.33333 }, "geometry": {
"type": "Point", "coordinates": [ -115.666, 43.247 ] } },
{ "type": "Feature", "properties": { "id": null, "attribute1": null,
"attribute2": null, "attribute3": null }, "geometry": { "type":
"Point", "coordinates": [ -78.691, 40.501 ] } }
]
}
```


47

This example is similar to the previous one except that this file now contains four points. For each point, attributes and x and y coordinates are provided.

## GeoJSON: Line


```
{
"type": "FeatureCollection",
"name": "line_example",
"crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
"features": [
{ "type": "Feature", "properties": { "id": 1, "attribute1": 1, "attribute2": 1111, "attribute3": "line" }, "geometry": {
"type": "MultiLineString", "coordinates": [ [ [ -118.903, 39.471 ], [ -112.086, 41.188 ], [ -105.957, 42.315 ], [ -
97.718, 41.237], [ -90.313, 39.030], [ -85.458, 36.234], [ -78.544, 38.883 ] ] ] } }
]
}
```


48

This example represents GeoJSON code for a line. This is pretty similar to the code for point features, except that the line is now represented as a series of x and y coordinate pairs as opposed to a single coordinate. The attribute definition is the same as that for the point.

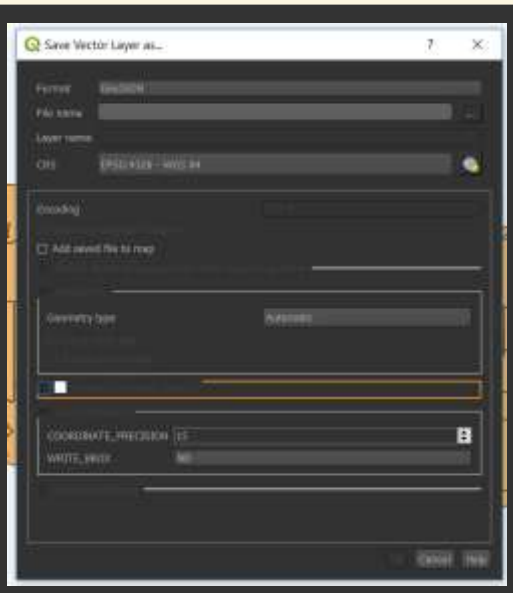
## GeoJSON: Polygon


```
{
"type": "FeatureCollection",
"name": "polygon_example3",
"crs": { "type": "name", "properties": { "name":
"urn:ogc:def:crs:OGC:1.3:CRS84" } },
"features": [
{ "type": "Feature", "properties": { "id": 1 },
"geometry": { "type": "MultiPolygon",
"coordinates": [ [ [ [ -101.2, 43.051 ], [ -99.238,
45.111 ], [ -99.238, 45.111 ], [ -93.697, 46.239 ],
[ -89.332, 42.463 ], [ -88.254, 36.333 ], [ -
92.275, 31.429 ], [ -100.219, 32.312 ], [ -
107.183, 36.235 ], [ -105.809, 42.07 ], [ -101.2,
43.051 ] ], [ [ -99.581, 40.354 ], [ -98.503,
34.371 ], [ -93.501, 36.823 ], [ -92.128, 39.373
], [ -95.364, 41.923 ], [ -99.581, 40.354 ] ] ] ] }
}
]
}
```


49

This final example is the result for a polygon feature. Similar to a line feature, this polygon is defined based on a set of coordinate pairs. The first and last coordinate is identical so that the feature is closed. This specific feature has an internal ring representing a hole. This hole is defined by coordinate pairs also.

## GeoJSON in QGIS



- Can export files to GeoJSON format
- Can create bounding box
- Can set coordinate precision
- QGIS can read an write many file types

50

GeoJSON files are directly readable in ArcGIS Pro and QGIS. Data can be read in, and other data types can be saved to GeoJSON. You can specify a bounding box as an extent and also the coordinate precision.

One benefit of QGIS is that it can read in and convert a wide variety of data types. I often use QGIS as a data conversion tool for this reason, and I specifically use it to convert to GeoJSON format.

## SpatiaLite


https://www.gaia-gis.it/fossil/libspatialite/index

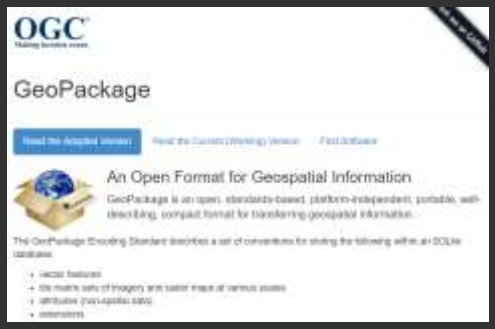
- Extension for SQLite
- Stores data in a file-based database
- Not a true client-server database
- Alternative to ESRI file geodatabase
- Only for vector data

51

SpatiaLite files are similar to file geodatabases in that they make use of a file-based database structure. File-based databases do not make use of true client-server database models. Currently, SpatiaLite files can only store vector data. Raster data cannot be stored.

## GeoPackage

- Developed by the Open Geospatial Consortium
- Store data in an SQLite database
- File-based database structure
- May see more support in the future
- Supports vector and raster storage
- Store tiled raster layers
- Lightweight



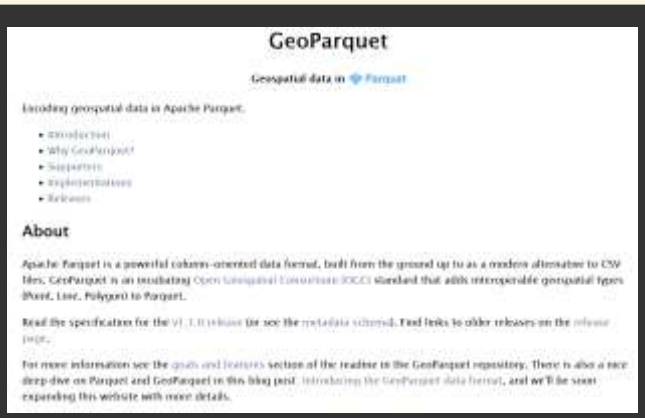

https://www.geopackage.org/

52

Similar to SpatiaLite and ESRI file geodatabases, GeoPackages make use of a file-based database structure. They build upon SQLite databases. In contrast to SpatiaLite, they can store vector and raster data. They are also lightweight.

I expect to see more use of GeoPackages in the future. We use this type within our QGIS examples.

## GeoParquet



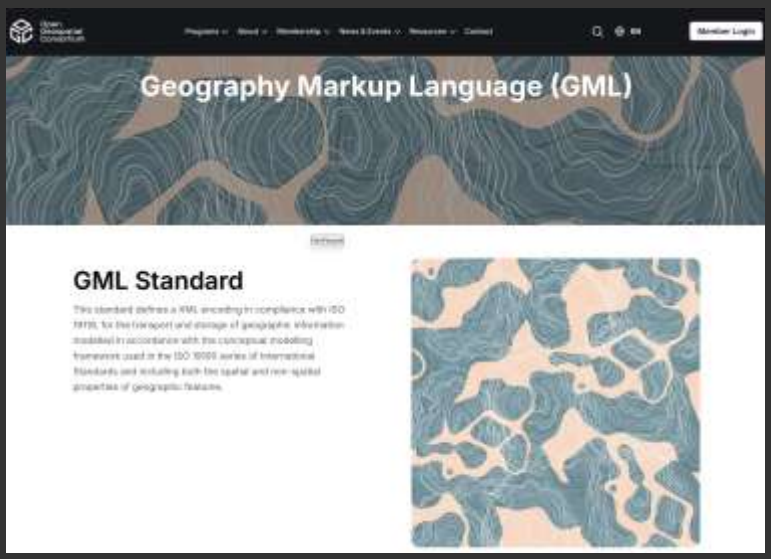

- Builds on Apache Parquet
- An Open Geospatial Consortium (OGC) standard
- Columnar data for vector geospatial data

https://geoparquet.org/

53

GeoParquet is an open, column-oriented geospatial data format that extends Apache Parquet for storing vector spatial data. Like standard Parquet, it is designed for efficient storage, compression, and querying of tabular data, making it useful for large datasets and cloud-based workflows. GeoParquet adds support for geospatial information by storing geometry columns, such as points, lines, and polygons, along with metadata that describes the geometry column, coordinate reference system, geometry types, and spatial extent. This allows both the attribute data and spatial geometry to be stored in a single efficient file while remaining interoperable across different geospatial and data science tools. Compared with older formats such as shapefiles, GeoParquet is better suited for large-scale analytics, modern data pipelines, and cloud-native geospatial workflows.

## Geography Markup Language (GML)




- eXtensible Markup Lanuage (XML) encoding for ISO 19118 for geographic data

https://www.ogc.org/standards/gml/

54

Geography Markup Language, or GML, is an open Extensible Markup Language- (XML-) based format used to describe and exchange geographic information. It was developed by the Open Geospatial Consortium and provides a standardized way to represent spatial features, including their geometry, attributes, coordinate reference system, and relationships. Because GML is based on XML, it is both human-readable and machine-readable, and it can be used to share geospatial data between different software systems. GML can represent points, lines, polygons, coordinate systems, and more complex geographic objects, making it more flexible than simpler vector formats. However, GML files can also be large and verbose, so they are often used for data exchange and web services rather than for fast display or compact storage.

## PostGIS

- Allows spatial data to be stored in a PostgreSQL database
- True client-server infrastructure
- Spatial data are stored as tables
- Geometry is stored in a column in the table
- Open and free to use
- Can be ran as a local host


https://postgis.net/

55

PostGIS is an extension for the free and open-source PostgreSQL database software. In contrast to SpatiaLite, GeoPackages, and ESRI file geodatabases, PostGIS makes use of true client-server architecture, though it can also be set up to run as a local host. This is another example of a georelational model where spatial and tabulated data are stored together in tables.

Although more complex to set up and use than file-based geodatabases, PostgreSQL and PostGIS can be a powerful solution for large-scale data management and allowing access to data through web apps.

Full use of PostgreSQL and PostGIS will require knowledge of structured query language (SQL).

## Note on PostGIS Geometry Types

- POINT = each point treated as a single feature
- MULTIPOINT = multiple points treated as a single feature
- LINESTRING = each line treated as a separate feature
- MULTILINESTRING = multiple lines treated as a single feature (do not need to be connected)
- POLYGON = each polygon treated as a single feature
- MULTIPOLYGON = multiple polygons treated as a single feature (do not need to share boundaries)
- GEOMETRYCOLLECTION = store multiple geometry types in the same layer

56

In PostGIS, vector spatial data are stored using geometry types. The most common geometry types are POINT, LINESTRING, and POLYGON, which represent individual locations, linear features, and area features, respectively. PostGIS also supports multi-part versions of these geometries, including MULTIPOINT, MULTILINESTRING, and MULTIPOLYGON, for features made up of multiple separate parts. A GEOMETRYCOLLECTION can store a mixture of different geometry types in a single feature. PostGIS also supports more advanced geometry types, such as curved geometries and 3D surface types, but most common GIS datasets use points, lines, polygons, or their multi-part equivalents.

# GPX

- GPS Exchange Format
- Uses Extensible Markup Language (XML)
- Open data format
- Store waypoints, tracks, and routes

```
<?xml version="1.0"?>
<gpx version="1.1" creator="GDAL 2.2.4" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ogr="http://osgeo.org/gdal"
xmlns="http://www.topografix.com/GPX/1/1" xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd">
<metadata><bounds minlat="37.283000000000001" minlon="-115.665999999999997" maxlat="43.247000000000000" maxlon="-
78.691000000000003"/></metadata>
<wpt lat="37.283" lon="-96.437">
 <extensions>
  <ogr:id>1</ogr:id>
  <ogr:attribute1>1</ogr:attribute1>
  <ogr:attribute2>point1</ogr:attribute2>
  <ogr:attribute3>0.1111</ogr:attribute3>
 </extensions>
</wpt>
<wpt lat="39.913" lon="-83.006">
 <extensions>
  <ogr:id>2</ogr:id>
  <ogr:attribute1>2</ogr:attribute1>
  <ogr:attribute2>point2</ogr:attribute2>
  <ogr:attribute3>0.22222</ogr:attribute3>
 </extensions>
</wpt>
<wpt lat="43.247" lon="-115.666">
 <extensions>
  <ogr:id>3</ogr:id>
  <ogr:attribute1>3</ogr:attribute1>
  <ogr:attribute2>point3</ogr:attribute2>
  <ogr:attribute3>0.33333</ogr:attribute3>
 </extensions>
</wpt>
<wpt lat="40.501" lon="-78.691">
 <extensions>
 </extensions>
</wpt>
</gpx>
```

57

GPX stands for GPS Exchange Format. As the name implies, it is a format designed to store and transfer GPS data. GPX makes use of Extensible Markup Language, or XML, and is a tagged text format. In a GPX file, you can store waypoints, tracks, and routes.

## Keyhole Markup Language (KML)

- Used in Google Earth
- KMZ is a compressed versions of a KML
- Uses XML

```
<?xml version="1.0" encoding="utf-8" ?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document id="root_doc">
<Schema name="kml_example" id="kml_example">
                <SimpleField name="id" type="int"></SimpleField>
                <SimpleField name="attribute1" type="int"></SimpleField>
                <SimpleField name="attribute2" type="string"></SimpleField>
                <SimpleField name="attribute3" type="float"></SimpleField>
</Schema>
<Folder><name>kml_example</name>
 <Placemark>
                <ExtendedData><SchemaData schemaUrl="#kml_example">
                                <SimpleData name="id">1</SimpleData>
                                <SimpleData name="attribute1">1</SimpleData>
                                <SimpleData name="attribute2">point1</SimpleData>
                                <SimpleData name="attribute3">0.1111</SimpleData>
                </SchemaData></ExtendedData>
   <Point><coordinates>-96.4367880740894,37.2826128971961</coordinates></Point>
 </Placemark>
 <Placemark>
                <ExtendedData><SchemaData schemaUrl="#kml_example">
                                <SimpleData name="id">2</SimpleData>
                                <SimpleData name="attribute1">2</SimpleData>
                                <SimpleData name="attribute2">point2</SimpleData>
                                <SimpleData name="attribute3">0.22222</SimpleData>
                </SchemaData></ExtendedData>
   <Point><coordinates>-83.0063918894116,39.9125142095311</coordinates></Point>
 </Placemark>
 <Placemark>
                <ExtendedData><SchemaData schemaUrl="#kml_example">
                                <SimpleData name="id">3</SimpleData>
                                <SimpleData name="attribute1">3</SimpleData>
                                <SimpleData name="attribute2">point3</SimpleData>
                                <SimpleData name="attribute3">0.33333</SimpleData>
                </SchemaData></ExtendedData>
   <Point><coordinates>-115.666206923678,43.2471499787955</coordinates></Point>
 </Placemark>
 <Placemark>
   <Point><coordinates>-78.6909808938 93,40.5009793452837</coordinates></Point>
 </Placemark>
</Folder>
</Document></kml>
```

58

Keyhole Markup Language, or KML, is the dominant file type used in Google Earth. It was originally developed by the company Keyhole, Inc. that developed Google Earth before it was acquired by Google. KMZ files are just compressed version of KML files. Similar to GPX, it makes use of tagged text and Extensible Markup Language, or XML.

## Keyhole Markup Language

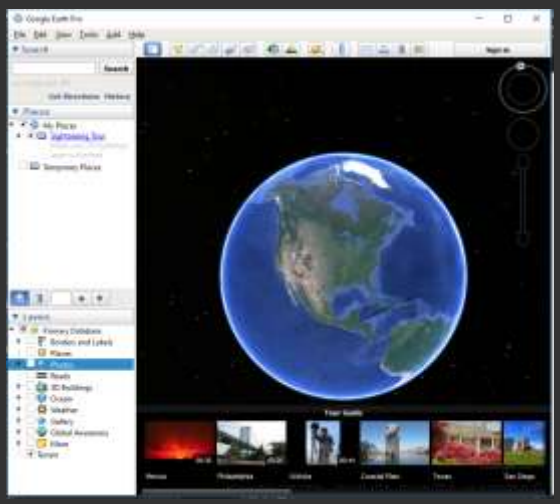
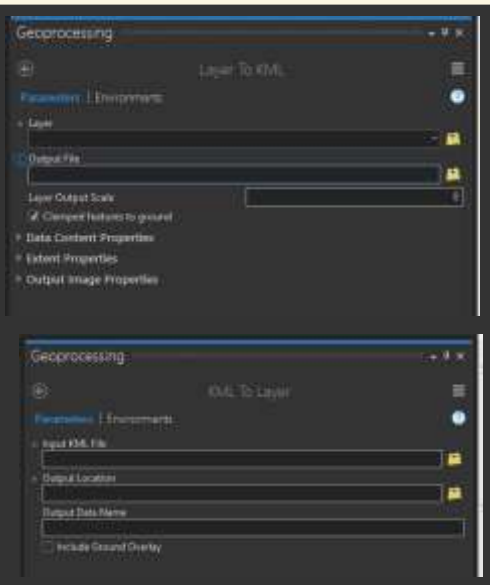


Image from Google Earth





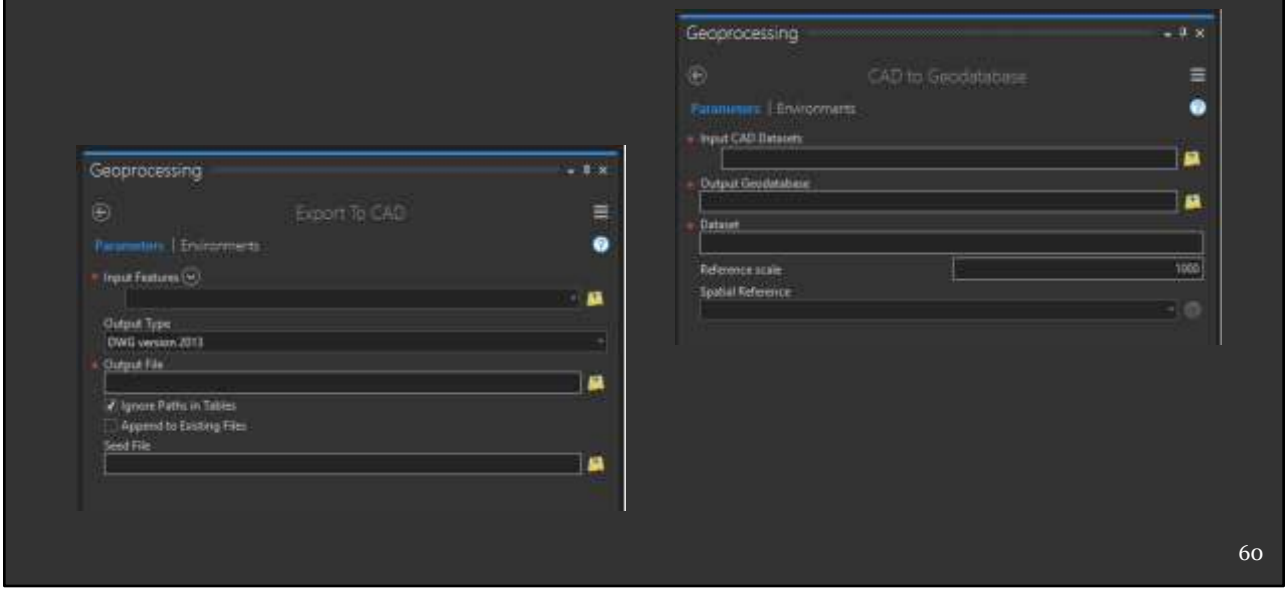


59

Within ArcGIS Pro, it is possible to convert Layers to KML for use in Google Earth. It is also possible to import KML files as layers to be stored as feature classes or shapefiles.

Since Google Earth and Google Earth Pro are both free, I have found that KML is a convenient format in which to share data.

# A Note on CAD Files

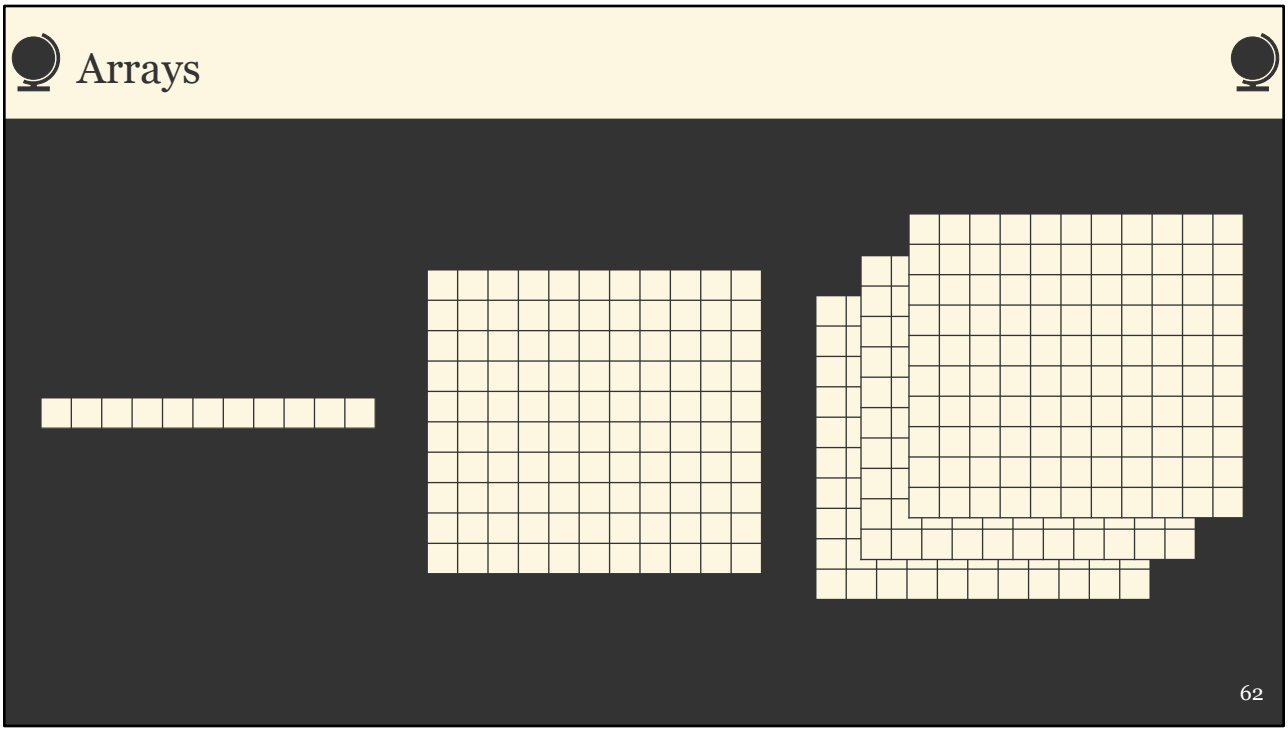



60

It is also possible to import computer-aided design (CAD) data into your GIS or export your geospatial data for use in CAD software.

pixels

# Raster Data



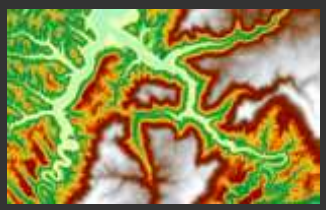

Raster grids can generally be conceptualized as arrays that store values. An array is a set of values that are organized into dimensions. For example, a sequence of values could be represented as a one-dimensional array, commonly referred to as a vector (not to be confused with a geospatial vector layer). A grayscale, or black-and-white, image could be represented as a two-dimensional array, or matrix. The brightness at the pixel location would be represented as a numeric value where higher values indicate brighter areas in the image and lower values indicate darker areas of the image.

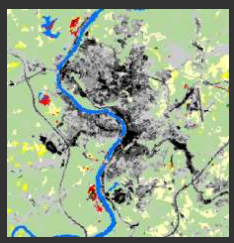
A true color image, such as those taken with a camera, have three separate bands or channels: red, green, and blue. Since we now need to store three values at each location, the array would need to be expanded to a three-dimensional array: height: width, and channels. The array model can be extended to store other types of data. For example, a video could be represented in four dimensions: height, width, channels, and time (or frame). A set of climate variables could be stored in a single data cube with multiple dimensions: height, width, altitude, time, and variable (e.g., temperature, pressure, and humidity).

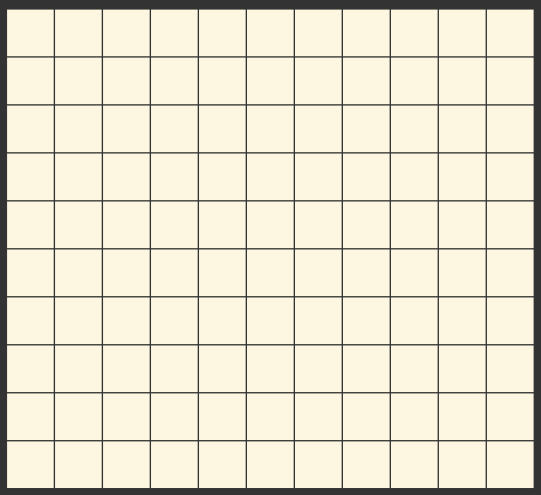
So, arrays offer a flexible and structured data model that can be used to represent a wide variety of phenomena.

## Raster Data

- Like a matrix or array of data
- Origin generally defined as coordinates of upper-left corner
- Cells have defined height and width relative to the coordinate system/projection







63

In contrast to vector data, raster data have a simple data structure. Raster data are a matrix or array of values stored in cells. This matrix or array is spatially referenced so that each cell has a defined extent on the landscape. The origin is generally defined as the coordinate of the upper-left corner of the grid extent.

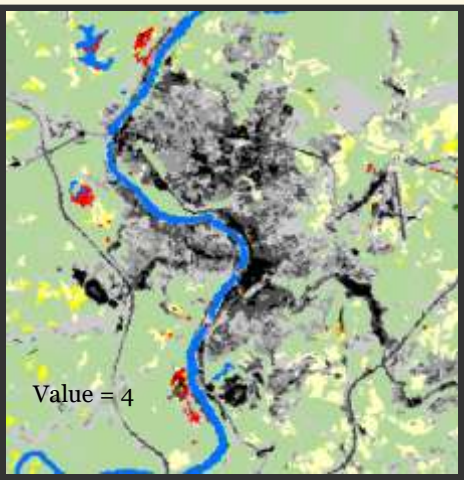
So, if the origin, projection, cell size, and number of rows and columns are known, then the data can be rendered to a map space.

The raster data model can be used to store a variety of data types, including categorical data, continuous variables, and multiband imagery.

## Raster Data: Categorical

- Cells hold a value
- Value is a place holder for a category (nominal data)
- Commonly has attribute table

| Rowid | VALUE | COUNT |
|---|---|---|
| 0 | 11 | 602997 |
| 1 | 21 | 3564731 |
| 2 | 22 | 989578 |
| 3 | 23 | 398382 |
| 4 | 24 | 92139 |
| 5 | 31 | 410192 |
| 6 | 41 | 54706787 |
| 7 | 42 | 1822981 |
| 8 | 43 | 1291515 |
| 9 | 52 | 15478 |
| 10 | 71 | 1018245 |
| 11 | 81 | 5361943 |
| 12 | 82 | 986030 |
| 13 | 90 | 51801 |
| 14 | 95 | 40861 |




64

Categorical raster grids represent nominal or categorical data.

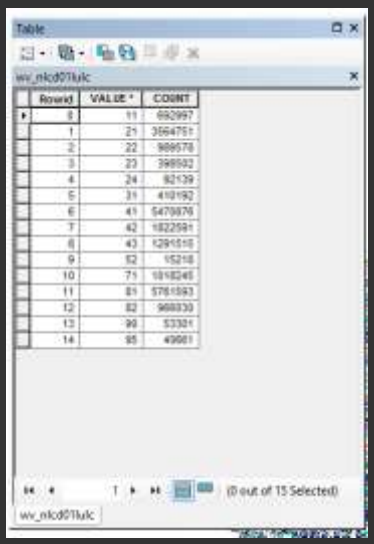
Examples of data that could be appropriately modeled as categorical grids include land cover, vegetation types, wetland types, soil types, and land ownership.

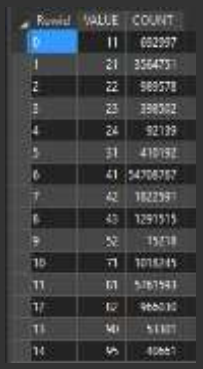
Each cell in a categorical grid will hold a number. This number will be a stand-in or placeholder for a category. For example, the value 41 could represent deciduous forest.

Categorical grids generally have attribute tables. This attribute table can have many fields, but it generally has at least a value field, which provides the code that represents the category of interest. It also contains a count field that provides the number of cells in that category.

It is possible to provide additional fields, for example you could include a name field to provide the category name.

## Attribute tables for categorical raster grids




- How do you convert from a count to an actual area?

65

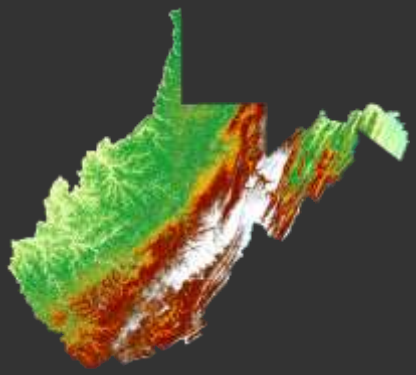
The count field in the attribute table provides the number of cells associated with each value or category. This does not represent a land area. To obtain a land area, you will need to multiply the count of cells by the size of each cell.

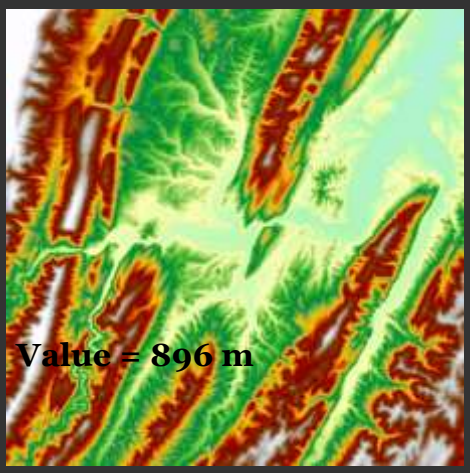
For example, if the cell size is 30 by 30 meters, then you would need to multiply the cell count by 900 to get a land area in square meters. You can then use conversion factors to convert to different area units.

This is a common mistake. So, if you need to obtain a land area, make sure to perform the correct calculations.

## Raster Data: Continuous

- Cells hold a value
- Value is a true quantity (numeric data), not a place holder






66

A continuous raster grid is used to represent continuous data, including interval and ratio type data.

The value stored in the cell represents the actual measure. It is not a placeholder for a category.

Examples of data that could be appropriately modeled using a continuous raster grid are elevation, topographic slope, topographic aspect, depth to bedrock, depth to water table, precipitation, temperature, and barometric pressure.

The example provided represents elevation measurements. So, each cell holds an elevation measurement.

## Radiometric Resolution

- Integer = Only holds whole numbers
- Float = Can hold decimals
- Signed = can hold negative values
- Unsigned = cannot hold negative values
- Pixel Type/Pixel Depth
  - 4-bit = $2^4$ (16) unique values
  - 8-bit = $2^8$ (256) unique values
  - 16-bit = $2^{16}$ (65,536) unique values
  - 32-bit = $2^{32}$ (4,294,967,296) unique values
  - 1-bit, 2-bit, 11-bit, 12-bit, etc.


Can think of this as the degree of brightness difference that can be quantified

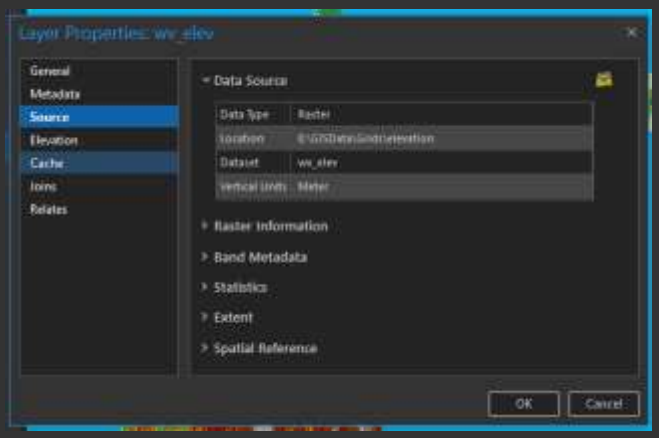
67

Radiometric resolution has to do with the levels in brightness or number of unique values that can be differentiated.

This relates to the pixel type or pixel depth of an image. For example, a 4-bit image would be able to differentiate 16, or 2 to the 4th power, discrete levels of brightness. In contrast an 8-bit image could differentiate 256, or 2 to the 8th power, discrete levels of brightness. A 16-bit image could differentiate a large number of discrete brightness differences. As an extreme case, a 1-bit image could only differentiate 2 to the 1st power, or 2, levels of brightness. So, the result would be black-and-white, or on-and-off.

You can think of this as the precision of the measurement. As an analogy, increasing radiometric resolution is like making length measurements to the nearest millimeter as opposed to the nearest meter. In other words, you will be able to differentiate smaller differences in length.

## Info about your raster data




- Number of columns
- Number of rows
- Number of bands
- Cell size
- File size
- Pixel type/Pixel depth
- NoData value
- Spatial reference

68

You can find information about a raster grid in ArcGIS Pro in the Layer Properties window.

The Raster Information will provide the number of columns, number of rows, number of bands, cell size, uncompressed file size, pixel type, and NoData assignment.

Spatial reference information is provided under Spatial Reference.

## Raster data are always rectangular


69

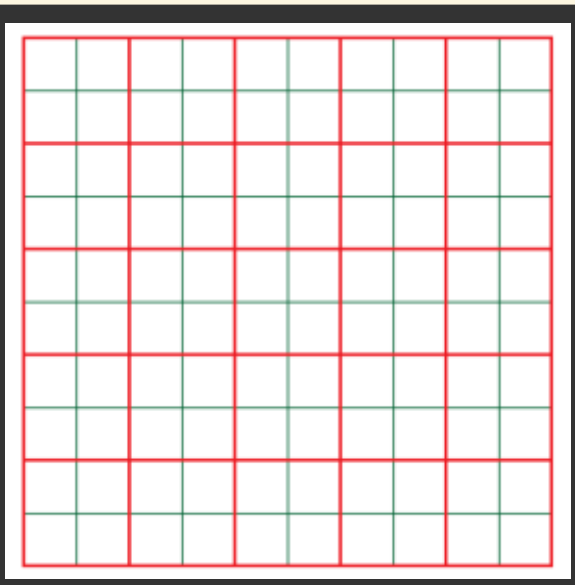
All raster grids are rectangular even if data are not provided in each cell. This is because each row and column must have a full set of cells.

Grid cells that do not hold a measurement still exist, but they store a NoData assignment. When the computer sees this value, it interprets it to mean empty, null, or missing.

For example, for an 8-bit raster grid either the lowest value, 0, or the highest value, 255, is commonly reserved as the NoData value.

In the example above, all cells within the state of West Virginia hold an elevation measurement. However, the full extent of the grid is rectangular. So, the remaining cells, shown here in pink, are coded with the NoData assignment.

## Cell Size



70

Reducing the cell size can have a large impact on the total number of cells and, as a result, the file size.

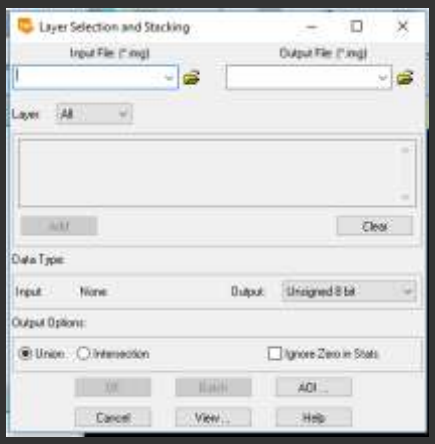
For example, in the image above the red grid represents cells that are 100-by-100 meters. The green grid represents cells that are 50-by-50 meters. Each 100-by-100 meter cell contains four as opposed to two 50-by-50 meters cells. So, decreasing the cell size by half quadruples the number of pixels.

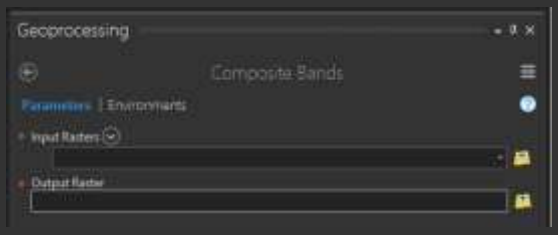
As another example, a 10-by-10 meter cell would contain 100 1-by-1 meter cells.

So, high spatial resolution raster grids can be very large files, especially if they cover a vast spatial extent.

## Raster Grids with Multiple Bands/Layers

- Can have multiband raster that acts as a singe file but is actually composed of multiple raster grids
- Common for multiband imagery
- However, can be use for other types of raster data




71

Other than panchromatic, or black-and-white, photography, image datasets contain more than one band. For example, a true color image contains three bands: red, green, and blue.

For multiband images, the raster data structure is that of an array. Each cell location holds multiple measurements, or a single measurement in each band. You can think of a multiband raster as a stack of single-band layers that act as one file.

Although we will focus on multiband imagery here, you can actually create multiband raster grids from a variety of raster data layers.

## Raster Grids with Multiple Bands/Layers

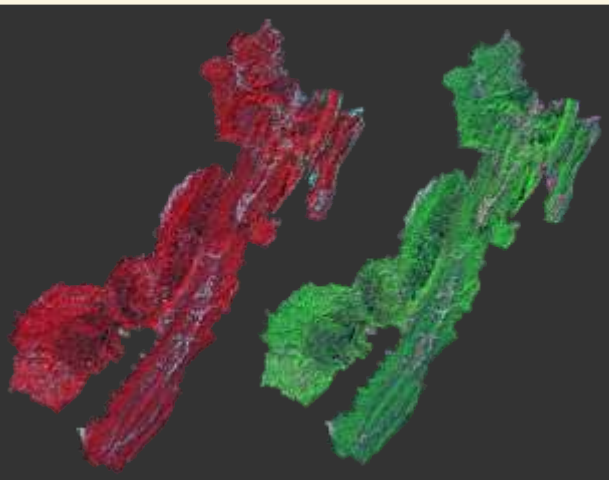
- Can have multiband raster that acts as a singe file but is actually composed of multiple raster grids
- Common for multiband imagery
- However, can be use for other types of raster data




This slide provides an example of a true color image where the raster layer consists of three bands: red, green, and blue.

## Raster Grids with Multiple Bands/Layers

- Can display single band as a grayscale image
- Can display multiple bands as a false color composite
- Common technique in remote sensing




73

It is also possible to collect data that includes bands outside of the visible spectrum. Such imagery is generally referred to as multispectral images. They are often displayed in false color, where the color channel displays a different color.

For example, in the first image the red channel is showing near infrared, the green is showing red, and the blue is showing green. In the second image, red is showing shortwave infrared, green is showing red, and blue is showing green.

False color composites can be used to visualize imagery to highlight specific features or make different cover types more differentiable.

## Raster File Formats

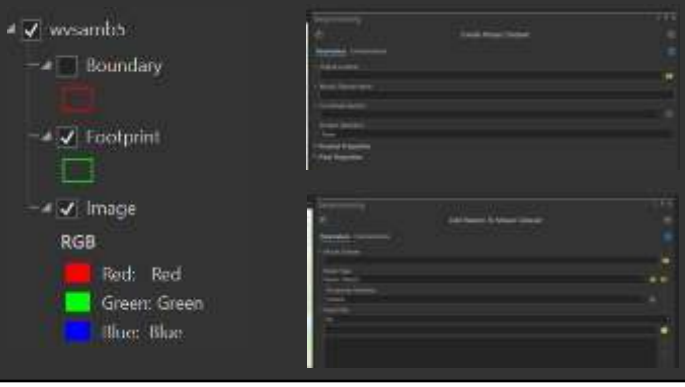
1. ESRI Grid
2. Erdas Imagine (.img)
3. GeoTIFF (.tif)

74

There are a variety of raster grid formats. However, I commonly use Erdas Imagine (.img) or GeoTIFF (.tif) files.

## Mosaic Datasets

- Reference to raster layers
- Does not require data to be copied into a new file




75

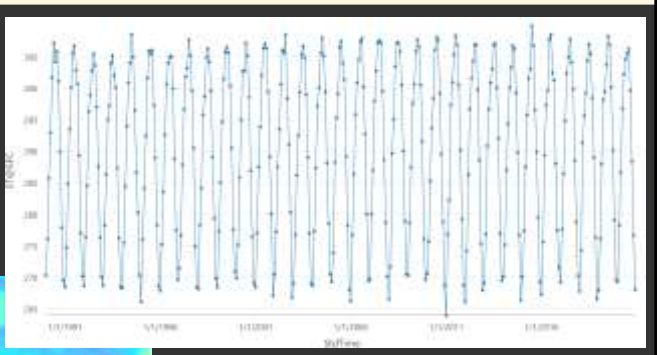
In the ArcGIS/ESRI environment, Mosaic Datasets allow for the generation of virtual raster grids that reference other data files. So, these are not true raster files but offer a means to reference multiple actual files to view and work with the associated data.

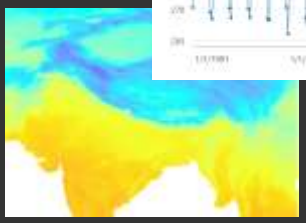
ArcGIS Pro includes a variety of tools for building and manipulating Mosaic Datasets. For example, you can alter the scheme, which changes the characteristics of the mosaic and the types of data that can be included, add and delete raster data, calculate the boundary of the mosaic and the footprints of individual images within it, define how data will be combined based on seam lines, define what data will be displayed or used at different scales, define color balancing rules, and build overviews, similar to raster pyramids.

Note that many geoprocessing tools will accept mosaic datasets as input. However, these datasets cannot be used outside of the ArcGIS environment.

## Multidimensional Raster Grids

- Store more than three dimensions
- Commonly used in climate science
- Time
- Depth
- Multiple Variables
- Common Formats
  - GRIP, HDF, NetCDF, CRF





76

Multidimensional raster grids further extend the raster data model to incorporate more than three dimensions (i.e., height, width, and channels). For example, a set of climate variables could be stored as a single file with height, width, variable, time, and altitude dimensions. Such models are sometimes referred to as data cubes. A series of multispectral images representing different dates, such as a series of Landsat images, could be stored as a single file with height, width, channel, and time dimensions. Some common file formats for multidimensional raster grids include:

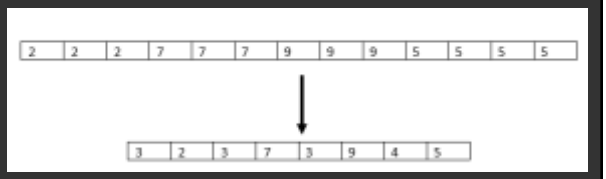
GRIB = General Regular-Distributed Information in Binary
HDF = Hierarchical Data Format
NetCDF = Network Common Data Form.

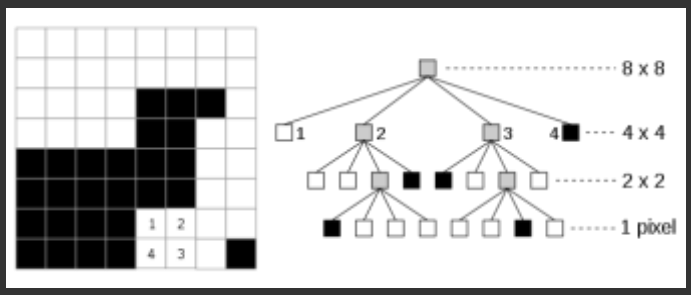
ESRI has developed a raster file type for multidimensional raster data called Cloud Raster Format (CRF).

Multidimensional raster grids expand the types of data that can be represented and also the types of analyses that can be performed. Specifically, multidimensional raster data support time series analysis, anomaly detection, aggregation over time or other domains, and trend analysis. Climate scientists and modelers have been using these types of file structures for some time because they undertake the types of analyses just listed.

## Data Compression

- Reduce file size
- Lossless = all information is maintained
- Lossy = some information is lost
- Limits redundancies in homogenous areas
- Types
  - Run-length encoding
  - Quad tree




https://commons.wikimedia.org/wiki/File:Quad_tree_bitmap.svg

77

In order to reduce the size of files, compression is often applied. An example of file compression is using an MP4 file to store video or an MP3 to store audio.

Compression methods can be grouped into two broad types: lossless and lossy.

When using lossless compression, the original data or values can be retrieved from the compressed file. In other words, there is no loss of information. When using lossy compression, some information is lost and the original data cannot be restored.

Compression is commonly applied to raster data. The goal is to remove or minimize redundant information. Example compression methods applied to raster data are run-length encoding or quad tree compression.

# Comparisons

## Comparing Raster vs. Vector

| Comparison | Vector | Raster |
|---|---|---|
| Data structure | Complex, can also include complex attributes | Simpler |
| Storage requirements | Small for most datasets | Can be quite large (file size) |
| Precision/level of detail | Unlimited | Set by cell size |
| Ease of use in modeling | Best for network models, may be more difficult to do overlay computations | Good for quick layer overlays |
| Display/output | Map-like output (smoother edges) | Can be blocky |

From *GIS Fundaments* by Bolstad

79

This table provides a comparison of vector and raster data.

Vector data are considered to have a more complex data structure than raster data since vector files rely on a complex structure of interrelated vertices and topological relationships while raster data are simply a matrix or array of values.

Vector files are generally smaller than raster files. However, this is not always true. Vector files can be quite large if a large number of features are included or a lot of detail is provided by including many vertices. Raster files can be small if the cell size is large and/or the data cover a small geographic extent.

The precision and level of detail for raster data are limited by the cell size. For example, an elevation raster can only store a single elevation measurement per cell. Multiple elevations measurements cannot be differentiated within a cell.

Vector data are used for some types of modeling, such as network analysis. Raster data are commonly used for overlay and digital terrain analysis. We will discuss a wide range of modeling techniques in later modules.

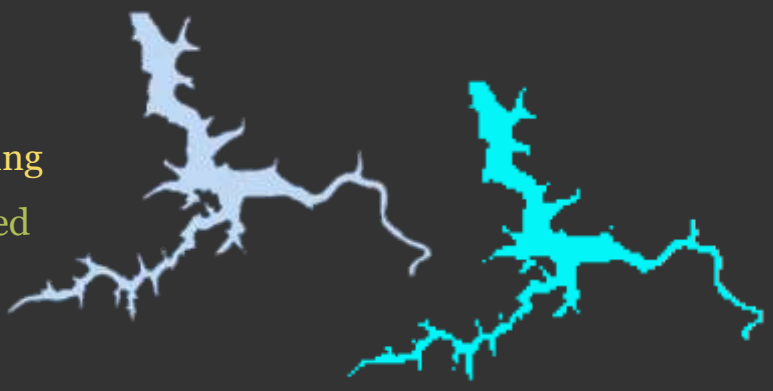
In regard to display and output, vector data are often cartographically pleasing and map-like as edges can be made smooth whereas raster data can appear

blocky. However, this depends on the nature of the data and the scale at which it is being viewed. Raster data will only look pixelated if zoomed in to a scale at which the raster data are not designed to be viewed. Vector data can also look coarse if not a lot of vertices are provided. So, I would argue that both vector and raster data can be appropriate for display as long as they are used at the appropriate scale.

## Conversions

Things to consider

1. Cell size
2. Generalization/smoothing
3. How the data will be used



80

It is possible to convert between the two data types.

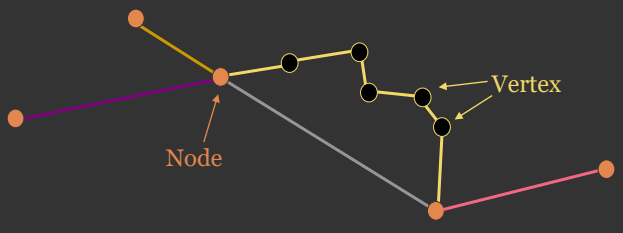
When converting from raster to vector, you must determine if any smoothing or generalization will be applied.

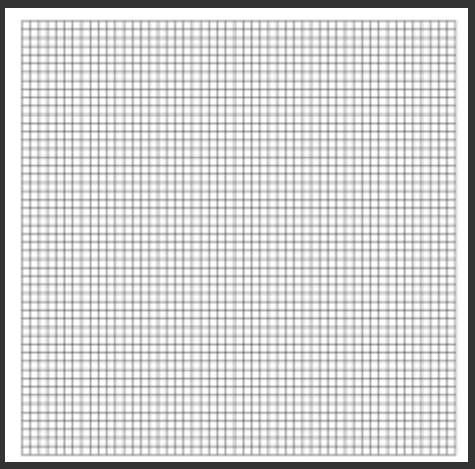
When converting from vector to raster, you must decide the cell size and what attribute will be mapped to the cells.

Conversion is a common task. For example, you may need to convert vector data to raster grids to include it in a raster-based modeling process.

How the data will be used often dictates whether it should be represented as vector or raster data.

## Vector or raster?



81

Many factors should be considered when trying to determine whether to use the vector or raster data model for mapping a certain variable or feature.

Discrete features, such as the location of fire hydrants, are commonly represented as vector features whereas continuous features, such as topographic slope, are commonly modeled as raster data. However, there are exceptions. For example, elevation can be represented as a TIN model.

The scale at which the data will be used or displayed should also be considered.

The purpose and use of the data should also be considered.

In summary, there are many factors to consider when determining the optimal method to represent your spatial data.

# Geospatial Data in ArcGIS Pro

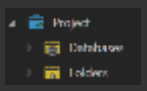
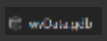
82

The following exercises ask you to work with and explore geospatial data in ArcGIS Pro.

# Change Base Map

1. With the wv map and Map Tab active, click on the Basemap icon. 
2. Select a new base map from the list of options.

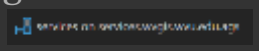
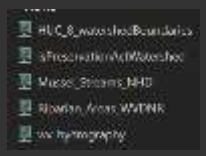
## Add a Local Layer

1. With the wv map and Map Tab active, click on the Add Data icon. 
2. Click on Databases under the Project heading. 
3. Click the wvData.gdb option. 
4. Within the database, select on the libraries file and click OK. 
5. The libraries layer should now be added to the wv map.

# Connect to a Server

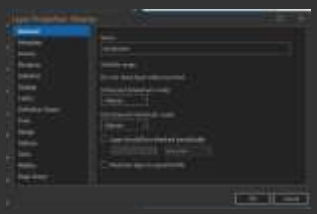
1. Navigate to the following link in a web browser: https://www.mapwv.gov/gis_services.html
2. Copy the link provided on the page and listed after ArcMap.
3. Back in ArcGIS Pro, navigate to the Insert Tab.
4. Select Connections → Server → New ArcGIS Server.
5. Paste the URL into the Server URL box (maintain https://). You do not need to provide a Username or Password.
6. Click OK.


## Add a Layer from the Server

1. With the wv map active, navigate to the Map Tab.
2. Click the Add Data icon. 
3. Under the Project section, select Servers. 
4. Click on the services.wvgis.wvu.edu.ags server. 
5. Within the server connection, select the Inland_Waters folder.
6. Within this folder, select the wv_hydrography map service. This will add a hydrography layer to the map. 

# Rename a Layer

1. With the wv map active, right-click on the libraries layer.
2. Select Properties from the menu.
3. Under the General Tab, change the **Name** of the layer to "wvLibraries".



This method can be used to give layers in the map more meaningful names. !

# Geospatial Data in QGIS

## Browser and Layers

### Browser

- Local Files
- Database Connections
- Service Connections

### Layers

- Similar to ArcGIS Pro Content Pane
- Right-click for layer options

89

As noted in a prior module, the Browser allows you to explore and connected to local and remote data layers while Layers lists the data displayed in the current map.

## File Info

- Right Click → Properties → Information

90

Information about a layer or file can be obtained by right-clicking on it within Layers, selecting Properties, then navigating to Information.

## Layer Information

- Use the utah project.
- Use the layer information to answer the following questions.

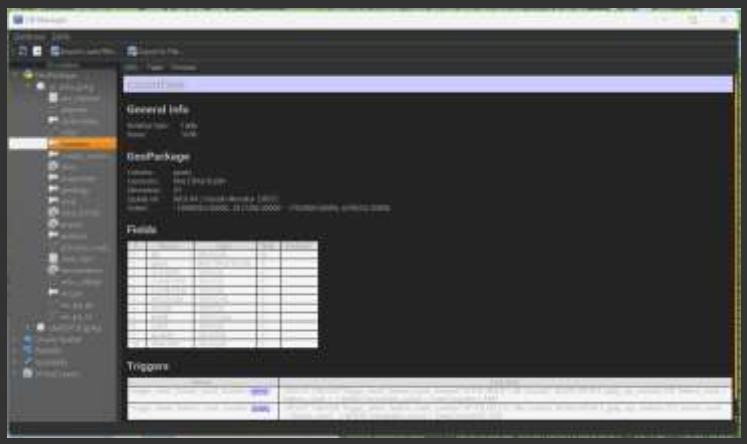
1. What is the CRS for the counties layer?
2. How many features are in the counties layer?
3. How many attributes are in the counties layer?
4. What is the file format of the nlcd layer?
5. What is the cell size/resolution of the nlcd layer?
6. What is the NoData value for the elev layer?
7. What is the maximum elevation in the elev layer?
8. What is the mean elevation in the elev layer?

91

This exercise asks you to explore layer information in QGIS.

## Connect to Database

- Use Browser
- Types
  - GeoPackages
  - SpatiaLite
  - PostgreSQL
- Database → DB Manager




92

Using the Browser, you can connect to a variety of local or remote databases including GeoPackages, SpatiaLite, and PostgreSQL/PostGIS.

The DB Manager allows you to access databases and perform queries.

## Connect to Map Services

- WMS/WMTS
- Vector Tiles
- XYZ Tiles
- WCS
- WFS
- ArcGIS Rest Servers
- GeoNode

93

You can also connect to a variety of published services.

## Connect to a Service

- Use the utah project.
- Follow the instructions on this page to add WMTS/WMS services to your map

https://gis.utah.gov/discover/#sign-up

- Note: you will need to create an account

94

This exercise asks you to connect to a service provided by the state of Utah.

## Format Conversion

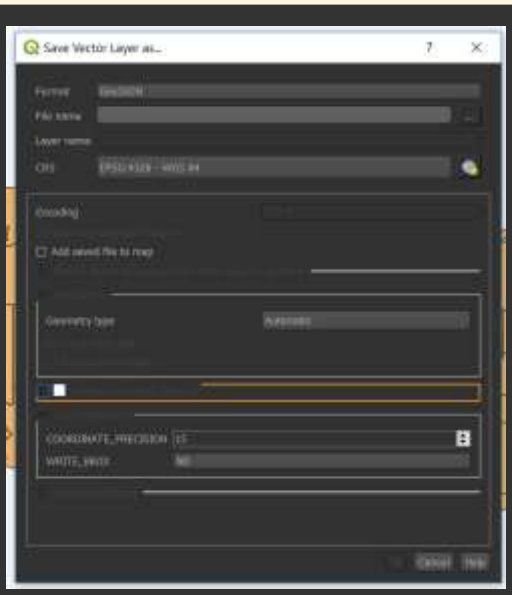
- Export → Save Feature As
- Vector: GeoPackage, ESRI Shapefile, AutoCAD DXF, Comma Separated Values (CSV), Geography Markup Language (GML), GeoJSON, GeoRSS, GPS eXchange (GPX), Keyhole Markup Language (KML), SpatialLite, etc.
- Raster: GeoTiff (.tif), GeoPackage, Erdas Imagine Images (.img), HDF4, Idrisi Raster, SAGA Binary Raster, R Raster, etc.

https://gdal.org/

95

QGIS is a great tool for converting between file types and makes use of GDAL.

## GeoJSON in QGIS



- Can export files to GeoJSON format
- Can create bounding box
- Can set coordinate precision
- QGIS can read and write many file types

96

As noted above, QGIS is my preferred tool for converting vector files to a GeoJSON format.

## Convert Layer to GeoJSON

- Use the utah project.
- Save a copy of the towns layer in GeoJSON format to the provided temp folder.

97

This exercise asks you to convert a layer to GeoJSON.

This is the end of this lecture module.

Please return to the West Virginia View Webpage for additional content.

Thanks! Hope you found this useful.