Full-function systems, like ArcGIS, do three things:

- **Compute**
- **Store & Manage**
- **Visualize**
Topics Covered

• What is Spatial Analysis?

• Getting technical and answer the two most common questions in geography:
  - What’s on top of what?
  - What’s near what?

• Best practices and Usage tips
Video: Solving Spatial Problems

- [Link](http://video.esri.com/watch/681/an-overview-of-solving-spatial-problems-using-arcgis)

  Examples of statistical and surface analysis
Geoprocessing

- Part of the ArcGIS system for computing, managing and manipulating data.

  - Ask questions; get results
  - 2 fundamental purposes:
    - modeling and analysis
    - automate GIS tasks
Spatial Analysis

• The process of modeling, deriving results by computer processing, then examining and interpreting the model results by visualization, queries, and reports.
The Basis of Spatial Analysis

- Spatial relationships
  - Containment
  - Adjacency
  - Distance
  - Selection and Statistics
Vector vs. Raster

• The two basic data structures for storing and manipulating GIS data

• Vector
  - A coordinate-based (points, lines, and polygons)
  - Attributes are associated with each vector feature
  - Good for representing clearly defined objects

• Raster
  - A cell-based data model
  - Each cell contains an attribute value
  - Especially suited for continuous data such as elevation, air pollution, and precipitation.
What’s on top of what? (Overlay)

The most basic question in geography
Question – *What was the first ever product by Esri?*

- *Answer – PIOS (polygon information overlay system)*
What’s on top of what?

- What land use is on top of what soil type?

- What parcels are within the 100-year floodplain? ("Within" is just another way of saying "on top of.")

- What roads are within what counties?

- What wells are within abandoned military bases?
Overlay Tools

• Most important and frequently used:
  - Intersect
  - Union

• Other overlay tools (optimized for type of query):
  - Identity
  - Erase
  - SymDiff
  - Update

• Feature attributes are carried through the operation
Demo: Basic Overlay - Two polygon layers with tabulation

• Data: Council districts and census tracts
  - Each track has a land value
  - Census tracts split by council districts

• Problem – what is the land value of each district?

• Two basic methods:
  - Create a new dataset of districts
  - Create a table (which can be joined back to the districts)
Key points of the demo

• Know your data - do attribute values need to be proportioned?
  - Use the **Make Feature Layer** tool and check “Use Ratio Policy” for attributes to be apportioned
    - Caveat: assumes data distributed uniformly across the area – a BIG assumption!

• Use simple table operations on output data
  - **Summary Statistics, Frequency**

• Join outputs back to input
  - **Join Field, Add Join**

• Experiment with small datasets!
  - Examine results with identify tool, select tool, opening table, etc.

• Use **in_memory** workspace for small to intermediate size data
  - No overhead of writing to disk
  - Useful for GP services
Key points of demo - Dissolve

• **Dissolve** allows you to aggregate geometries based on attribute

• Result is a new dataset

• You are responsible for choosing how to aggregate attribute values
  - Many options how to aggregate attribute values
  - We used “Sum” in demo
Demo: More tabulation

• Scenario:
  - Dataset of forest types
  - Dataset of districts

• For each district, find the area of each forest type
  - (Additionally, find the percent area of each forest type)

• Tabulate Intersection - New tool at 10.1
  - Computes the intersection between two feature classes and cross-tabulates the area, length, or count of the intersecting features
  - Used to be possible by combining numerous tools
  - No intermediate data
Dissecting overlapping polygons

Spaghetti and Meatballs
Demo: Dissecting overlapping polygons

- This is a common scenario:
  - Single feature class of polygons
  - Polygons overlap each other due to how data was collected
  - You need to characterize the overlap areas

- Methodology – spaghetti and meatballs
  - ‘Spaghetti’ = line work from multiple sources made into polygons
  - ‘Meatballs’ = centroids (points) of each polygon
The spaghetti and meatballs recipe

- Create polygons from ‘spaghetti’
  - Feature To Polygon tool
- Create ‘meatballs’ – a point inside of polygons
  - Feature to Point tool
- Use Spatial Join or Intersect with the meatballs and the original polygons to collect attributes
- Summarize attributes and join them back to spaghetti polygons
- Remove ‘holes’ created by the Feature To Polygon tool using the Select tool
Overlaying points, lines, and polygons
Demo: Point in Polygon, Polygon on Line, Line on Line

• Oftentimes, you need to overlay different feature geometries

• Very common queries:
  - Lines intersecting polygons
  - Lines intersecting lines
  - Points in polygons
  - Point on lines

• Let’s examine some typical cases
Key Points of demo

- **Spatial Join** great for summarizing points falling within polygons
- Discovered what polygons were on top of what lines
- Created points where lines intersect
- Split polygons with line features
  - Used our new friend **Features To Polygons** along with **Identity**
Overlay is computationally crazy

Things to watch for
Overlaying large datasets

- Overlay algorithms are extremely intricate and complex
  - An ‘11’ on the scale of computational intensity
  - Esri has at least 100 programmer years refining overlay and we’re constantly working on it

- Overlaying large datasets is CPU and RAM intensive
  - Schedule large overlays accordingly (i.e., lunch, after hours)
  - Shut down all other applications
  - Use computers with lots of memory
Scalability – dataset size and integrity

- Biggest problem: *Number of vertices per feature*
  - Not the number of features
  - Coastlines, Road casings, Boundary/outline polygons
    (single polygon around boundary of entire dataset)

- For datasets from external sources...
  - Use *Dice, Check Geometry* tool
  - See blog post [Dicing Godzillas (features with too many vertices)](https://geoprocessingresourcecenter.com) on the geoprocessing resource center.
Large data processing – Best Practices/Recommendations

- Enterprise or local file geodatabases should be used as the output workspace
- Do not start other operations
- In 10.0 desktop, run in foreground
- Check for any bad geometries first
- Do not use in_memory workspace for output
- Check for “huge” features & Dice them if necessary
- Analyze the feature vertex density & simplify if possible
- Tolerances / Resolution – Use Defaults!
- GeoDatabase design
- Invest time up front planning

Cont …
Large data processing – Best practices/Recommendations

• Run ArcGIS on 64bit OS with ample RAM
  - ArcGIS Desktop can access twice as much memory when run on 64bit OS because of being Large Address Aware
  - Python script tools must be run in-process
  - Stand-alone scripts should be run from the Python window or wrapped as script tools

• Use a 64bit offering of ArcGIS
  - 10.1 – ArcGIS Server is natively 64bit
    - Using Python to run tools will take advantage of more memory
    - Does not require services
Large data processing - ArcGIS 10.1

• Better memory management while an overlay operation runs

• No hard limit to amount of memory
  - Adds considerable scalability in 64bit environment
64bit Background – ArcGIS 10.1 (SP1 Time frame)

- ArcGIS for Desktop – Background Geoprocessing (64x) (Windows)
- ArcGIS Engine – Background Geoprocessing (64x) (Windows)
  - Separate install (release date same as SP1)

This is **not** a solution which answers performance questions. 64x BG is not always faster, but does scale to provide ability to crunch large data that may have not been possible before.
Conclusions

- Overlay is the most basic question made of geography
  - What’s on top of what
  - (The other most basic question – what’s near what – is covered next)
- Overlay is almost always accompanied by some sort of table analysis
  - Summary Statistics, Frequency, Pivot Table, Make Query Table, Join Field
- Keep watchful eye out for field bloat
  - Delete unnecessary fields
Raster Overlay
Raster Overlay tools

- Zonal Statistics
- Tabulate Areas
- Combine (equiv. to Union)
- Weighted Overlay
  - Classic Suitability Overlay
  - All inputs converted to the same scale
  - Set weights for each input
  - Weights must sum to 1
- Weighted Sum
  - Similar to Weighted Overlay
  - (weights can add up to whatever I want)
  - Allows float output
  - Any weight values are allowed
- Fuzzy Overlay (Fuzzy attributes) – new in 10
Demo:
Spatial Analyst suitability
Basic geographic question: What’s near what?
Waldo Tobler’s first law of geography:

“Everything is related to everything else, but near things are more related than distant things.”
Three proximity “datums”

• **Areas** (area expanding tools)
  - Buffer, Thiessen, Network Service Area (network), Euclidean allocation, Cost Allocation
  - The areas produced often used in overlay
    - Such as students w/in 1 mile of school

• **Numerical Value**
  - Distance returned as a number
  - Near, Near Table, Point Distance, Select By Location, Spatial Join, etc.

• **Linear Measure** (routes on a network)
  - Measures could be distance, time, etc.
  - Raster Cost Path
  - i.e. like a ruler
3 metrics

- **Euclidean**
  - Distance on a flat map (Cartesian coordinates)

- **Cost**
  - Cost on a network
  - Cost of a surface

- **Geodesic**
  - Distance on the globe
  - This is the only “true” distance measure as any flat map distorts scale
Buffer

The most-used area expanding method
Don’t ever forget: Projection matters! Anyone see a problem with this?
The retraction...

Circumference of the earth is 40,000KM...so 15000 is almost half...not so comforting, but correct
All projections distort distance

• Even equidistant projections – they only minimize distortion for a region

• The only projection that doesn’t distort distance is an Azimuthal Equidistant centered on a single point
  - Distances from that single point to all other points are correct

• Some projections greatly distort distances
  - Mercator at high latitudes – i.e., Buffer around Iceland, missile reach from North Korea
  - Any conformal projection like Mercator trying to preserve angles
When creating large buffers at continental scale

• If buffering points in 10.0:
  - Use un-projected data (geographic coordinate system) as input and Buffer will create geodesic buffers
    - (Same result as Azimuthal Equidistant centered on each individual point)

• If buffering lines or polygons in 9.x or 10.0:
  - Coordinate system environment should be an equidistant projection (i.e. buffering Norway)
  - There are predefined equidistant projections for world, continents, and countries
Buffer 10.1

- Creates true geodesic buffers for point, line and polygon data
  - geodesic buffers truly are more accurate than Euclidean
  - geodesic buffers will take more time than generating Euclidean buffers
Demo: Geodesic Buffers
Create Thiessen Polygons (or Voronoi)

- Constructed from points

- Each polygon contains only one input point, and any location within a polygon is closer to its associated point than to the point of any other polygon

- This is a class of area expanding that we call ‘allocation’
  - Everything inside the area generated is closer to the feature that generated the area
  - Unlike buffers

- Like Buffer, output frequently used in overlay
Selecting nearby features

Select Layer By Location

Spatial Join
Select by Location

- Use this tool to answer spatial relationship questions
- Fast and scales extremely well
- Uses layer as input
- Can be use to split up large data
  - i.e. Tweet data (massive amounts of points) by continent
- Lots of options, including distance - provides 13 “relational operators”
Tools that return distance values (Euclidean)

Near
Near Table
Point Distance
Near, Generate Near Table, and Point Distance

- **Near** – adds attributes to the input feature
  - Records the nearest feature and its distance
  - Multiple near features allowed
- **Generate Near Table** – creates a new table
  - Many records for each input feature (1:M)
  - Records all features within search radius
  - Multiple near features allowed
- **Point Distance**
  - Points only
- **Generate Near Table and Point Distance** allow input and near features to be the same
Demo: Generate Near Table demo

- **Scenario:**
  - Point locations of delivery trucks
  - Each location time-stamped
  - Find locations where trucks are within 500 feet of each other within 1 minute
Generate Near Table demo key points

• Use CUTOFF DISTANCE

• Generate Near Table used to get all combinations of points within specific distance of each other

• Can use the same features as the input features and the near features

• Use Add Join or Join Field to get a table with the time for the from features (in_fid) and the time for the to features (near_fid)

• Calculate the time difference
  - \[ \text{Abs (DateDiff ("s", [Time], [Time_1]) / 60} \]

• Query route id 1 not equal to route id 2 and time less than 1 minute
  - "RID" <> "RID_1" AND "TimeDiff" <= 1.0
Generate Near Table demo key points

• Use Calculate Field Code Block to generate a unique id for each route to route connection

  Expression:
  calcID(!RID!, !RID_1!)

  Code Block:
  ```python
def calcID(id1, id2):
    if(id1 > id2):
      cid = str(id2) + "_" + str(id1)
    else:
      cid = str(id1) + "_" + str(id2)
    return cid
  ```

• Use Make Query Table for one to many join
Network

- Most human goods and services travel across networks
- Distance based on an impedance attribute: the cost to traverse a line or the intersection of lines:
  - Length
  - Time
  - Grandmother cost: minimize number of left hand turns at uncontrolled intersections
Demo: Shortest route
Demo review – Network Analysis layers

• Use Make <network analysis> Layer
  - We used Route
  - We’ll look at the others in the next demo

• Network Analysis are group layers containing individual layers
  - Each individual layer expresses some facet of the analysis, including the solution
  - Use the Select Data tool to select layer to work on

• Output features are lines with measures
  - You can hatch the measures
  - You can query measures at locations along route (next demo)
More network analysis

- **Closest Facility**
  - Given an “incident”, find the N closest “facilities”
  - Incident = your house and it’s on fire
  - Facilities = fire stations

- **Service Area**
  - Create buffers around a facility
  - 5, 10, 15 minute drivetime polygon around facilities

- **OD Matrix**
  - A list of origins and a list of destinations (OD)
  - Give cost between all pairs
  - Network equivalent of Generate Near Table
Raster

Euclidean buffer

Cost buffer
Buffering with Raster

- **Euclidean Allocation**
  - Similar to Thiessen
  - Works for points and lines
  - Gives what is the closest feature to the given location
  - Much more efficient (faster) than Buffer and Feature to Raster

- **Path Distance Allocation**
  - Can give elevation surface so that distances are distance on the surface instead of straight line distance
  - With Path Distance Allocation, generate unlimited allocation and distance
  - Use Con to get allocation value where distance is less than threshold
Buffering with Raster

• Feature
• Buffer

• Euclidean
• Allocation

• Path Distance
• Allocation
Take away points…

- Spatial analysis is the true power of GIS. It is our core competence

- ArcGIS spatial analysis capabilities help advanced users to do their work better with sophisticated analytical tools

- Being able to share the GIS Analysis online reduces the learning curve for complex analysis
Analysis and Geoprocessing Resource Center

- **resources.arcgis.com**
  - (This is the home page. From here, navigate to the Analysis community page.)

**Option 1: from the home page**

**Option 2: click “Communities” to get a list of all resource centers**
Features of the Analysis Resources Center

- Features Stories
- Model and script tools
- Education Gallery
- Blogs
- Forums
- Videos
- Twitter feed