

The ArcGIS Compendium

— Volume A: Foundations

ON THIS PAGE

- [The ArcGIS Compendium — Volume A: Foundations](#)
 - [How GIS Thinks: Vector, Raster, and Data Models](#)
 - [The ArcGIS Ecosystem: Every Product and When to Use It](#)
 - [Coordinate Systems and Projections, Complete](#)
 - [Cartographic Design: Making Maps People Can Read](#)
 - [Finding Data: Living Atlas, Open Portals, and Evaluating Quality](#)

The ArcGIS Compendium

— Volume A: Foundations

The concepts under everything else — how GIS represents the world, the full Esri product family, coordinate systems, cartographic design, and finding good data.

One of eight volumes. Approximately 22890 words. Chapters cross-reference the whole set by number.

In this volume:

- Chapter 1 — How GIS Thinks: Vector, Raster, and Data Models
- Chapter 2 — The ArcGIS Ecosystem: Every Product and When to Use It
- Chapter 3 — Coordinate Systems and Projections, Complete
- Chapter 4 — Cartographic Design: Making Maps People Can Read
- Chapter 5 — Finding Data: Living Atlas, Open Portals, and Evaluating Quality

How GIS Thinks: Vector, Raster, and Data Models

A map is a set of decisions pretending to be a picture. Before a single road appears on your screen, someone decided that a road is a line, that the line has a name and a speed limit attached to it, that its curves would be captured with a certain fidelity, and that below a certain size, driveways simply don't exist. Geographic information systems — GIS, the software family that ArcGIS belongs to — run entirely on decisions like these. The world itself is continuous, messy, and indifferent to categories; a GIS forces everything into one of a few formal structures called data models, and every analysis you'll ever run inherits the strengths and blind spots of whichever structure was chosen.

This chapter is about those structures. If you understand them, the rest of the Compendium gets dramatically easier, because almost every confusing behavior in ArcGIS — why an analysis tool rejects your layer, why zooming in makes imagery blocky, why two "identical" datasets give different answers — traces back to a data model doing exactly what it was designed to do.

Two Ways of Seeing the World

Every geographic dataset you'll ever touch descends from one of two ancestral ideas: the world as a collection of *things*, or the world as a *surface of values*.

The vector view: the world as things

The vector model says: reality is made of discrete objects with locations and shapes. A fire hydrant. A hiking trail. A county. Each object — GIS calls it a **feature** — is stored as geometry (coordinates describing where it is and what shape it takes) plus attributes (facts about it, held in a table). The word "vector" is inherited from computer graphics: it means geometry built from coordinate points connected in sequence, as opposed to a grid of pixels.

Vector data is precise in a particular way: a feature has an exact boundary. A parcel polygon claims to know exactly where one property ends and the next begins. Whether

that claim is *true* is a separate question — see Chapter 5 (Finding Data) on evaluating quality — but the model itself deals in crisp edges.

The raster view: the world as a surface of values

The raster model says: forget objects. Lay a regular grid of square cells over the landscape, and record one value per cell. Elevation: this cell is 412 meters above sea level. Temperature: this cell averaged 18.3 degrees. Land cover: this cell is category 42, "evergreen forest." A raster is essentially a spreadsheet draped over the Earth, and in fact a digital photograph is a raster too — each pixel holds a color value instead of an elevation. That's why aerial imagery, scanned historical maps, and elevation models all live comfortably in the same raster machinery.

Where vector data has features, raster data has **cells** (often called pixels). There is no "thing" in a raster — no forest object, no lake object — only cells that happen to share a value. If you want the forest as an object, you have to derive it, by grouping contiguous cells and converting them to a polygon.

Why both exist

Neither model is better; they answer different questions natively.

QUESTION	NATURAL FIT	WHY
Where are the fire hydrants within 200 m of this school?	Vector	Hydrants are countable objects with exact positions
What's the average slope of this watershed?	Raster	Slope is a value that exists everywhere, not a set of things
Which parcels intersect the flood zone?	Vector	Parcels have legal boundaries; you need crisp edges
How does temperature vary across the county?	Raster	Temperature has no edges; it varies continuously
How many customers live in each sales territory?	Vector	Customers are points, territories are polygons

QUESTION	NATURAL FIT	WHY
Where will water flow after a storm?	Raster	Flow is computed cell-to-cell across a terrain surface

Most real projects use both. A typical analysis might **buffer** vector wells (draw a zone extending a set distance around each one), overlay the buffers on a raster of groundwater depth, and summarize the raster values inside each buffer. ArcGIS moves between the two models constantly, with conversion tools in both directions — and each conversion loses something, which we'll return to at the end of the chapter.

Tip: When you meet a new dataset, your first question should be "vector or raster?" — before projection, before symbology, before anything. It determines which tools apply, how the data behaves when you zoom, and what kinds of questions it can answer. In ArcGIS, the layer's item page or its entry in the contents list will tell you: a feature layer is vector; an imagery layer is raster; a tile layer is a pre-rendered picture of a map, which may have started as either.

Vector Geometry: Points, Lines, and Polygons

Vector features come in three fundamental shapes, and a vector layer holds exactly one shape type. You will never find a single layer that mixes points and polygons — if a dataset seems to, it's actually several layers presented together.

Points

A point is a single coordinate pair: one X, one Y (plus optionally a Z for height — more on that shortly). Points represent things whose extent doesn't matter at your working scale: wells, crime incidents, tree locations, store addresses, GPS readings. A point has no length or area, which sounds obvious until you try to answer "how much of this park does the playground cover?" with a point layer and realize the model has already discarded the answer.

Lines

A line — ArcGIS usually says **polyline** — is an ordered sequence of coordinate points connected by straight segments. The individual coordinates along the way are called **vertices** (singular: vertex), and the endpoints are sometimes called **nodes**, a term you'll meet again when networks arrive. A smooth-looking river bend is actually dozens of tiny straight segments; add more vertices and the curve gets smoother, at the cost of a bigger file. Lines have length but no area: roads, streams, pipelines, transit routes, fault lines. A line can also carry direction — the vertex order runs from a start point to an end point — which matters enormously for stream networks and one-way streets, and not at all for a fence.

Polygons

A polygon is a closed ring of vertices enclosing an area: a lake, a census tract, a building footprint, a country. Polygons have both perimeter and area, and they introduce the first genuinely tricky geometry concepts:

- **Holes.** A polygon can have interior rings — a donut. A lake with an island is one polygon whose boundary includes an outer ring (the shoreline) and an inner ring (the island's shore).
- **Multipart features.** One feature can contain several disconnected shapes. Hawaii is typically stored as *one* polygon feature with many parts — one per island — because for most purposes you want one row — one "Hawaii" — in the table, not a row per island. The same idea applies to lines (one "Route 66" feature made of disconnected segments) and points (a "multipoint" feature holding a cluster of coordinates that share one set of attributes).

Multipart features trip people up constantly: you click what looks like a small island and the whole archipelago highlights, or a count of features returns 50 when the map clearly shows hundreds of shapes. Both are the multipart model working as intended. Tools exist in ArcGIS Pro to split multipart features into singleparts and to merge them back, and Chapter 13 (Editing Workflows) covers when you'd want each.

Z-values, M-values, and true curves

Two refinements you'll eventually encounter. Geometries can carry **Z-values** — a height at every vertex — turning a flat pipeline into a 3D one. They can also carry **M-values** ("measures"), which store a distance-along-the-line at each vertex, letting a highway

department say "the pothole is at mile 12.4" without splitting the road feature; this technique is called linear referencing. And while web layers generally deal in straight segments, ArcGIS Pro can store *true curves* — mathematically perfect arcs — which matter in surveying and cadastral (land-ownership) work. You don't need any of these yet; you need to not be startled when a dataset's description mentions them.

The Attribute Table: Where the Meaning Lives

Strip the attributes away and a vector layer is just anonymous shapes. The attribute table is what makes it *information*: one row per feature, one column per fact, with the geometry invisibly bound to its row. Select the row and the shape highlights on the map; select the shape and the row highlights in the table. They are two views of the same feature. In Map Viewer you can open it from a layer's options menu (**Layers** > ... > **Show table**); in ArcGIS Pro, right-click the layer and choose the attribute table.

Columns are called **fields**, and every field has a type — text, integer, decimal number, date, and a few others — which controls what can be stored and what operations make sense. You can average a decimal field; you cannot average a text field, even if it happens to contain numbers, which is a genuinely common data-cleaning headache. Field design — choosing types, constraining values with domains, relating tables to each other — is a deep enough topic to own Chapter 12 (Schema Design); for now, three ideas carry you a long way:

1. **The table is half the dataset.** A "roads" layer with no attributes can be drawn but not queried, filtered, styled by category, or analyzed by any property other than raw geometry. Most of the practical power of GIS — "show me only the arterial roads," "color parcels by assessed value" — is attribute work.
2. **Every layer has system fields.** ArcGIS maintains a unique ID for each feature (commonly named **OBJECTID**) and often auto-computed length or area fields. Don't fight them, and don't build your workflows on the assumption that the IDs are permanent — they can change when data is copied or republished.
3. **Attributes carry the definitions.** A field named **STATUS** with values 1, 2, and 3 is useless without knowing what the codes mean. That mapping lives in metadata, in a coded-value domain, or in someone's head — and only the first two survive contact with a new user.

Watch out: Numbers stored in text fields are one of the most common silent failures in GIS. The layer looks fine, the pop-ups look fine, and then sorting puts "100" before "20" and your map styles values alphabetically instead of numerically. If a numeric field misbehaves, check its type before anything else.

Rasters Up Close: Cells, Bands, and Resolution

A raster is defined by surprisingly little: an origin point, a cell size, a number of rows and columns, and the grid of values. Everything else follows.

Cell size is resolution. A raster with 10-meter cells stores one value for every 10×10 m patch of ground; whatever variation exists inside that patch is gone, averaged or sampled away at creation time. This is the single most important fact about any raster you use. A 30-meter land cover grid cannot tell you about a 15-meter garden, no matter how far you zoom in — zooming just makes the cells bigger on screen. When people say a raster is "high resolution," they mean small cells.

Values have types. Some rasters store continuous quantities — elevation, rainfall, reflectance — where cell values are measurements and arithmetic on them is meaningful. Others store categories — land cover classes, soil types — where the values are just codes, and averaging them would be nonsense (the "average" of water and asphalt is not a real land cover). ArcGIS treats these differently for styling and analysis, and so should you.

Bands stack. A raster can hold several aligned grids at once, called bands. A natural-color aerial image has three (red, green, blue); satellite sensors often capture additional bands, such as near-infrared, that are invisible to the eye but priceless for detecting vegetation health. Chapter 15 (Imagery and Rasters in Practice) covers band combinations and everything else practical here.

NoData is not zero. Cells where nothing was measured — outside the flight path, under cloud cover — hold a special NoData marker. Zero means "the measurement was zero"; NoData means "there is no measurement." Confusing them corrupts statistics: an average elevation that treats NoData as 0 will pull mountain ranges down toward sea level.

Watch out: Because rasters and photographs share a format family, it's easy to treat an analytical raster as "just an image" — recoloring it, exporting it to a picture file, resampling it for a slideshow. Every one of those steps can destroy the underlying values. Keep a bright line in your head between rasters as *data* (values you compute with) and rasters as *pictures* (colors you look at), and never do analysis on something that has made a round trip through an image editor.

Discrete vs Continuous: The Question Underneath

Behind the vector/raster split sits a deeper distinction about the phenomena themselves. Some things in the world are **discrete**: they are countable objects with more-or-less real boundaries — buildings, wells, road segments, administrative districts. Other things are **continuous**: they have a value at every location and no boundaries at all — elevation, temperature, air pressure, noise, probability of landslide.

Discrete phenomena map naturally to vector. Continuous phenomena map naturally to raster (or to other surface models we'll meet shortly). But the world loves to blur the line, and recognizing the blur is a professional skill:

- A *soil type* map looks discrete — crisp polygons, one type each — but soils actually grade into each other over meters or kilometers. The crisp boundary is a cartographic convenience, and analysis that leans hard on its exact position is building on sand, sometimes literally.
- A *city boundary* is genuinely discrete legally but continuous experientially; "urban-ness" fades outward. Which one you model depends on whether your question is about jurisdiction or about lived density.
- *Population* is made of discrete people, but is almost always handled as counts inside polygons (census units) or as a density surface — a continuous abstraction over discrete reality.

The habit to build: before modeling anything, ask *does this thing have real edges, or am I about to invent some?* Invented edges aren't wrong — all models invent — but you want to invent them on purpose.

Topology in Plain Terms

Topology is the branch of the model that cares about how features relate spatially: what touches what, what connects to what, what contains what. In everyday GIS conversation it means something concrete: **the rules and shared-geometry relationships that keep a dataset internally consistent.**

Consider two neighboring counties. Geometrically, they're two polygons. Topologically, they *share a boundary* — the same line, in reality, serves as the east edge of one and the west edge of the other. If your data stores that boundary twice, once per polygon, the copies can disagree by a hair's width. Where they overlap, you get slivers of land belonging to both counties; where they gap, you get slivers belonging to neither. These **sliver polygons** are the classic symptom of topology-free data, sometimes called *spaghetti data* — a pile of independent geometries with no knowledge of each other, like dropped noodles.

Topological awareness shows up in three practical places:

1. **While editing.** Snapping — the magnetic pull that makes your cursor lock onto an existing vertex or edge — is topology's front-line defense. So are shared-edge editing tools, where moving a county line moves it for *both* counties at once. Chapter 13 covers the workflows.
2. **As validation rules.** ArcGIS Pro lets you declare rules — "parcels must not overlap," "every stream must connect to another stream or a lake," "address points must fall inside a parcel" — and then audits the data against them, flagging violations for repair. This is the heart of Chapter 14 (Data Quality).
3. **As connectivity.** For networks — roads, pipes, wires — topology stops being about tidiness and becomes the entire point. A routing system doesn't care what streets look like; it cares which segments connect at which junctions. Two lines that visually cross but don't share a node (think: a highway overpass) must *not* connect, or your routes will teleport between road levels.

Tip: You don't need formal topology rules to benefit from topological thinking. Even in the simplest web-map editing, turning snapping on and tracing existing boundaries instead of redrawing them will save you from most of the sliver-and-

gap misery that plagues hand-digitized data (digitizing means tracing real-world shapes into the computer, by hand or from imagery).

Scale and Generalization: Why Data Has a "Zoom Level" Baked In

Every dataset was captured *for* a certain scale, and it silently misbehaves outside it.

Scale, in map terms, is the ratio between map distance and ground distance — 1:24,000 means one unit on the map is 24,000 on the ground. Confusingly, a *large-scale* map shows a *small* area in high detail (the fraction $1/24,000$ is larger than $1/1,000,000$). Web maps hide the ratio behind zoom levels, but the concept survives fully intact.

Why does scale get baked into data? Because capturing geography means **generalizing** — simplifying reality enough to be storable and legible:

- **Simplification.** A coastline digitized for a national map might place a vertex every few hundred meters; every cove smaller than that never enters the data. Zoom into a harbor and the "coastline" cuts blithely across docks and inlets. The data isn't wrong; it's answering at the scale it was built for.
- **Collapse.** At small scales, a city becomes a point, a river becomes a line, a divided highway becomes a single centerline. The same real object gets a different geometry type at different scales — which is why "what geometry is a river?" has no single answer.
- **Selection and omission.** Small-scale data simply leaves things out: minor roads, small ponds, buildings entirely. Related is the idea of a *minimum mapping unit* — the smallest real-world area a dataset promises to represent. A land cover product with a half-hectare minimum mapping unit will absorb your backyard pond into the surrounding category without comment.
- **Aggregation.** Individual things become summaries: houses become a "residential area" polygon; people become a count per census tract.

Rasters express the same truth through cell size — a 30-meter grid *is* a generalization decision — and imagery through ground resolution.

The practical consequences are blunt. Mixing a parcel layer built for 1:1,200 work with a road layer generalized for 1:100,000 display will show roads slicing through the

corners of houses; neither layer is broken, but their marriage is. Measuring the length of a generalized coastline understates it — perhaps wildly, since coastline length famously grows the finer you measure. And overlaying two layers digitized at very different scales produces boundary mismatches everywhere they should agree.

Watch out: Metadata usually states a source scale, a ground resolution, or an intended use — and it's the first thing to check before trusting data at close zoom. If a layer's boundaries look increasingly ragged, angular, or offset as you zoom in, you've zoomed past its native scale. Nothing in ArcGIS will stop you; the software happily displays any data at any zoom. Knowing when to stop is on you. Chapter 5 covers how to read these signals in a dataset's documentation.

Beyond the Big Two: Grids, TINs, and Networks

Vector and raster cover most of daily life, but a few specialized models earn their keep for particular jobs.

Regular grids and elevation surfaces

The plain raster grid is itself the workhorse model for terrain: a **digital elevation model (DEM)** is nothing more than a raster whose values are heights. From that one humble grid, ArcGIS derives slope, aspect (which compass direction a hillside faces), shaded relief, watersheds, flow paths, and visibility — the entire toolkit of Chapter 19 (Terrain and Raster Analysis). Grids win on simplicity: every cell the same size, neighbors trivially findable, arithmetic between aligned grids as easy as adding spreadsheets.

TINs: triangles where it matters

A **TIN — triangulated irregular network** — models a surface as a mesh of triangles stretched between measured points, like a faceted gem. Where a raster spends cells uniformly (flat farmland gets the same cell density as a canyon), a TIN spends vertices where the terrain demands them: dense triangles across broken ground, a few huge ones across a plain. TINs can also honor exact measured points and hard linear features — a surveyed ridgeline, the crisp top edge of a road cut — which a grid can only approximate. The trade-offs: TINs are heavier to process, mostly confined to ArcGIS Pro

and 3D work, and overkill for regional analysis. You'll meet them in engineering-adjacent work — cut-and-fill volumes, site design, precise drainage — and hardly anywhere else. When TIN-grade source data needs raster-grade convenience, the standard move is to build the TIN from survey points, then convert it to a DEM.

Networks: geometry plus connectivity

A **network** model is vector line data with formalized connectivity: edges (street segments, pipes, wires), junctions (intersections, valves, switches), and rules about how travel or flow moves through them — one-way restrictions, turn penalties, valve states. The geometry tells you where things are; the network tells you how they *function*. Two flavors dominate: transportation networks, where movement can go any legal direction (Chapter 18, Network Analysis, lives here), and utility networks, where flow has sources, sinks, and direction — water flows from the plant outward, and shutting one valve isolates a predictable set of pipes. Building network datasets is specialist work; *using* them, via routing and service-area tools, is thoroughly approachable.

A note on 3D and beyond

Modern ArcGIS also handles point clouds (billions of raw points from lidar, or aerial laser scanning), 3D object models of buildings, and voxel grids for volumetric data like ocean temperature at depth. These are variations on themes you now know — point clouds are points at extreme density, voxels are rasters with a third dimension — and none require new mental furniture yet.

Choosing a Representation: A Field Guide

So: a real-world thing lands on your desk, and you must pick its data model. The question that decides everything is not "what is this thing?" but "**what do I need to do with it?**" The same phenomenon takes different representations for different jobs, and all of them can be correct.

Work through four questions:

1. **Is the phenomenon discrete or continuous?** Countable objects with meaningful edges lean vector; everywhere-valued fields lean raster. When it's genuinely both

(population, soils), let the *question* pick: jurisdictional and per-object questions want vector; gradient and surface questions want raster.

2. **At what scale will it be used?** This sets geometry type for discrete things (city as point vs polygon; river as line vs polygon) and cell size for continuous ones. Model for the most detailed scale you'll honestly need — you can generalize detailed data downward, but you cannot manufacture detail that was never captured.
3. **What will you measure?** If you'll need areas, points and lines are disqualified. If you'll need routing, you need network-ready lines with real connectivity, not spaghetti. If you'll need per-object attributes ("this building's year built"), a raster can't hold them — attributes belong to features, and a raster has no features.
4. **What form does the source data arrive in, and what do your tools expect?**
Sometimes the model is chosen for you: census data arrives as polygons, lidar arrives as points, satellite data arrives as rasters. Conversion is always possible and never free — vector-to-raster snaps crisp boundaries to a cell grid; raster-to-vector invents crisp edges where the data had gradients. Convert when a tool demands it, as late in the workflow as you can, and keep the original.

Here's how common phenomena typically shake out — with the caveat that "typically" is doing real work in that sentence:

REAL-WORLD THING	COMMON REPRESENTATION	THE MAIN ALTERNATIVE, AND WHEN
Customer locations	Points	Aggregated to polygons or a density raster for privacy or pattern analysis
Roads	Lines (network-enabled for routing)	Polygons for pavement-management at engineering scales
Rivers	Lines	Polygons at large scales where width matters; part of a raster flow model for hydrology

REAL-WORLD THING	COMMON REPRESENTATION	THE MAIN ALTERNATIVE, AND WHEN
Land ownership (parcels)	Polygons	Essentially never anything else — legal edges are the point
Elevation	Raster DEM	TIN for engineering precision; contour lines for display
Land cover	Categorical raster	Polygons when clean per-area reporting matters more than gradient fidelity
Rainfall	Raster surface, interpolated (estimated between measurement stations)	The original gauge readings stay as points — keep both
Buildings	Polygon footprints	Points at small scales; 3D objects for city modeling
Administrative boundaries	Polygons	Lines only when the boundary itself (not the area) is the subject

Tip: When two representations both seem defensible, prefer the one that preserves more information, and derive the other from it. Points can always be aggregated into polygons; polygons cannot be disaggregated back into the points they swallowed. Detailed geometry can be simplified; simplified geometry cannot be re-detailed. Store rich, publish light — a principle that returns with force in Chapter 10 (Hosted Feature Layers) when performance enters the picture.

The Model Is the Message

Everything downstream in this Compendium sits on the foundations laid here. When Chapter 3 wrestles coordinate systems, it's deciding what the numbers in vector geometry *mean*. When Chapter 7 styles layers, the options on offer are dictated by geometry type and by whether raster values are categories or quantities. When

Chapters 16 through 20 run analysis, every tool's input requirements are really data-model requirements in disguise. And when something goes wrong — Chapter 39's territory — the diagnosis usually starts with the question this chapter taught you to ask first: what is this data, really, underneath the picture?

Hold on to the central discipline: a GIS layer is not the world. It is somebody's answer to "how shall we represent this, at this scale, for this purpose?" Learn to read that answer in every dataset you open — vector or raster, crisp or continuous, generalized to what scale, topologically sound or spaghetti — and you'll spend your career working *with* your data's grain instead of against it.

The ArcGIS Ecosystem: Every Product and When to Use It

Esri, the company behind ArcGIS, sells what looks at first like several dozen different products. There are desktop programs, web apps, phone apps, app builders, developer kits, and a bewildering pile of names that seem to change every couple of years. Here is the secret that makes all of it manageable: ArcGIS is not dozens of products. It is one platform with two centers of gravity — a cloud portal and a desktop application — and everything else is a doorway into one or both of them.

Once you see that shape, every product name snaps into place. The mobile apps are doorways for people standing in a field. The app builders are doorways for people looking at a browser. The developer products are doorways for programmers. They all read from and write to the same shared pool of maps and data. Learn the shape once, and you will never again be intimidated by an Esri product announcement.

The Two Centers of Gravity

ArcGIS Online: the cloud portal

ArcGIS Online is a website — arcgis.com — run entirely by Esri. You sign in, and you get a personal (or organizational) workspace in the cloud. Everything you make or upload lives there as an **item**: a map is an item, a layer of data is an item, an app is an item,

even a PDF you upload is an item. Each item has its own page with a description, settings, and sharing controls, a bit like a document in Google Drive.

The most important kind of item is the **hosted feature layer** — your data (points, lines, polygons, and their attribute tables) stored on Esri's servers and served out to any map or app that asks for it. When you built your first map, you were almost certainly working in ArcGIS Online's Map Viewer on top of hosted layers, whether you realized it or not. Chapter 6 (Map Viewer: The Complete Reference) covers the map-making side in depth, and Chapter 10 (Hosted Feature Layers) covers the data-hosting side.

ArcGIS Online is where sharing happens. Every other product in the ecosystem ultimately publishes to it, reads from it, or both. Think of it as the town square.

ArcGIS Pro: the desktop workhorse

ArcGIS Pro is a program you install on a Windows computer. It is the heavy machinery of the platform: full-strength analysis with hundreds of **geoprocessing** tools (Esri's word for the data-processing and analysis operations that take layers in and produce new layers out), precision editing, print-quality cartography, 3D scenes, and the ability to work with data on your own hard drive or in corporate databases — no internet required for most of it.

Pro and Online are not rivals; they are partners. You sign in to Pro with the same account you use on arcgis.com, and Pro can pull layers straight from your portal or push finished work up to it. A very common rhythm is: do the hard analytical work in Pro, publish the results to ArcGIS Online, then share them through a web app. Volume E of this compendium (Chapters 21 through 25) is devoted entirely to Pro.

A note on history you will run into: Pro replaced an older desktop program called **ArcMap**, which Esri has retired. Tutorials and forum posts about ArcMap still litter the internet. The concepts usually transfer; the click paths almost never do.

Tip: If you are ever unsure whether a task belongs in Pro or Online, ask two questions. Does it involve heavy computation or large local files? Pro. Does it involve sharing with other people? Online. Tasks that involve both — analyze then share — start in Pro and end in Online.

ArcGIS Enterprise: the portal you run yourself

ArcGIS Enterprise is, functionally, ArcGIS Online that you install on your own servers. Same portal concept, same items, same hosted layers, and most of the same app builders — but running on machines your organization controls, behind your firewall if you want. (A few pieces remain Online-only, Hub among them.)

Why would anyone take on that burden? Three common reasons: data that legally or contractually cannot live in a public cloud; integration with internal systems and databases that the outside world must never touch; and workloads so heavy or specialized that renting Esri's cloud does not fit. Enterprise is a serious commitment — it needs server administrators the way ArcGIS Online needs none. As a solo learner or a small team, you will almost certainly live in ArcGIS Online and can treat Enterprise as trivia. Chapter 35 (ArcGIS Enterprise and When You Need It) gives it the full treatment, including the honest answer to "do I need it?" (usually: no).

Throughout this book, when you see the word **portal**, it means "ArcGIS Online or Enterprise, whichever you use" — the two are deliberately interchangeable in most workflows.

The Mobile Apps: Getting Data In From the Field

Three Esri apps handle field data capture on phones and tablets, and all three exist to solve the same problem — capturing data where the data actually is — with three different philosophies. All three write their results into hosted feature layers in your portal, which means the moment a field worker taps Submit, the office can see the new point on a map. Chapter 30 (Field Operations) walks through complete field workflows; here is the lay of the land.

ArcGIS Field Maps: the map-first app

Field Maps puts a map on the field worker's screen — typically a map you prepared in advance in Map Viewer — and lets them add and edit features directly on it. Tap a location, fill in the attributes, attach a photo, done. It supports taking maps offline (download an area before leaving cell coverage, sync when you return), high-accuracy GPS receivers, and location tracking for crews. If the field task is fundamentally "interact with a map" — inspect these hydrants, map this trail, update these parcels — Field Maps is the tool.

One naming trap: Field Maps absorbed several older Esri apps (Collector, Explorer, and Tracker among them). Older tutorials referencing those names are describing what is now Field Maps.

ArcGIS Survey123: the form-first app

Survey123 inverts the philosophy: the field worker sees a form, not a map. Questions flow top to bottom — multiple choice, text, numbers, photos, signatures, and a location question that quietly captures the map part. Its superpower is **smart forms**: questions can appear or disappear based on earlier answers, values can be validated as they are typed, and calculations can run inside the form. If your task sounds like "conduct an inspection," "administer a questionnaire," or "fill out an incident report," Survey123 fits better than Field Maps. It also works in a plain web browser, which means respondents do not need to install anything or even have an ArcGIS account — that makes it Esri's answer to public surveys.

ArcGIS QuickCapture: the big-button app

QuickCapture strips field collection to its absolute minimum: a screen of large buttons, one tap per observation. Tap "Pothole" and the app records a pothole at your current location with a timestamp — no form, no map interaction, no stopping. It is designed for capture at speed: from a moving vehicle, from a helicopter, while walking a survey line, with gloves on. You trade richness for velocity.

APP	WORKER SEES	BEST FOR	TYPICAL PACE
Field Maps	A map	Editing and inspecting features in place	Minutes per feature
Survey123	A form	Structured questionnaires and inspections	A form at a time
QuickCapture	Big buttons	Rapid observations in motion	Seconds per record

Tip: These apps combine well. A crew might use QuickCapture to flag issues at speed from a moving vehicle, then return with Field Maps to inspect each flagged site in detail. Both write to layers in the same portal, so the handoff is just a shared map.

The App Builders: Sharing Without Writing Code

A **web map** — the saved, shareable map item you assemble in Map Viewer — is a fine thing on its own, but most audiences need it wrapped in something: a title, some explanation, filters, charts, a layout that guides them. Esri provides a ladder of no-code app builders, ordered here from simplest to most flexible. All of them consume web maps you have already made, and all of them produce a shareable URL. Volume F of this compendium gives each one a full chapter.

ArcGIS Instant Apps: pick a template, done

Instant Apps is a gallery of pre-built app templates — a simple viewer, a side-by-side comparison, a countdown of "nearest locations," an attachment gallery, and a couple dozen more. You pick a template, point it at your web map, toggle a handful of options, and publish. Five minutes from map to app is realistic. The trade-off is rigidity: you get exactly what the template offers. When someone says "can you just put this map on a page people can open?", Instant Apps is the answer. See Chapter 26.

ArcGIS StoryMaps: maps woven into narrative

StoryMaps is a builder for scrolling, long-form pages that mix text, images, video, and live maps — think of a magazine feature where the maps are interactive. It shines when the point is persuasion or explanation: a report on neighborhood change, a guided tour, a project retrospective. If your audience should *read* rather than *explore*, reach for StoryMaps. See Chapter 27.

ArcGIS Dashboards: live numbers at a glance

Dashboards arranges a screen into panels — a map, plus charts, gauges, counters, and lists — all fed live from your layers. When the underlying data changes (say, a Survey123 submission arrives), the dashboard updates. It is the tool for operations views and monitoring: shelter capacity during a storm, daily inspection counts, fleet

status. If the audience's question is "what is the current state of things?", it is a dashboard. See Chapter 28.

ArcGIS Experience Builder: the flexible one

Experience Builder is the drag-and-drop construction kit: blank canvas, a library of widgets (maps, lists, filters, charts, text, buttons), multiple pages, and control over layout on desktop and phone separately. It can reproduce most of what the other builders do and combine them — a page with a map, a survey, and a dashboard-like sidebar, wired together so clicking one thing filters another. The cost is time: real flexibility means real configuration work. Reach for it when the simpler builders almost-but-not-quite fit. See Chapter 29.

You may also encounter **Web AppBuilder**, an older builder that Experience Builder replaces; treat it as legacy and build new work in Experience Builder.

ArcGIS Hub: a website for engagement

Hub sits slightly apart: rather than building one app, it builds a *site* — a branded web presence with pages, searchable open-data catalogs, and embedded maps and apps. Local governments use it to publish open data portals; organizations use it to run engagement initiatives where the public can find data, take surveys, and follow projects. A basic tier of Hub comes with ArcGIS Online organizations; a premium tier (adding features for managing community members and events) is a paid add-on. If your goal is "a website where the public finds all our maps and data," that is Hub.

BUILDER	ONE-LINE PITCH	REACH FOR IT WHEN
Instant Apps	Template, map, publish	You need a simple app fast
StoryMaps	Narrative with maps inside	The audience should read a story
Dashboards	Live panels of charts and counts	The audience monitors current status
Experience Builder	Blank-canvas widget kit	Nothing simpler fits the requirement

BUILDER	ONE-LINE PITCH	REACH FOR IT WHEN
Hub	A whole engagement website	The public needs a front door to many things

Watch out: The builders overlap on purpose, and it is easy to burn a week building in Experience Builder what Instant Apps would have delivered in an afternoon. Work up the ladder from simplest: only move to a more powerful builder when a concrete requirement forces you to.

The Developer Products: Doorways for Programmers

You can go very far in ArcGIS without writing code, and this book mostly assumes you will. But the programmable layer is worth knowing about, because it explains what is possible when the no-code tools run out.

Maps SDKs: build your own apps

An **SDK** (software development kit) is a library of building blocks programmers use inside their own applications. The **ArcGIS Maps SDK for JavaScript** is the flagship: it is what lets a web developer embed live, styled, interactive ArcGIS maps inside any website — in fact, Esri's own Map Viewer and app builders are built on it. Chapter 33 offers a gentle primer. Parallel **Maps SDKs for native apps** exist for building iOS, Android, and desktop applications, and there are even kits for game engines. The common thread: they all talk to the same portal items and hosted layers you have been making all along.

Python: arcpy and the ArcGIS API for Python

Two Python libraries with confusingly similar names serve two different jobs. **arcpy** ships inside ArcGIS Pro and automates Pro's geoprocessing tools — if you find yourself running the same twelve-step analysis every Monday, arcpy turns it into a script. The **ArcGIS API for Python** is a separate, free library that talks to your *portal*: it can search items, publish layers, manage users, and run analysis in the cloud, from any machine with Python — no Pro required. Chapter 31 covers both and when to use which.

REST services: the plumbing everything shares

Underneath every product in this chapter, data moves over the web as **REST services** — a standard way for software to request data over URLs. You rarely need to think about this, but it is the reason the ecosystem hangs together: a hosted feature layer has a URL, and Pro, Field Maps, Dashboards, and a custom JavaScript app can all read that same URL simultaneously. Chapter 32 shows how to peek under this hood, which turns out to be one of the most useful troubleshooting skills in the platform.

Living Atlas: Data You Do Not Have to Make

The **ArcGIS Living Atlas of the World** is Esri's curated library of ready-to-use layers: **basemaps** (the ready-made background maps your own layers draw on top of), satellite and aerial imagery, demographics, boundaries, live weather and traffic feeds, terrain, land cover, and thousands more, contributed by Esri and vetted partners. It is built into the platform — in Map Viewer, when you add a layer and switch the source from your own content to Living Atlas, you are shopping in it.

Living Atlas matters strategically: the fastest way to make a rich map is to make only *your* layer and borrow everything else. Most Living Atlas content is free to use with any organizational account; a minority is **premium content** whose use consumes credits (more on credits below), and layers are labeled accordingly. Chapter 5 (Finding Data) covers how to search it well and how to judge what you find.

Accounts, User Types, and Licensing in Plain Terms

Esri's licensing has a reputation for opacity. The core of it fits in a few paragraphs.

Public account versus organizational account

A **public account** is free, meant for personal and noncommercial use, and deliberately limited: you can make maps and use free content, but the serious machinery — publishing hosted feature layers, running analysis, using premium content, most of the app builders' output for real audiences — is fenced off. It is a sandbox, and a good one for a hobbyist.

An **organizational account** means you belong to an ArcGIS Online **organization**: a paid subscription, administered by someone (perhaps you), with a pool of member accounts, shared groups, security settings, and a monthly allotment of credits. Everything this

book teaches assumes an organizational account. If you have one through work or school, you are set; if you are solo, Esri sells small subscriptions and steeply discounted personal-use and student options that include both ArcGIS Online and Pro.

User types: what your seat can do

Within an organization, every member is assigned a **user type** — essentially the size of the toolbox their seat includes. The names have shifted over the years (Esri renames these more often than anyone would like), but the ladder is stable in concept:

RUNG ON THE LADDER	WHAT THAT SEAT IS FOR
Viewer-level	Look at maps and apps others made; no creating or editing
Field/mobile-level	Use the mobile apps to collect and edit data; not much else
Creator-level	The full web platform: make maps, layers, and apps
Professional-level	Everything Creator has, plus ArcGIS Pro

Two practical consequences. First, licenses are per-person ("named user"), so organizations mix and match — a public-works department might hold two Professional seats, five Creators, and thirty field-level seats. Second, if a colleague complains they "can't publish a layer," the very first thing to check is their user type; a viewer-level seat cannot, no matter what buttons they hunt for.

Alongside user types sit **roles**, which control *permissions* rather than capability — whether a member may share publicly, create groups, or administer the organization. User type is the toolbox; role is the set of rules about using it. And beyond both sit **add-on licenses** for specialist tools (Pro extensions for advanced spatial statistics, imagery, networks, and so on), which are purchased and assigned individually. Chapter 34 (Administration) covers managing all of this.

Watch out: ArcGIS Pro's advanced capabilities come in tiers (commonly described as basic, standard, and advanced levels) plus optional extensions. If a Pro tutorial's tool is grayed out for you, the likely cause is a licensing tier or an

unassigned extension — not a bug. The tool's documentation page always states what license it needs.

Credits: the platform's metered currency

Credits are the unit ArcGIS Online uses to meter consumption of Esri's cloud resources. Your organization receives an allotment with its subscription and can buy more. Think of them like a utility meter attached to specific activities — not a paywall on everything.

What consumes credits, qualitatively:

- **Storage** of hosted content — an ongoing trickle proportional to how much you store. Feature-layer storage is metered at a much higher rate per gigabyte than plain file storage, which is why Chapter 10's advice about lean layers doubles as financial advice.
- **Premium services** — batch geocoding (turning many addresses into map points), routing and drive-time calculations, and demographic enrichment each bill a small number of credits proportional to how much you ask for.
- **Spatial analysis run in the browser** — the analysis tools inside Map Viewer execute on Esri's servers and charge accordingly, scaled to the size of the job.
- **Some hosted compute**, such as cloud-hosted notebooks (browser-based Python coding environments in your portal) at their more powerful settings.

What does *not* consume credits is just as important: viewing maps, using standard basemaps and most Living Atlas layers, editing features in existing layers (field collection does not burn credits per point), and — significantly — any analysis you run locally in ArcGIS Pro on your own machine. One catch: Pro tools that reach out to Esri's online services — geocoding a table of addresses, requesting drive times — consume credits even though you launched them from the desktop. A modest allotment goes a long way for a small organization, but a single careless act (uploading an enormous dataset as a feature layer, or geocoding a giant address file twice) can bite.

Tip: Administrators can see a credit-usage report and set per-member credit budgets in the organization's settings. If you are about to run something

unfamiliar that might be credit-hungry, run it once on a tiny subset of data first and check the meter. Chapter 34 covers credit budgeting in detail.

The Decision Guide: Which Product for Which Job

Bookmark this table. The left column is the job in plain words; the right columns are where to do it and where this book covers it.

I WANT TO...	USE	COVERED IN
Make an interactive web map	Map Viewer in ArcGIS Online	Ch 6
Store data others can use in maps and apps	Hosted feature layers	Ch 10
Run heavy analysis or work with local files	ArcGIS Pro	Vols D, E
Make a print-quality map or map series	ArcGIS Pro layouts	Ch 24
Collect field data on a map	Field Maps	Ch 30
Run a structured survey or inspection form	Survey123	Ch 30
Log rapid observations from a moving vehicle	QuickCapture	Ch 30
Put a map on a simple public web page	Instant Apps	Ch 26
Tell a narrative story around maps	StoryMaps	Ch 27
Monitor live operations with charts and counts	Dashboards	Ch 28
Build a custom multi-widget web app, no code	Experience Builder	Ch 29
Give the public a data-and-engagement website	Hub	this chapter

I WANT TO...	USE	COVERED IN
Find ready-made data and basemaps	Living Atlas	Ch 5
Automate portal chores or Pro workflows	Python (ArcGIS API / arcpy)	Ch 31
Embed maps in software I am programming	Maps SDK for JavaScript	Ch 33
Run the whole platform on my own servers	ArcGIS Enterprise	Ch 35
Administer users, groups, and credits	Organization settings	Ch 34

When a job seems to fit two products, prefer the simpler and more specialized one. The ecosystem rewards using each tool for the thing it was built for, and the shared portal means you can always chain tools rather than forcing one to do everything.

One Workflow, Many Doorways

To make the shape concrete, follow one project through the ecosystem — the same arc Chapter 36's worked project takes at full length.

A small city wants to track and fix damaged sidewalks. A GIS analyst designs a "sidewalk reports" layer and publishes it as a hosted feature layer in the city's ArcGIS Online organization. She builds a Survey123 form so inspectors — holding field-level seats and phones — can file structured reports with photos; each submission lands in the hosted layer instantly. Back at the office, a Dashboards app counts open reports by neighborhood on a wall monitor. Once a quarter, the analyst pulls the accumulated data into ArcGIS Pro, runs a hot-spot analysis (see Chapter 17 for the method), and overlays demographic context from a Living Atlas layer to see which affected neighborhoods have the most foot traffic. She wraps those findings in a StoryMap for the city council, and the public sees a filtered Instant App on the city's Hub site showing which repairs are complete.

Count the products: eight. Count the datasets: one. That is the ecosystem working as designed — a single shared pool of data in the portal, with each audience given the doorway that fits them. Every chapter that follows teaches you one of those doorways in depth; you now know how they all connect.

Coordinate Systems and Projections, Complete

Sooner or later, every GIS user hits the moment: two layers that should sit on top of each other don't. The roads are a few hundred feet east of the parcels. The survey points are floating in the ocean off the coast of West Africa. The county boundary looks right until you zoom in, and then everything is subtly, maddeningly shifted. Nothing else in GIS produces as much quiet panic as misaligned data, and nothing else has a more learnable explanation. Every one of those symptoms traces back to coordinate systems — the machinery that translates "a spot on the curved surface of the Earth" into "a pair of numbers a computer can draw." This chapter gives you the whole machine: what geographic and projected coordinate systems actually are, what a datum is in plain language, why the web runs on a projection cartographers love to hate, how ArcGIS quietly reprojects your data behind your back, and a step-by-step diagnostic routine for the day your layers stop lining up.

The good news up front: you do not need any of the trigonometry. You need about eight concepts, a healthy respect for one specific trap in ArcGIS Pro, and a checklist. That's what follows.

Geographic Coordinate Systems: Addresses on a Sphere

A **geographic coordinate system** (often abbreviated GCS) describes locations on the curved surface of the Earth using angles: **latitude** (how far north or south of the equator you are) and **longitude** (how far east or west of a reference line through Greenwich, England). Both are measured in degrees. Latitude runs from -90 at the South Pole to +90 at the North Pole; longitude runs from -180 to +180, with negative numbers meaning west. Seattle sits at roughly latitude 47.6, longitude -122.3 — that is, 47.6 degrees north and 122.3 degrees west.

This feels like a complete answer to "where is it?", and for casual purposes it is. GPS receivers report latitude and longitude. Web APIs pass them around. When someone hands you a spreadsheet of coordinates, this is almost always what's in it (see Chapter 11, Creating Data, for turning that spreadsheet into a layer).

But there are two catches, and they drive the rest of this chapter.

First, degrees are angles, not distances. A degree of latitude is always about 111 kilometers, but a degree of longitude shrinks as you move toward the poles — the lines of longitude converge until they all meet at a point. At the equator a degree of longitude is about 111 km; in Anchorage it's roughly half that. So you can't do honest math directly on raw latitude-longitude values. "This polygon spans 0.5 degrees, so it's X kilometers wide" is a trap; the answer depends on where on the planet the polygon sits.

Second — and this surprises almost everyone — latitude and longitude are not absolute. They only mean something relative to a specific mathematical model of the Earth's shape. That model is called a datum, and it's the next piece.

Datums: Which Version of the Earth You're Using

The Earth is not a sphere. It's slightly squashed at the poles, bulges at the equator, and lumps and sags irregularly on top of that because gravity varies from place to place. Mapmakers handle this by approximating the planet with a smooth mathematical shape — a slightly flattened sphere called an **ellipsoid** — and then pinning that ellipsoid to the real Earth in a specific way. A **datum** is exactly that: a chosen ellipsoid plus the decision about how it's anchored to the physical planet. Change the anchoring, and the latitude-longitude of every point on Earth changes slightly, even though nothing physically moved.

Think of it like calibrating a bathroom scale. Two scales can be perfectly consistent internally and still disagree with each other by a pound because they were zeroed differently. Two datums are two calibrations of the coordinate grid.

The datums you'll actually meet:

DATUM	PLAIN-LANGUAGE DESCRIPTION	WHERE YOU'LL ENCOUNTER IT
WGS 1984 (WGS84)	The global datum used by GPS satellites and most of the web. Tied to the Earth as a whole, so coordinates in it slowly drift as continents move.	GPS devices, web maps, most global datasets, anything with the well-known ID 4326

DATUM	PLAIN-LANGUAGE DESCRIPTION	WHERE YOU'LL ENCOUNTER IT
NAD 1983 (NAD83)	The North American datum. Anchored to the North American tectonic plate, so coordinates of fixed objects on that plate stay put over time.	US government data, state and county GIS, surveying, State Plane coordinates
NAD 1927 (NAD27)	NAD83's predecessor, built from ground surveys anchored at a ranch in Kansas. Obsolete but still lurking in old datasets.	Historical data, old USGS maps, legacy oil-and-gas records

Here is the practical payoff. WGS84 and NAD83 were nearly identical when NAD83 was defined, but because one is fixed to the whole Earth and the other to a drifting continental plate, they have slowly diverged — today they disagree by roughly a meter or two across the continental United States, and the gap grows a little every year. For a national-scale web map, a meter is invisible and you can treat the two as interchangeable. For utility work, parcel boundaries, or anything survey-grade, a meter or two is the difference between the water main and the gas line.

NAD27 versus NAD83 is a much bigger jump — tens to a couple hundred meters depending on where you are. If your "old county data" lands consistently offset from everything else by roughly a city block, NAD27 is the prime suspect.

Watch out: "It's all lat-long, so it all matches" is the most common false belief in GIS. Two lat-long datasets on different datums are two different coordinate systems, and mixing them without a proper transformation produces small, consistent, hard-to-spot offsets — the worst kind of error, because the map looks almost right.

There are also **vertical datums**, which do the same calibration job for elevation ("height above what, exactly?"). They matter for terrain and flood work and are covered where they bite, in Chapter 19 (Terrain and Raster Analysis).

Projected Coordinate Systems: Flattening the Globe

A geographic coordinate system describes the round Earth. Your screen, and every printed map, is flat. A **projection** is the mathematical recipe for flattening — for converting angles on an ellipsoid into x and y positions on a plane. A **projected coordinate system** (PCS) is a complete package: a geographic coordinate system (including its datum) plus a projection recipe plus linear units, usually meters or feet. Every projected system is built on top of a geographic one; you never escape the datum question by projecting, you just add a flattening step on top of it.

Flattening a globe is like flattening an orange peel: you cannot do it without tearing or stretching something. Every projection distorts, and the entire craft consists of choosing *what* to distort and *where*. The cruel mathematics is that no flat map can stay truthful about everything at once — each projection family chooses what to keep:

WHAT'S KEPT TRUE	AT WHAT COST	FAMILY NAME	TYPICAL USE
Shape (locally): angles and small shapes look right	Areas lie	Conformal	Navigation, topographic maps, most local mapping
Area: regions compare truthfully in size	Shapes stretch	Equal-area	Density maps, statistics, "how much land" questions
Distance: true distance from specific points or lines	Everything else, elsewhere	Equidistant	Range rings, "how far from here" maps
Nothing perfectly	A pleasing balance of moderate lies	Compromise	World maps for general display

Two rules of thumb carry you a long way. First, distortion is smallest near wherever the projection touches the globe (its center or standard lines) and grows as you move away — which is why local projections for a single state can be extremely accurate, while any single projection of the whole world is badly distorted somewhere. Second, the projection that makes a map look right and the projection that makes the math come out right are often different projections. The last section of this chapter is about managing exactly that tension.

Every coordinate system in ArcGIS — geographic or projected — carries a **well-known ID** (WKID), a stable catalog number. WGS84 is 4326. Web Mercator is 3857. These numbers are the fastest, least ambiguous way to communicate a coordinate system to another person or to search for one inside ArcGIS.

Tip: When someone sends you data, the single most useful question you can ask is "what's the coordinate system — do you know the WKID?" It short-circuits an hour of guessing.

Web Mercator: The Default You Didn't Choose

Open Map Viewer, add a basemap, and you are in **Web Mercator** (WKID 3857) — the projected coordinate system used by essentially every consumer web map since the mid-2000s. Google adopted it, everyone else followed, and Esri's basemap tiles are built in it. Unless you deliberately choose otherwise, every web map you make lives here.

Web Mercator is a variant of the classic Mercator projection, a conformal projection invented for sea navigation in the 1500s. Its virtues are real: north is always straight up, small shapes look correct everywhere (streets meet at true angles, buildings aren't skewed), and the world fits into a square that can be chopped into tiles and cached at every zoom level — which is precisely why the web loves it. Panning and zooming a tiled Web Mercator map is fast because the hard work was done once, in advance (Chapter 10 covers how hosted layers and tiles exploit this).

Its vice is equally real: to keep shapes true, Mercator inflates size with latitude, and the inflation is dramatic. Greenland renders about as large as Africa; in truth Africa is roughly fourteen times bigger. Features at 60 degrees latitude are stretched to about twice their true scale; near the poles the stretching runs away entirely, which is why Web Mercator maps simply cut off before reaching them.

What this means for you, concretely:

- **For display of local areas, it's fine.** At city or county scale, the distortion within your view is uniform and invisible. Most web maps never need to leave it.

- **For visual comparison across latitudes, it lies.** A choropleth (a map that shades regions by a statistic), or anything where the audience will eyeball "which region is bigger," misleads on Web Mercator. Consider an equal-area projection for that map (Chapter 4, Cartographic Design, discusses when the projection is the design decision).
- **For measurement and analysis, never trust its raw numbers.** Planar areas and lengths computed in Web Mercator are wrong nearly everywhere, and badly wrong away from the equator. The fix — geodesic measurement — is a section of its own below.

Watch out: The most common analysis error in web GIS is computing area in Web Mercator and not noticing the result is inflated. A parcel in Minneapolis measured planar in Web Mercator comes out roughly double its true size. If an area number surprises you, check the projection before you check anything else.

UTM and State Plane: The Local Workhorses

If Web Mercator is the display default, **UTM** and **State Plane** are the projections where real local work happens.

Universal Transverse Mercator (UTM) carves the world into 60 north-south strips, each 6 degrees of longitude wide, and gives each strip its own tuned projection with coordinates in meters. Because each zone is narrow, distortion inside a zone stays small — about a tenth of a percent at worst — which makes UTM an excellent general-purpose choice for measurement and analysis anywhere on Earth. You just need to know your zone (Seattle is in zone 10, New York in 18; any "UTM zone map" will tell you yours) and whether you're in the northern or southern hemisphere variant. The weakness is the flip side of the strength: the tuning only holds inside the zone. A study area that straddles a zone boundary should usually use one zone for the whole area, or a different system entirely.

State Plane is the United States' surveying-grade system: each state is divided into one or more zones — over a hundred nationwide — each with a projection chosen to fit that zone's shape (zones that run east-west get one flavor, north-south zones another). Distortion within a zone is even lower than UTM's. State Plane is the native language of

county parcel data, city engineering, and American surveyors, and it comes with a quirk you must respect: many State Plane definitions use **US survey feet** rather than meters. A units mismatch — data in feet drawn as if it were meters — produces one of the most recognizable failure signatures in GIS, covered in the diagnosis section below.

The mental model: Web Mercator for the web's display plumbing; UTM when you need honest meters anywhere on the planet; State Plane when you're working with American local-government data or need to match surveyors. All three are projected systems sitting on top of a datum — UTM zones come in WGS84 and NAD83 flavors, and State Plane is almost always NAD83, so the datum question from earlier never goes away.

On-the-Fly Reprojection: Why Things Usually Just Work

Here's a fact that explains a great deal of ArcGIS behavior: a map has one coordinate system, and every layer keeps its own. When you add a layer whose coordinate system differs from the map's, ArcGIS does not convert your data. It **reprojects on the fly** — recomputing where each feature should be drawn, every time it draws, while leaving the stored data untouched.

This is why you can drag a NAD83 State Plane parcel layer into a Web Mercator web map and it lands in the right place. It's a genuinely great convenience, and it's on by default everywhere: Map Viewer does it silently (the basemap's coordinate system — normally Web Mercator — sets the map's system, and everything else bends to match), and ArcGIS Pro does it too, with one important difference: Pro will warn you when the bending requires a datum change, which we'll come back to.

The convenience has three costs worth knowing:

Performance. Reprojecting every feature at every redraw is work. For small layers it's negligible; for a heavyweight layer redrawn constantly, storing the data in the map's coordinate system can noticeably speed things up (Chapter 10 covers this among the hosted-layer performance levers).

Silent datum shifts. On-the-fly display between different datums needs a transformation (next section), and if the software picks a default or skips a subtle one, layers can display with a small offset — real data misalignment that exists only in the drawing, not in the data.

A false sense of interchangeability. Because display "just works," it's easy to believe coordinate systems no longer matter. They matter exactly where display stops and computation begins: measurement, geoprocessing (ArcGIS's word for running analysis tools on data), and editing. On-the-fly projection fixes what you see, not what you *calculate*.

In ArcGIS Pro, the map's coordinate system lives in the map properties — right-click the map's name in the Contents pane and choose **Properties > Coordinate Systems**. A map takes its coordinate system from the first layer added to it — usually the default basemap, which is why new maps typically open in Web Mercator. Start a map without a basemap, though, and whatever layer you add first sets the system, so the map you build tomorrow may not match the map you built today.

Tip: In Pro, the current map units and coordinate readout sit on the status bar at the bottom of the map view. Glancing at it is the fastest way to confirm what system you're actually working in — and hovering over any location shows you live coordinates in that system.

Measuring the World: Geodesic vs Planar

There are two ways to measure distance and area in a GIS, and they can disagree wildly.

Planar measurement does geometry on the flat, projected map: the distance between two points is a ruler laid across the paper. It inherits every distortion of the projection. In a well-chosen local projection (your UTM zone, your State Plane zone), planar measurements are excellent. In Web Mercator away from the equator, they're fiction.

Geodesic measurement ignores the flat map and computes along the curved surface of the ellipsoid itself — the true shortest path over the Earth. It's the honest answer regardless of projection. The classic demonstration is a flight path from Seattle to London: geodesically it arcs far north over Greenland, which is genuinely the shortest route, even though on a Web Mercator map it looks like a bizarre detour compared to the "straight" planar line.

When does each apply in ArcGIS?

- The **measure tools** in Map Viewer measure geodesically — which is why you can trust them even though the map is Web Mercator.
- Many geoprocessing and analysis tools offer a **geodesic option** — buffers are the flagship example. A planar 100 km buffer around a northern city built in Web Mercator draws as a tidy circle but covers far too little real ground — at Minneapolis's latitude, only about 70 km of it. A geodesic buffer is correct everywhere; it's the one that looks slightly egg-shaped on a Web Mercator screen, and that's the projection lying, not the buffer. When a tool offers the geodesic method, and your data spans a large area or lives in Web Mercator, take it (Chapter 16, Proximity and Overlay, revisits this tool by tool).
- **Field calculations** in Pro can compute geodesic lengths and areas into attribute fields; the shape-length and shape-area fields that come free with feature classes are *planar*, in the layer's own coordinate system, so their trustworthiness depends entirely on that system.

The decision rule is simple. Small area plus appropriate local projection: planar is fine and fast. Large area, high latitude, or Web Mercator: geodesic, always. And when precision genuinely matters, an equally good alternative is to project the data into a proper local system first and then measure planar — that's the "choosing a projection for analysis" question, handled at the end of this chapter.

Datum Transformations: Moving Between Earths

Converting between two projected systems that share a datum is pure arithmetic — exact, automatic, nothing to decide. Converting between systems on *different* datums (NAD83 to WGS84, NAD27 to NAD83) requires a **geographic transformation** (also called a datum transformation): a recipe for shifting coordinates from one calibration of the Earth to another. Unlike projection math, transformations are empirical — derived from measurements — so several published transformations often exist between the same pair of datums, differing in method, vintage, and accuracy.

You'll meet transformations in two places. When ArcGIS Pro notices that layers in your map sit on a different datum than the map itself, it raises a **transformation warning** and picks a default; you can review and change the choice in the map properties, on the **Properties > Transformation** tab. And when you run the Project tool (next section) across datums, it asks you to choose a transformation as one of the tool's inputs.

How do you choose? In practice: Pro's suggested default, which is sorted by suitability for your data's extent, is right far more often than not. The transformations to prefer are the grid-based ones for your region when offered (in North America these have names containing terms like NADCON or HARN), because they interpolate from dense measurement networks and are the most accurate available. And there's a special case worth knowing: for many years the standard NAD83-to-WGS84 transformation was effectively "treat them as identical," which is fine at map scale and wrong by a meter or two at survey grade — precisely the drift discussed earlier. If sub-meter accuracy matters to your work, this is a place to slow down and pick the more rigorous, time-aware transformation rather than the legacy shortcut.

Tip: If two layers you *know* should align are showing a consistent offset of a meter or so in Pro, open **Properties > Transformation** on the map before touching the data. Nine times out of ten the fix is choosing a better transformation there, not editing anything.

The Project vs Define Projection Trap

ArcGIS Pro ships two tools with confusingly similar names, and mixing them up is the single most destructive coordinate-system mistake available to you. Learn this distinction cold.

Project (in the Data Management toolbox — find it via **Analysis > Tools** and search for "Project") converts data from one coordinate system to another. It does real math: every coordinate is recalculated, a new dataset is written, and the metadata is updated to match. Use it when your data has a *correct* coordinate system label and you want the data to actually live in a different one.

Define Projection does no math at all. It only rewrites the metadata label — the little tag that tells ArcGIS "these numbers are in coordinate system X." The coordinates themselves don't change by a hair. Its one legitimate purpose is repairing data whose label is *missing or wrong*: a shapefile that arrived without its .prj sidecar file, or data mislabeled by whoever exported it. You use Define Projection to declare what the coordinates *already are* — never to declare what you wish they were.

The trap runs both directions:

- **Using Define Projection when you meant Project:** your data is correctly labeled State Plane, you want Web Mercator, and you "define" it as Web Mercator. Now State Plane numbers are wearing a Web Mercator name tag. ArcGIS believes the tag and plots the same numbers as positions in the wrong system — and the same pair of numbers means a completely different place in each system, so your data teleports, often thousands of kilometers, frequently into the ocean near the equator. Nothing was converted; everything was mislabeled.
- **Using Project on mislabeled data:** the label was already wrong, and Project faithfully does correct math on top of the wrong premise — garbage in, precisely transformed garbage out. The output lands wrong and now has plausible-looking metadata, which makes the error harder to find later.

The decision rule fits in one sentence: **if the label is right and you want different coordinates, Project; if the coordinates are right and the label is wrong or missing, Define Projection.** If you're not sure which situation you're in, the diagnostic routine below tells you — and note that hosted feature layers in ArcGIS Online don't expose this footgun directly; it's a Pro-side hazard that arises while preparing data for publishing.

Watch out: Define Projection modifies the dataset in place — there's no output dataset, and no undo. Before using it on anything you can't easily re-download, copy the data first. It's one of the few tools in Pro where a wrong click permanently rewrites the original.

Diagnosing Misaligned Data, Step by Step

When layers don't line up, resist the urge to start "fixing." Diagnose first — the *size* of the misalignment nearly always identifies the cause. Work through this sequence:

Step 1: Measure the offset. Use a measure tool or just eyeball against the basemap. How far off is the layer, roughly?

OFFSET MAGNITUDE	MOST LIKELY CAUSE
A meter or two, consistent everywhere	NAD83 vs WGS84 datum mismatch, or a missing/legacy transformation

OFFSET MAGNITUDE	MOST LIKELY CAUSE
Tens of meters up to ~200 m, consistent	NAD27 data mixed with NAD83/WGS84 data
A few hundred meters to a few km, and the layer looks subtly the wrong <i>size</i>	Units mismatch (feet read as meters, or vice versa) — a scale problem, not a shift
Hundreds or thousands of km; wrong state, ocean, or wrong continent	Missing or wrongly defined coordinate system — a labeling problem
Features stacked at a single point in the Gulf of Guinea, off West Africa	Coordinates of 0,0 — null or failed geocodes, not a projection issue at all
Data mirrored, rotated 90 degrees, or in the wrong hemisphere	Latitude and longitude swapped, or a lost negative sign on longitude

That last pair deserves a note: the point where the equator meets the Greenwich meridian — 0,0, affectionately "Null Island" — is where features with empty coordinates wash up. If your missing data is all there, hunt for blanks in the source table, not for a projection error.

Step 2: Read the labels. In Pro, open the misbehaving layer's **Properties > Source** and find its spatial reference (Pro's name for the layer's coordinate-system information); do the same for a layer that displays correctly, and check the map's own system under the map's **Properties > Coordinate Systems**. You're looking for three things: Is the layer's coordinate system *unknown/undefined*? Does its named system plausibly match where the coordinates actually are? And do the two layers differ only in datum?

Step 3: Branch on what you found.

- **Undefined coordinate system:** you must discover what it truly is before anything else. Ask the data provider; check accompanying metadata or a readme; or detect it experimentally — look at the raw coordinate values in the attribute table or status bar. Values between -180 and 180 mean a geographic system (then the datum question remains). Six- or seven-digit values in a US context suggest State Plane or UTM. Test a hypothesis by setting the map's coordinate system to the candidate and

seeing whether the layer snaps into place against the basemap. When you've confirmed it, apply **Define Projection** with the true system — this is its legitimate use case.

- **Wrong label** (system named X, but the coordinates are clearly Y): same treatment — confirm the true system, then Define Projection to correct the label.
- **Correct labels, small offset**: it's a datum/transformation issue. In Pro, set the right geographic transformation in the map's `Properties > Transformation`; for a permanent fix, run **Project** into your working coordinate system, choosing a quality transformation when prompted.
- **Scale problem** (wrong size, roughly right place): almost always feet-versus-meters. Check whether the named system's units match what the data was actually recorded in — a State Plane feet dataset labeled with the meters variant of the same zone is the classic case. Fix the label with Define Projection using the correct-units variant.

Step 4: Fix once, at the source. After diagnosis, prefer repairing the dataset itself (Define Projection for labels, Project for conversions) over compensating inside one map. A map-level workaround leaves a landmine for the next map. And once fixed, note the true coordinate system in the item's metadata so nobody repeats the investigation (Chapter 5 covers documenting data quality; Chapter 39, the Troubleshooting Encyclopedia, cross-indexes these symptoms).

Choosing a Projection: Display vs Analysis

The choice of coordinate system is really two separate choices, and they're allowed to have different answers.

For display, the question is what the map is *for*. A local web map can simply stay in Web Mercator — the audience will never notice, and the tiled basemaps work best there. A thematic map where readers compare sizes across a country or the world should use an **equal-area** projection so the visual comparison is honest. A map centered on a place — polar regions, a single continent, one state — should use a projection centered there; ArcGIS's coordinate-system pickers can filter systems by your map's extent, which turns "choose from thousands" into "choose from a sensible handful." Changing a Pro map's display projection is free and non-destructive — it's just the map's property, and every layer follows along on the fly — so experiment. In web maps,

using a non-Mercator projection means choosing or publishing a basemap in that projection, a more involved path touched on in Chapter 6.

For analysis, the question is what the *math* needs, and the rules are firmer:

- **Match the projection's preserved property to the question.** Computing areas or densities? Equal-area projection (an Albers variant is the standard choice for the continental US). Measuring distances from a point? Equidistant, centered there, or just measure geodesically. Anything local and general-purpose? Your UTM zone or State Plane zone is nearly distortion-free for every purpose at once.
- **Do heavy analysis with all inputs in one projected system.** On-the-fly reprojection covers display, but many geoprocessing operations are more trustworthy and faster when the inputs genuinely share a suitable projected system with linear units. In Pro, rather than projecting every input by hand, you can set the **output coordinate system** environment setting — on a tool's **Environments** tab, or project-wide — and geoprocessing outputs will be produced in it (Chapter 22 covers environments in depth).
- **Never analyze in a geographic coordinate system when linear units matter.** Tools that need distances — buffers, densities, anything with "per square kilometer" in it — behave badly when the units are degrees. If a distance-based tool offers you a linear unit and your data is in raw WGS84, that's the cue to project first or use the tool's geodesic option.
- **Keep raw data in its native system; project copies for work.** The original is evidence. Projecting is cheap and repeatable; un-projecting a chain of half-remembered conversions is not.

A reasonable default posture for most people, most of the time: store and share data in WGS84 or your national datum, display on Web Mercator without guilt, and project into a local system (UTM or State Plane) or use geodesic options whenever a number — a length, an area, a density — is going to leave the map and be believed. Coordinate systems reward exactly one habit above all others: knowing, at every step, which one you're in. Everything else in this chapter is a footnote to that.

Cartographic Design: Making Maps People Can Read

A map is an argument. Every color you pick, every label you place, every layer you stack is a claim about what matters and what doesn't. Most maps fail not because the data is wrong but because the design buries the point — the reader can't tell what they're supposed to look at, the colors imply relationships that don't exist, or the classification quietly distorts the numbers. This chapter is about making deliberate design choices instead of accepting defaults, and about recognizing the handful of ways maps routinely mislead people, including you.

None of this requires artistic talent. Cartographic design is closer to technical writing than to painting: there are conventions, there are known failure modes, and there are fixes you can apply mechanically. The mechanics of *applying* these choices in ArcGIS — the Styles pane, smart mapping, symbol editors — belong to Chapter 7 (Styling and Smart Mapping) and Chapter 23 (Symbology and Labeling: The Pro Engine). This chapter is about knowing what to ask those tools for.

Visual Hierarchy: Deciding What Readers See First

Visual hierarchy is the order in which a reader's eye lands on things. On a well-designed map, that order matches the order of importance: the subject of the map first, supporting context second, reference material last. On a badly designed map, everything shouts at the same volume and the reader's eye wanders until they give up.

You control hierarchy with a small set of levers:

- **Size.** Bigger things get seen first. Your most important symbols and labels should be the largest.
- **Color intensity.** Saturated, vivid colors advance toward the reader; muted, grayish colors recede. Your subject gets the vivid treatment; everything else gets muted.
- **Value contrast.** Value means lightness — how close a color is to white or black. High contrast against the background pulls attention; low contrast pushes things back.
- **Detail.** Crisp edges and fine detail read as "foreground." Soft, simplified, or slightly transparent features read as "background."

- **Position in the stack.** In ArcGIS, layers draw in order, and whatever draws on top literally covers what's beneath. Your subject layer belongs near the top of the layer list.

The practical test is brutal and simple: lean back from the screen, squint at your map, and ask what you see first. If the answer isn't the thing the map is about, the hierarchy is wrong. If everything blurs into an even gray-green mush, there is no hierarchy at all.

A useful mental model is to assign every layer a role before you style anything. On a map of flood risk to homes, the flood zones and the affected homes are *subject* layers. Roads and city boundaries are *context* layers — useful for orientation, never allowed to compete. The basemap is *reference* — it should be nearly invisible until the reader goes looking for it. Style each layer according to its role, and the hierarchy mostly builds itself.

Figure and Ground

Figure-ground is the specific version of hierarchy that separates "the thing" from "the setting." The figure is what the map is about; the ground is everything behind it.

Human vision is wired to split scenes this way automatically — your job is to make sure the split happens where you intend.

The classic figure-ground failure is a thematic layer at full opacity in a mid-intensity color sitting on a full-color basemap. The basemap's greens, blues, and road oranges compete with the data, and the reader can't establish which plane is which. Three fixes, in increasing order of effort:

1. **Mute the ground.** Switch to a light gray or dark gray basemap (more on basemaps below). This alone rescues most beginner maps.
2. **Strengthen the figure.** Give subject features a thin, dark outline, slightly larger symbols, or a subtle drop-shadow-like halo so they detach from the background.
3. **Dim what's between.** If context layers (roads, parcels, admin boundaries) sit between subject and basemap, render them in thin, light gray lines — present but silent.

Tip: Transparency is a figure-ground tool, not just a "see the streets underneath" trick. Making a context layer 50 to 70 percent transparent pushes it into the

ground plane without removing it. But keep your subject layer at or near full opacity — a transparent subject on a busy basemap produces muddy in-between colors that belong to neither plane.

Color: The Most Abused Channel on the Map

Color does more work on a map than any other visual variable, and it's the one beginners get wrong most often — usually by asking color to say something it can't, or by letting it say something they didn't intend.

Start with the three properties of any color, because map color rules are written in these terms:

- **Hue** is what we casually call the color: red, blue, green.
- **Value** is lightness: how close the color is to white (light) or black (dark).
- **Saturation** is intensity: vivid versus washed-out or grayish.

The core principle: **hue distinguishes kinds; value orders amounts**. Different hues (red vs. blue vs. green) read as *different categories* with no ranking implied. Differences in value (light blue to dark blue) read as *less to more*. When you violate this — using a rainbow of hues to show a numeric variable, or a light-to-dark ramp to show unranked categories — the reader's visual instincts fight the data, and the instincts usually win.

The Three Kinds of Color Ramps

Nearly every colored map you'll make uses one of three ramp types. Choosing the right type is the single highest-leverage color decision, and it's determined entirely by the *shape of your data*, not by taste.

RAMP TYPE	DATA IT FITS	HOW IT LOOKS	EXAMPLE
Sequential	Numeric data running from low to high	One hue, light to dark (light = low, dark = high)	Population density, rainfall, income

RAMP TYPE	DATA IT FITS	HOW IT LOOKS	EXAMPLE
Diverging	Numeric data with a meaningful midpoint	Two hues meeting at a neutral middle	Change since last census (loss vs. gain), above/below average, election margin
Qualitative	Categories with no order	Distinct hues at similar value and saturation	Land use types, soil classes, school districts

Sequential ramps carry a convention worth respecting: on a light background, **darker means more**. Readers assume it without looking at the legend. If your ramp runs the other way, some fraction of your audience will read the map backwards and never check. (On a dark basemap the convention flips — lighter and brighter reads as more against a dark ground.)

Diverging ramps are the right choice *only* when the midpoint means something real: zero change, the national average, a regulatory threshold. The neutral middle color says "nothing notable here," and the two hues say "notable in opposite directions." If there's no meaningful midpoint, a diverging ramp invents a story — it splits your data into two teams that don't exist. Conversely, mapping a change-over-time variable with a sequential ramp hides the most important boundary in the data: the line between places that grew and places that shrank.

Qualitative ramps have one rule and one hard limit. The rule: keep all the hues at roughly equal value and saturation, so no category accidentally looks more important than the others. The limit: humans can reliably distinguish only a handful of categorical colors on a map — somewhere around six to eight before the legend becomes a matching puzzle. If you have fifteen land-use classes, the fix is not fifteen colors; it's grouping minor classes into an "other" bucket or into broader parent categories, then mapping those.

Watch out: The rainbow ramp — red through orange, yellow, green, blue, violet — is the most tempting and most misleading choice for numeric data. Hue has no natural order (is green more than orange?), the yellow band grabs attention regardless of whether those values matter, and the ramp collapses completely in

grayscale or for colorblind readers. If a rainbow default appears, change it. A single-hue sequential ramp is almost always what you actually wanted.

Colorblind Safety

Roughly one in twelve men and a much smaller share of women have some form of color vision deficiency, most commonly reduced ability to distinguish red from green. On any map with a real audience, some readers are colorblind. Design for them from the start:

- **Avoid red-green opposition.** For diverging schemes, blue-to-orange, blue-to-red, or purple-to-green pairs survive most forms of color vision deficiency; red-versus-green does not.
- **Lean on value, not just hue.** A ramp that also runs light-to-dark stays readable even if the hues collapse, because lightness perception is largely unaffected by color vision deficiency. This is one more reason sequential ramps built on value are so robust.
- **Double-encode when stakes are high.** If distinguishing two categories really matters — "compliant" vs. "violation," say — differ in shape or pattern as well as color.
- **Test.** Free simulators (Color Oracle is a well-known desktop tool; several websites do the same) show your map as a colorblind reader would see it. The ColorBrewer palettes — a research-derived set of map color schemes that ArcGIS's built-in ramps draw on heavily — flag which schemes are colorblind-safe, and ArcGIS's styling interfaces increasingly do the same.

A related habit worth building: check your map in grayscale. If the story survives with the hue stripped out, your value structure is doing its job, and the map will also survive a bad projector, a cheap printout, and most vision deficiencies in one shot.

Classification: Where Honest Data Becomes a Dishonest Map

When you map a numeric field with distinct colors — the classic *choropleth* map, meaning areas shaded by a statistical value — you must first chop the continuous range of values into a few classes. Every county with a value in class 3 gets the same color, regardless of whether it sits at the bottom or top of that class. This chopping is called

classification, and it is the most consequential decision on any statistical map, because *the same data classified two different ways can tell two contradictory stories* — with neither map technically containing an error.

ArcGIS offers several classification methods (you'll meet the controls in Chapter 7). Here is what each one actually does, and — more importantly — how each one lies.

Equal Interval

Equal interval divides the data range into classes of identical width. If values run from 0 to 100 and you want five classes, the breaks fall at 20, 40, 60, and 80.

Strengths: The legend is instantly understandable — every class spans the same amount, so readers can do arithmetic in their heads. Good for familiar, evenly spread quantities like percentages or temperature.

How it lies: Real-world data is rarely spread evenly. Most social and economic variables are skewed — many small values, a few huge ones. Equal interval on skewed data dumps nearly everything into the bottom one or two classes and leaves the top classes almost empty. The map comes out nearly monochrome, whispering "nothing varies here" about data that varies enormously.

Quantile

Quantile puts an equal *count* of features in each class. Five classes, 100 counties: each class holds 20 counties, wherever the breaks must fall to make that true.

Strengths: Every class is populated, so the map always shows full contrast and spatial pattern. Excellent for rankings — "which fifth of the counties is highest" — and it never produces the washed-out equal-interval failure.

How it lies: Quantile ignores the actual values entirely. Two nearly identical counties can land in different classes because a break happened to fall between them; two wildly different counties can share a class. Quantile *manufactures* apparent differences in flat data and *hides* real gaps in clustered data. It always shows a dramatic map — even when the honest map would be boring.

Natural Breaks (Jenks)

Natural breaks — often labeled Jenks after the cartographer who formalized it — uses an algorithm to place breaks where the data has genuine gaps, minimizing the value spread inside each class and maximizing differences between classes.

Strengths: For a single dataset viewed on its own, it usually produces the most honest-looking map: classes correspond to real clusters in the data. It's the default in much of ArcGIS for good reason.

How it lies: The breaks are bespoke to this exact dataset. Map the same variable for 2010 and 2020 with natural breaks, and the two maps use *different class breaks* — comparing them side by side is meaningless, though nothing on the maps warns you. The break values themselves are also awkward numbers (say, 3.7, 11.2, 28.9) that mean nothing to readers.

Standard Deviation

Standard deviation classes show how far each value sits from the average, in units of the data's own spread. The map becomes a picture of "above average / below average / way above / way below."

Strengths: Perfect when the question is deviation from a norm. Pairs naturally with a diverging ramp centered on the mean.

How it lies: It assumes the data is roughly bell-curve shaped. On skewed data the mean itself is misleading, and the classes inherit that distortion. It also tells readers nothing about the actual values — only about relative position.

Manual Breaks

Manual (or defined) breaks are ones you set yourself: round numbers, regulatory thresholds, historically meaningful cutoffs.

Strengths: This is the professional's workhorse. Breaks at meaningful values (0, the poverty line, a legal limit, nice round numbers) make the legend genuinely informative. And manual breaks are the *only* way to compare maps: use identical breaks across years, regions, or scenarios, and differences between the maps become real differences in the data.

How it lies: However you like. Manual breaks are the tool of choice for deliberate distortion — nudge one break past an inconvenient value and a problem district changes color. The method has no built-in honesty; it has *your* honesty.

METHOD	BEST FOR	CHARACTERISTIC LIE
Equal interval	Evenly spread data; familiar units	Washes out skewed data — "nothing to see here"
Quantile	Rankings; guaranteed contrast	Dramatizes flat data; hides real gaps
Natural breaks	One-off maps of one dataset	Uncomparable across datasets or time
Standard deviation	Distance from an average	Distorted by skew; hides actual values
Manual	Meaningful thresholds; comparison series	Whatever the author wants

Two habits protect you. First, **look at the histogram** — the little bar chart of the value distribution that ArcGIS shows in classification interfaces. Ten seconds with the histogram tells you whether the data is skewed, clustered, or smooth, and therefore which methods will misbehave. Second, **try two or three methods** and watch how the story changes. If the pattern survives every method, it's probably real. If it appears only under quantile, you were about to publish an artifact of your classification, not a fact about the world.

Tip: Also consider *unclassed* mapping — a continuous color ramp with no classes at all, where each feature's color is computed directly from its value. ArcGIS's smart mapping defaults often work this way (see Chapter 7). Unclassed maps can't lie through break placement, at the cost of making it harder to read specific values off the legend. For exploratory work, they're an excellent honesty check on your classified version.

The Cardinal Rule: Normalize Your Data

If you remember one rule from this chapter, make it this one: **never map raw counts with shaded areas**. This mistake is so common and so corrosive that it deserves its own section.

Here's the failure. You map "number of burglaries" by county, shading counties darker where the count is higher. The map you get is, almost regardless of the crime data, a population map: big and populous counties have more of *everything* — more burglaries, more bakeries, more left-handed dentists — simply because more people live there. Your reader concludes the big dark counties are dangerous. The actual burglary *rate* there may be the lowest in the state.

The fix is **normalization**: dividing the count by something that puts places on equal footing.

- Divide by **population** to get a rate: burglaries per 10,000 residents.
- Divide by **area** to get a density: people per square kilometer.
- Divide by a **total** to get a percentage: households below the poverty line as a share of all households.

Which denominator? The one that answers your actual question. "Where is a resident most at risk?" wants per-capita. "Where is development most intense?" wants per-area. "Where is the problem the largest share of the whole?" wants a percentage. Choosing the denominator *is* choosing the question — make it consciously.

ArcGIS supports this directly: choropleth styling interfaces include a normalization option (often a "divided by" field choice), or you can compute the rate into its own field, or derive it on the fly with an Arcade expression — Arcade being ArcGIS's built-in scripting language for calculations like this (Chapter 8 covers it from zero). Chapter 12 (Schema Design) discusses when a computed field beats an on-the-fly expression.

Counts aren't unmappable — they're just wrong for *shaded areas*. Areas visually assert "this whole surface has this intensity," which only makes sense for rates and densities. To show counts honestly, use **graduated symbols**: circles sized by the count, placed on each area. A big circle says "a lot happened here" without claiming anything about intensity. Sized symbols for magnitude, shaded areas for rate — that pairing covers

most statistical mapping you'll ever do, and putting *both* on one map (circles sized by count over areas shaded by rate) is a genuinely powerful combination.

Watch out: Normalization has its own trap: small denominators. A county with 90 residents and one burglary posts a rate of 111 per 10,000 — likely the "worst" rate on your map, driven by a single event. Sparse places produce wild, unstable rates. When you see extreme values in barely populated areas, suspect the denominator before you trust the pattern. Chapter 17 (Statistical and Pattern Analysis) covers more rigorous responses; the quick defenses are to filter out or gray out areas below a minimum population, or at least to flag the instability in your map notes.

Label Design: Type as a Map Symbol

Labels are symbols made of letters, and they obey the same hierarchy rules as everything else. Chapter 9 covers the mechanics of label classes, expressions, and placement engines in Map Viewer, and Chapter 23 covers Pro's engine; here is the design layer that sits above the mechanics.

Label selectively. The strongest label decision is what *not* to label. A map where every feature is labeled is a map where no label can be read. Label the features that serve the map's purpose — the ten cities that anchor orientation, not all four hundred; the rivers your story mentions, not every creek. Use scale-dependent labeling so detail appears only as the reader zooms in.

Build a type hierarchy. Readers should be able to tell what *kind* of thing a label names before they read it. The traditional conventions do this work for free, because readers have absorbed them from a lifetime of maps:

- Water features: *italic*, usually in a blue that matches the water.
- Cities and towns: upright (roman) type, size stepped by importance — capitals larger than towns.
- Regions, countries, mountain ranges: capitals with extra space between the letters, often stretched across the feature's extent.
- Roads: small labels aligned along the line.

You don't have to follow these exactly, but notice their structure: each feature class gets a consistent, distinguishable treatment, and importance maps to size. Two or three typefaces maximum — commonly one serif family and one sans-serif family — with variation coming from size, weight, italics, and color rather than from more fonts.

Keep labels legible against the map. Labels sit on top of visually noisy backgrounds, so give them help: a **halo** (a thin outline around each letter, usually in the background color) or a subtle mask lets text float over busy imagery or dense linework. Keep halos thin — one that's too thick makes text look like it's been cut out of the map with scissors. On dark basemaps, use light text with dark halos; on light basemaps, dark text with light halos. And tint labels toward their feature's color — blue-gray for water labels, warm gray for terrain — so they read as part of the map rather than stamped on it.

Respect placement conventions. Point labels sit best to the upper right of their symbol; line labels follow the line's curve; area labels stretch comfortably inside their polygon. ArcGIS's placement engines handle most of this automatically once you've told them the feature type. The design decision you own is the *priority ranking*: when two labels collide, which one survives? Rank label classes by importance so the engine drops the right ones.

Legend Design: The Contract With Your Reader

The legend is where your map's encoding gets explained, and it's usually written last, worst, and in the software's default language. A few principles turn it from an afterthought into part of the argument.

The title does half the work. A legend titled `POP2020_SUM` has failed before the first entry. A legend (or better, a map title) that says "Population change, 2010–2020" and a subtitle that says "percent change by county" tells the reader what they're looking at before they decode a single color. Write titles as answers to "what am I seeing?" — plain words, units included, source and date noted somewhere on the map or its item page.

Don't legend the obvious. Every entry in a legend costs reader attention. The basemap doesn't need entries. Context layers styled in quiet gray rarely need entries. If a symbol

is labeled directly on the map, it doesn't need a legend row. A good legend for a thematic map often has exactly one block: the classification of the subject variable.

Make class labels humane. Default class labels like "12.34567 – 28.91014" are machine output. Round them ("12 – 29"), state the units ("% of households"), and consider replacing pure numbers with words where it helps ("Below state average," "More than double the average"). If you used manual breaks at meaningful thresholds, say so — that's a feature of your map, not a footnote.

Order and orient ramps naturally. Vertical legends should run high values at top, low at bottom, matching the "more is up" instinct. Show the classes as a connected ramp rather than scattered swatches when the variable is continuous, so the legend itself communicates "this is one ordered quantity."

In web maps, remember that the legend is generated from your layer names and style settings — so renaming a layer from `census_tracts_join_final_v3` to "Median household income" fixes the legend and the layer list in one move. You'll typically find the control on the layer's properties in Map Viewer or on the layer's item page.

The Basemap Is a Design Decision

The basemap — the pre-built reference map your data sits on — feels like a given, something you pick once from `Basemap` in Map Viewer and forget. It's actually one of your loudest design choices, because it occupies more of the screen than anything else on the map.

The organizing question: **how much attention can the ground afford to take?** Ranked from quietest to loudest:

- **Light gray / dark gray canvas.** Deliberately minimal — pale ground, faint labels, no color. Built precisely so thematic data can own the color channel. The light gray canvas is the default answer for statistical maps; the dark canvas makes vivid and glowing colors pop, and suits dashboards and nighttime themes.
- **Terrain-oriented and muted topographic styles.** Some landform context without much color competition. Good when physical geography is relevant background — hiking, hydrology, wildfire.

- **Streets and navigation styles.** Full road-network detail with conventional coloring. Right when addresses and driving are the point; noisy under a choropleth.
- **Imagery (satellite/aerial photography).** Maximum realism, maximum noise. Every pixel varies, so anything you draw on it fights for contrast. Use it when the photography itself is evidence — land cover, construction progress, damage assessment — and give overlaid features strong outlines and halos. The "hybrid" variant adds reference labels on top.

Two more considerations. First, **match label density to purpose:** many basemaps come in variants with and without place labels, and ArcGIS often composes the basemap as a detailed ground layer plus a separate *reference layer* of labels drawn above your data — which is why basemap labels sometimes sit on top of your features. If those labels crowd your subject, look for a variant without the reference layer or reorder what you can. Second, **check the basemap at your working scales** — a basemap that looks calm zoomed out may be busy at neighborhood scale.

Tip: If none of the standard styles suits your design, the ArcGIS Vector Tile Style Editor lets you recolor an Esri vector basemap — dim the roads, recolor the water to match your palette, strip label layers — and save the result as your own basemap item. It's the quickest route to a map that looks designed rather than assembled. Chapter 5 covers finding basemaps and other reference layers in the Living Atlas.

Common Map Crimes and Their Fixes

Every one of these appears constantly in the wild. Learn to spot them in your own drafts.

Mapping counts in a choropleth. The classic. The map is secretly a population map. *Fix:* normalize to a rate or density, or switch to graduated symbols. (See the cardinal rule above.)

The rainbow ramp on numeric data. Implies categories where there's a continuum, and highlights the arbitrary yellow band. *Fix:* single-hue sequential ramp, or diverging if there's a true midpoint.

Red-green diverging schemes. The two ends collapse into one color under the most common form of colorblindness. *Fix:* blue-orange or purple-green pairs; verify with a simulator.

Everything at full volume. Six layers, all saturated, all opaque, on a full-color basemap. No figure, no ground, no point. *Fix:* assign layer roles (subject / context / reference), mute everything but the subject, switch to a canvas basemap.

Default class breaks presented as truth. The map's story is an artifact of whatever method the software picked. *Fix:* look at the histogram, try multiple methods, prefer meaningful manual breaks — and use identical breaks for any maps meant to be compared.

Labeling everything. Four hundred overlapping labels, none readable. *Fix:* label selectively, set scale ranges, rank label priorities.

The raw-field legend. SUM_POP_20 and eight-decimal class bounds. *Fix:* rename layers, round breaks, add units, title the legend with the question it answers.

Imagery basemap under a subtle thematic layer. Photography eats the color contrast; the data disappears. *Fix:* gray canvas basemap, or strong outlines and halos if imagery is genuinely needed.

3D, glow, and effects for their own sake. Extruded polygons, drop shadows, and animated effects that add drama but no information. *Fix:* ask what question the effect answers. If the answer is "it looks cool," remove it.

Ignoring projection at the design stage. Area comparisons on a Web Mercator map — the standard web map projection, which dramatically inflates areas far from the equator — visually exaggerate high-latitude regions. This is a design problem as much as a technical one: a choropleth's ink is proportional to area, so distorted areas mean distorted emphasis. *Fix:* for continental- and world-scale area comparisons, use an equal-area projection where your workflow allows it, or at minimum say so in the map's notes. Chapter 3 (Coordinate Systems and Projections, Complete) explains the machinery.

Watch out: The most dangerous maps are the ones that lie by accident and look professional doing it. A clean layout, a tasteful ramp, and a confident legend can dress up a count-not-rate choropleth with quantile breaks into something that will be screenshotted into a slide deck and believed. Design polish raises the credibility of whatever the map says — which is exactly why the honesty checks in this chapter come *before* the polish.

A Working Checklist

Before you publish or share any thematic map, run through this list. It takes two minutes and catches most of the failures above.

1. Squint test: does the subject of the map read first?
2. Is the ground quiet — canvas basemap or muted context layers?
3. Does the ramp type match the data shape (sequential / diverging / qualitative)?
4. Does darker (or brighter, on dark ground) mean more?
5. Is the map readable in grayscale and under colorblind simulation?
6. If it's a choropleth: is the variable a rate, density, or percentage — not a count?
7. Did you look at the histogram before accepting class breaks? Do breaks land on meaningful values?
8. If maps will be compared, do they share identical breaks and ramp?
9. Are labels selective, ranked, and legible against their background?
10. Does the legend state the variable, units, date, and source in plain words?

Design choices compound. A muted basemap makes your colors work harder; honest classification makes your legend trustworthy; a selective labeling scheme makes your hierarchy legible. Get the habits right on small maps and they'll hold when the maps — and the audiences — get bigger. The next chapter turns to the other half of map quality: finding data worth mapping in the first place.

Finding Data: Living Atlas, Open Portals, and Evaluating Quality

Most maps are built mostly from other people's data. That is not a shortcut or a compromise — it is how the field works. Somebody at the Census Bureau maintains the boundary files. Somebody at your state's transportation department maintains the road centerlines. Somebody at Esri curates a world imagery **basemap** (the background reference layer that sits under everything else in a map) assembled from dozens of providers. Your job, most of the time, is not to create data from scratch but to find the right existing dataset, judge whether it is good enough for your purpose, and bring it into your work without breaking anything downstream.

That last sentence contains three distinct skills, and this chapter covers each in turn: knowing where to look, knowing how to judge what you find, and knowing how to import it safely. The first skill is the easiest and the one people over-invest in. The second is where projects quietly go wrong — a map built on a boundary file that is eight years out of date looks exactly as polished as one built on current data, right up until someone who knows the area spots the error. The third determines whether your map still works six months from now.

One framing to carry through the whole chapter: **there is no such thing as good data, only data that is fit for a particular use.** A parcel layer that is perfect for a county tax assessor may be uselessly detailed for a statewide overview map, and a generalized world-countries layer that is perfect for a global dashboard would be laughably wrong for a boundary dispute. Every evaluation question in this chapter is really the same question in different clothes: *fit for what?*

The data landscape at a glance

Before diving into each source, here is the territory. Nearly everything you will ever add to a map comes from one of five places:

SOURCE	WHAT IT IS	BEST FOR	WATCH FOR
Living Atlas	Esri's curated collection of ready-	Basemaps, demographics,	Curation varies; some entries are

SOURCE	WHAT IT IS	BEST FOR	WATCH FOR
	to-use layers	imagery, environment, boundaries	community- contributed
Government open portals	Federal, state, and local agencies publishing their own data	Authoritative boundaries, infrastructure, environment, statistics	Formats and freshness vary wildly by agency
ArcGIS Hub sites	Organizations' public data catalogs built on ArcGIS	Local government data already in ArcGIS-friendly form	The site's polish says nothing about the data's quality
OpenStreetMap	A global, volunteer- built map database	Roads, buildings, points of interest, especially outside the US	Completeness varies block by block; share-alike license
Commercial providers	Companies selling data products	Demographics, traffic, business listings, premium imagery	Cost, licensing restrictions, redistribution limits

You will use all five eventually. Most projects start with the first two.

The Living Atlas

The **ArcGIS Living Atlas of the World** is Esri's curated collection of geographic content: layers, maps, apps, and imagery that Esri and its partners maintain and make available to every ArcGIS user. Think of it as a library where the librarians have already done a first pass of quality control. It is the single best starting point for almost any data hunt, because everything in it is already in a form ArcGIS understands — no downloading, converting, or uploading required. You search, you click add, and the layer appears in your map, streaming live from wherever it is hosted.

You can reach the Living Atlas two ways. The first is directly, at its own website (search the web for "ArcGIS Living Atlas"), where you can browse and read about content before you ever open a map. The second, and the one you will use most, is from inside Map Viewer: in the **Layers** pane choose **Add layer > Browse layers**, then switch the source list from your own content to **Living Atlas**. Chapter 6 (Map Viewer: The Complete Reference) covers the mechanics of the add-layer workflow; here we care about what you are choosing from.

What is actually in it

The Living Atlas organizes its content into broad categories, and knowing them saves you from aimless keyword searching. The exact category names shift occasionally, but the durable themes are:

- **Basemaps and imagery.** The reference layers that sit under everything else: streets, topography, satellite and aerial imagery, and hillshade (shaded-relief) terrain. Chapter 15 (Imagery and Rasters in Practice) goes deep on the imagery side.
- **Boundaries and places.** Administrative boundaries at every level — countries, states, counties, cities, postal areas, census units. These are the workhorse layers of thematic mapping.
- **Demographics and lifestyle.** Population, income, age, housing, and consumer data, much of it derived from national censuses and Esri's own demographic modeling. Note that some of these layers are **subscriber or premium content**: both require signing in with an organizational account, and premium layers additionally consume credits (the metered currency of ArcGIS Online usage — see Chapter 34 for how credits work) when you use them.
- **Environment.** Land cover, soils, watersheds, ecological classifications, protected areas.
- **Earth observations and live feeds.** Weather, stream gauges, wildfire perimeters, earthquake feeds — layers that update themselves on a schedule, some of them nearly in real time.
- **Transportation, infrastructure, and society.** Roads, rail, utilities, schools, hospitals, and similar human-built features.

A useful mental split: some Living Atlas content is **Esri-authored and maintained**, some comes from **authoritative partners** (government agencies and NGOs that publish through the Atlas), and some is **community-contributed** and merely nominated into the collection. All of it passed a curation bar, but the bar is not uniform. The item page — which we will dissect later in this chapter — tells you which kind you are looking at.

Tip: In the Living Atlas browse interface, look for filter options that narrow results to authoritative content and to your region of interest. Filtering by region matters more than people expect: a search for "parcels" unfiltered returns content from all over the world, most of it irrelevant to you.

How to search it well

Naive keyword search in any GIS catalog is frustrating, because the words you would naturally type are rarely the words in the item title. Three habits fix most of this:

Search for the noun, not the question. You want to know where flooding risk is high; the layer is called something like "FEMA flood hazard areas." Search catalogs the way a data publisher names things: the agency, the program, the official noun. "Flood hazard" beats "where does it flood." "Census tracts" beats "neighborhood statistics."

Use the filters, always. Every ArcGIS search interface lets you filter by item type (you usually want feature layers — layers made of vector features you can style and query — or imagery layers), by region, by categories, and often by an authoritative flag. An unfiltered search on a common word returns a haystack. Two filters in, you are usually looking at a short, plausible list.

Read past the first result. Catalogs rank by a blend of relevance and popularity, and popularity is a lagging, herd-driven signal. The third result maintained by a federal agency and updated recently is usually a better bet than the first result someone published years ago and abandoned, no matter how many views it has accumulated.

Watch out: Popularity metrics on an item page — view counts, favorites — measure attention, not quality. An outdated layer with a great title can rack up

views for years precisely because it ranks well, which keeps it ranking well. Never let view count substitute for reading the item details.

Layers versus maps versus apps

A Living Atlas search returns more than layers. You will see web maps (a saved arrangement of layers plus styling), apps, and styles. For building your own work you almost always want the **layer** — the raw ingredient — not someone else's finished map. That said, opening a publisher's own web map is a legitimate research move: it shows you how the data's owner thinks it should be styled and filtered, which often reveals quirks (a status field you should filter on, a category worth excluding) that the metadata never mentions.

Government open data portals

Governments are the world's largest producers of geographic data, and over the past two decades most of them have adopted an open-data posture: publish it, let people use it, require attribution at most. The catch is that "government data" is not one place. It is thousands of places, with patterns that repeat at each level.

Federal patterns

In the United States, a handful of federal sources cover an enormous share of common needs:

- **The Census Bureau** publishes the boundary and statistical backbone of American mapping: states, counties, tracts, block groups, and the TIGER/Line files (the Census's geographic database of boundaries, roads, and addresses), plus the demographic estimates that attach to them. If your map involves "how many people" or "which official area," the Census is upstream of it somewhere.
- **USGS** (the U.S. Geological Survey) covers terrain, hydrography, geology, and topographic mapping, and operates The National Map as its distribution hub.
- **NOAA** covers weather, climate, oceans, and coasts.
- **FEMA** publishes flood hazard data; the **EPA** publishes environmental facilities and monitoring data; the **USDA** publishes soils and agricultural data; the **Bureau of Transportation Statistics** and **FAA** cover transport.

- **Data.gov** is the federal umbrella catalog that indexes across agencies. It is a decent discovery tool and a poor download experience — treat it as a card catalog that points you to the agency's own site.

Other countries follow the same shape: a national statistics agency for boundaries and demographics, a national mapping agency for terrain and topography (Ordnance Survey in Britain, IGN in France, and so on), specialized agencies for weather and environment, and a national open-data umbrella portal indexing all of it. The names change; the pattern does not.

A pleasant surprise: much federal data is *also* in the Living Atlas, republished as ready-to-use layers, often by the agencies themselves. Check the Atlas first; go to the agency's portal when you need the raw files, the full attribute set, or a vintage the Atlas does not carry.

State and local patterns

Nearly every U.S. state runs a **GIS clearinghouse** — a portal aggregating statewide layers: parcels, roads, address points, elevation, imagery, administrative boundaries. Naming varies (a "geoportal," a "GIS data hub," a state "geographic information office"), but a web search for your state's name plus "GIS data portal" almost always lands on it. States are typically the best source for data that federal sources hold only in generalized form: detailed parcels, school districts, precise municipal boundaries.

Counties and cities are where the freshest, most detailed data lives — and where consistency ends. A large metro county may run a superb open-data site with nightly-updated parcels; the neighboring rural county may offer a phone number for the assessor's office. Local data is worth the hunt when your map is local: the county's parcel layer, the city's zoning layer, and the public works department's utility data have no substitutes at any other level of government.

The seams between levels are where errors hide. City, county, and state versions of "the same" boundary frequently disagree in the details, because each maintains its own copy on its own schedule. When boundaries matter legally or financially, find out which agency is the **authoritative source** — the office legally responsible for maintaining that data — and use theirs.

Tip: When two datasets claim to represent the same thing, prefer the one published by the agency that *creates* the data over any agency that merely *aggregates* it. Aggregators add convenience and subtract freshness.

ArcGIS Hub sites

Many of those government portals — plus universities, utilities, and nonprofits — run their public data catalogs on **ArcGIS Hub**, Esri's product for building open-data websites on top of an organization's ArcGIS content. You can recognize a Hub site by its look and by its behavior: a searchable catalog where every dataset page offers the data as a live layer, as downloads in common formats (spreadsheet, shapefile — Esri's long-standing file format for vector data — and GeoJSON, a plain-text format for geographic data), and through an API (a programmatic interface, covered in Chapter 32).

Hub sites are the best of both worlds for an ArcGIS user. Because the underlying data is already hosted in ArcGIS, you often do not need to download anything: the dataset page exposes the live feature layer, and you can add it straight to your map by URL or by finding the same item in ArcGIS Online search. You get government-authoritative data with Living-Atlas-level convenience.

The one mental adjustment: a Hub site is a *skin* over an organization's content, and organizations vary in discipline. The platform guarantees a consistent interface; it guarantees nothing about whether the layer behind the page was updated last week or during a previous administration. The evaluation habits later in this chapter apply with full force.

OpenStreetMap

OpenStreetMap (OSM) is a collaborative, volunteer-built map database of the entire world — roads, buildings, shops, trails, land use, and much more — released under an open license. It is the Wikipedia of maps, with the same profile of strengths and weaknesses: astonishing coverage and freshness where the volunteer community is active, gaps and staleness where it is not, and quality that varies from block to block rather than from dataset to dataset.

For ArcGIS users, OSM shows up three ways. First, as **basemaps**: Esri offers OSM-based basemaps alongside its own. Second, as **extracted layers**: Esri and others publish ready-to-use layers of OSM buildings, highways, points of interest, and similar themes, many available through the Living Atlas — searchable by the "OpenStreetMap" keyword. Third, as **raw data** you extract yourself through third-party download services, useful when you need a custom slice.

Where OSM shines: pedestrian detail (paths, crossings, building footprints), points of interest, fast-changing urban areas, and above all coverage in countries where no open government alternative exists. Where it demands caution: attribute consistency (a volunteer in one town tags things differently than a volunteer in the next) and completeness you cannot assume — the absence of a feature in OSM means only that nobody has mapped it yet.

Watch out: OSM's license (the Open Database License) is a **share-alike** license: if you build a derivative *database* from OSM data and distribute it, you may be required to share that database under the same open terms. Displaying OSM on a map with attribution is routine; blending OSM data into a proprietary dataset you sell is a decision that needs actual license reading, and possibly legal advice, first.

Commercial data

Sometimes nothing free fits. Commercial providers sell data that open sources cannot match: current-year demographic estimates and forecasts, consumer spending patterns, live and historical traffic, business listings with attributes, premium high-resolution imagery, precise address databases. Esri itself is a commercial data provider — its demographic and lifestyle datasets, its geocoding service (address lookup), and its routing services are all products that consume credits or require licenses, and many surface through the Living Atlas marked as premium content.

Two things distinguish commercial data workflows. First, **cost structure**: you may pay per use (credits), by subscription, or by one-time purchase, and the pricing usually depends on geography and use case. Second, and more important, **licensing**: commercial licenses typically restrict redistribution. You might be allowed to make a public map that *displays* the data but forbidden to let users *download* it; allowed

internal analysis but not publication. The license, not the technology, defines what you can build.

The buying decision comes down to the same fitness questions as everything else, plus one: *is the delta over the free alternative worth the money?* Often the honest answer is that the free Census-derived layer is a year or two staler and completely adequate. Sometimes — a site-selection analysis hinging on current spending patterns, a logistics app needing live traffic — the delta is the whole point.

Reading an item page critically

Everything in ArcGIS Online — every layer, map, and app — has an **item page**: the page you land on when you click a search result's title, showing its description, details, and settings. The item page is where you stop being a browser and start being an evaluator. Most people skim the thumbnail and the first sentence. You are going to read the whole thing, and here is what to read for.

The description and summary. Beyond the obvious (what is this?), you are reading for scope and method. Does it state what geography is covered, what time period, and how the data was produced? A description that answers those three is the mark of a serious publisher. A description that is one vague sentence — or that is clearly the default text — tells you the publisher's care level, which usually correlates with the data's.

The owner. Who published this? An account named for a government agency or a known organization is a different proposition from an individual username. Click through to the owner's profile when unsure; an agency account typically has a portfolio of related, maintained content. Also look for an **authoritative badge** — ArcGIS lets organizations mark their official content as authoritative, and search interfaces can filter on it. The badge is a genuine signal, though not a guarantee: it marks the publisher's status, not the dataset's fitness for your use.

The dates — all of them. Item pages carry several dates and it is critical not to confuse them. The *created* date is when the item was added to the catalog. The *updated* date is when the item or its data last changed — but beware, edits to the description alone can refresh it. Neither is the same as the **currency of the data itself**: a layer uploaded yesterday can contain a survey from a decade ago. The data's actual vintage should be stated in the description or metadata; if it is not, that is a finding in itself.

Update frequency. Is this a living layer on a maintenance schedule, or a snapshot published once? Serious publishers say so explicitly ("updated nightly from the county assessor's database," "static extract, published for the 2020 census"). A live-feed layer and a frozen snapshot are both legitimate — but they are fit for different uses, and mistaking one for the other is a classic failure.

Terms of use. This section carries the license: what you may do with the data, what attribution is required, whether commercial use or redistribution is allowed. An empty terms-of-use section does not mean "no restrictions" — it means the publisher didn't state them, which leaves you guessing. For anything beyond casual use, prefer items whose terms are explicit.

The layer's actual contents. The item page lists the layers inside the item and, crucially, lets you open the data itself — look for a way to view the attribute table (the spreadsheet-like view of every feature's values) or the fields list (often under a **Data** or **Fields** tab on the item page). Five minutes in the attribute table teaches you more than any description: Are the field names comprehensible? Are values populated or full of blanks? Is there a date field telling you per-feature currency? Chapter 9 (Pop-ups, Fields, and Labels) covers working with fields in depth; here you are just auditing.

Formal metadata. Some items carry structured metadata — a standardized document recording lineage (where the data came from and how it was processed), accuracy statements, and contact information. When present (look for a metadata link or section on the item page), it is the gold standard of documentation. Its absence is common and not disqualifying, but its presence moves an item sharply up your trust ranking.

Tip: Before committing to a layer, add it to a scratch map and zoom to a place you know personally — your neighborhood, your hometown. Ground truth against your own knowledge is the fastest quality check that exists. If the layer is wrong where you can check it, assume it is wrong where you cannot.

Evaluating fitness: the four questions

You have found a candidate layer and read its item page. Now the decision: is it fit for *your* use? Four questions cover nearly everything.

Currency: is it recent enough?

Not "is it recent" — *recent enough for the decision this map informs*. Geological data from decades ago is fine; a road layer from decades ago is not; for an evacuation dashboard, even a week-old wildfire perimeter is dangerously stale. Work out how fast the real-world phenomenon changes, then demand data at least that fresh. And distinguish, again, publication date from content vintage: the honest question is "when was this true?", not "when was this uploaded?"

Completeness: is everything there that should be?

Completeness failures are the most insidious because absence is invisible. A map of reported potholes shows where reports happen, which is partly a map of where people report things. A business-listings layer might cover chains thoroughly and independents barely. Ask: what is the collection method, and what would that method systematically miss? Check the geographic edges too — datasets aggregated from multiple sources often have seams where one contributor's diligence ends. And check attribute completeness separately from feature completeness: all the buildings may be present while the "year built" field is blank for half of them.

Scale and resolution: is it detailed enough — or too detailed?

Every vector dataset was digitized at some **scale** — the level of detail at which its shapes are faithful. A coastline generalized for a world map turns to visible straight-line nonsense when you zoom to a harbor; conversely, survey-grade parcel boundaries are wasteful overkill on a national overview and can bog performance down. For rasters, the equivalent is **resolution** — the ground size of each pixel (Chapter 15 covers this in depth). Metadata sometimes states the source scale; when it does not, the zoom test from the tip above reveals it quickly. The rule: the data's native detail should match or exceed the finest zoom at which your audience will scrutinize it — and not exceed it by orders of magnitude.

Licensing: are you allowed to do what you are planning?

Run your *actual plan* against the terms: display in a public map? Analysis whose results you publish? Extraction and merging into your own dataset? Commercial use? Each may have a different answer under the same license. Watch specifically for attribution requirements (nearly universal, and cheap to honor — put the credit in the layer's attribution and your map's description), non-commercial clauses, redistribution bans,

and share-alike provisions like OSM's. When a use matters and the terms are silent, ask the publisher; agencies answer these emails more often than you would expect.

Watch out: "Public" and "open" are not synonyms. Data you can see on a government website is not automatically data you may republish. Genuine open data carries an explicit open license; anything without one deserves a moment's thought before it ends up in something you distribute.

Beyond the four, two cross-cutting habits. **Prefer sources with a maintenance story** — an update schedule, a responsible agency, a changelog — over one-time heroic uploads, because your map will outlive the upload. And **triangulate anything load-bearing**: if a single layer's correctness determines your map's central claim, find a second source and compare them before you trust either.

Bringing found data into your work safely

You have chosen a layer. There are two fundamentally different ways to use it, and the choice matters more than beginners realize.

Reference it, or copy it?

Referencing means adding the publisher's live layer to your map. Your map holds a pointer; the data stays on the publisher's servers. You get their updates automatically, you spend no storage, and you stay effortlessly compliant with most licenses. The cost is dependency: if the publisher deletes the layer, restructures its fields, or changes its sharing settings, your map breaks or silently changes — and you find out when a user does.

Copying means exporting the data and republishing it as your own hosted layer (the full publishing workflow is Chapter 10's territory; the many import paths are Chapter 11's). You get independence and stability, the ability to edit and restyle the schema, and immunity to upstream changes. The costs: you now own storage and maintenance, your copy begins going stale the moment you make it, and you must confirm the license actually permits redistribution — copying is exactly the act that restrictive terms restrict.

A practical rule of thumb:

SITUATION	PREFER
Live feeds and frequently updated data	Reference — the updates are the point
Casual or exploratory maps	Reference — cheap and easy
Data you need to edit, subset, or re-schema	Copy — you cannot edit someone else's layer
Long-lived public maps where stability is critical	Copy (license permitting) — control your dependencies
Anything with redistribution restrictions	Reference — the license decides for you

For maps that matter, many practitioners do both: reference the live layer day to day, but keep a periodic exported snapshot in reserve so an upstream disappearance is an inconvenience, not a crisis.

The import checklist

Whichever path you choose, a short discipline pays off every time:

1. **Filter before you carry.** If you need one county from a national layer, apply a filter (or export just your slice, license permitting) rather than dragging the whole country through every draw and query. Performance implications live in Chapter 10.
2. **Mind the coordinate system.** Web maps reproject on the fly for display, so mixed layers *look* fine — but analysis across mismatched coordinate systems is where subtle errors breed. Chapter 3 (Coordinate Systems and Projections) is the full story; the habit here is simply to check what you have imported.
3. **Preserve the paper trail.** Record where the data came from, when you got it, and its stated vintage — in your copied item's description at minimum. Six months from now, "where did this layer come from?" will be asked, and the person asking may be you.
4. **Keep attribution intact.** Fill in the attribution/credits on any copy you publish, and honor required credit text exactly as the license states it.

5. Re-verify before big moments. Before a public launch or a decision meeting, re-open the source item page. Confirm the layer still exists, still updates, and still says what you think it says. Upstream data is a living thing; treat it like one.

Data sourcing is not a phase you complete; it is a relationship you maintain. The habits in this chapter — search like a publisher, read item pages like a skeptic, evaluate against your actual use, and import with a paper trail — turn the world's enormous, uneven supply of geographic data from a hazard into the greatest force multiplier in mapping. From here, Chapter 6 covers the Map Viewer where these layers come together, and when you are ready to publish data of your own, Chapters 10 and 11 pick up the thread.