Introduction

This session is all about turning non-spatial data into spatial data. There are three general methods to achieve this goal.

- *Geocoding:* Turning a name of some geographical feature into coordinates, a line or a polygon.
- *Georeferencing:* Taking an image of some place and giving it a known reference system.
- *Digitizing:* Taking a georeferenced image and extracting the vector features in some way.

Geocoding is (at least partially) driven by algorithms, so that it can be done nicely within *R*. The latter two require a lot of human input, for which we will turn to *QGIS*.

Geocoding

Geocoding transforms some written description of a location such as an address (physical or internet IP) or a place name to a location on the earth's surface.

A *geocoder* is a software, a web service, or a person that implements a geocoding process, that is some standardized operations, algorithms, and data sources that produce the desired spatial representation.

Reverse geocoding uses geographic coordinates to find a description of the location, most typically a postal address or place name.

Street address interpolation

Address geocoding often assumes that the addresses are evenly spaced along a linear network.

Here it would code 150 Easy St. to be about halfway down the block irrespective of the actual building footprints.

Providers and iteration

Geocoding is usually an iterative process, you will often find some places immediately and others will be misspelled or hard to find. Different geocoders have a different fault tolerance. You will want to use more than one.

Common geocoding services are

- Google Maps API
- OpenStreetMap Nominatim API
- Geonames API
- Bing API and more ...

There are many *R* functions and packages that interface with these providers. It is not super hard to write your own specialized interface.

API keys and query limits

Most providers will limit the searches you can do per day (or for free), often the limit is around 2500 queries.

- Google stopped unregistered API access this year, so we will not use it anymore (but could after obtaining a free API key).
- OpenStreetMaps still allows free access if your request rate is lower than one request per second.
- Geonames just needs a free username, which you can make in five minutes.

Beware of these limits if you need to geocode a lot of data. You might want to build waiting times into your code.

An example using tmap's geocode_OSM

```
require(tmaptools)
geocode_OSM("Moshi, Tanzania", as.sf = T,
            details = F) %>% glimpse()

## Observations: 1
## Variables: 8
## $ query    <chr> "Moshi, Tanzania"
## $ lat      <dbl> -3.348646
## $ lon      <dbl> 37.34352
## $ lat_min  <dbl> -3.508646
## $ lat_max  <dbl> -3.188646
## $ lon_min  <dbl> 37.18352
## $ lon_max  <dbl> 37.50352
## $ geometry <POINT [°]> POINT (37.34352 -3.348646)
```

Installing *QGIS*

Georeferencing and digitizing can only be done in a *click-and-point* GIS environment. QGIS is a great open source GIS.

You can install it from:

`www.qgis.org/en/site/forusers/download`

Windows users:

- For now, I would recommend the 64-bit standalone installer for the long-term support version (2.18).
- If you end up using QGIS more, then you should install the OSGeo4W suite, which contains many additional features and algorithms (including GRASS, Saga, lwgeom).

Mac and Unix users should follow the instructions on the site.

Georeferencing

Hard copy maps have a lot of valuable data on them. If you have a high-quality scan of these data, then you already have a raster image in some local coordinate system created by the scanner.

Georeferencing is the process of transforming the CRS of a raster data set into a new coordinate reference system that relates the input data to points on the earth's surface.

The coordinate system of the raster to be georeferenced is called the *source CRS* and the coordinate system of the output is the *destination CRS*.

The transformation from one to the other may involve shifting, rotating, skewing, or scaling of the input raster until it fits to the destination CRS.

Ground control points

Georeferencing is done by identifying ground control points (GCPs), i.e. locations on the input raster where the *destination* CRS and the associated coordinates are known.

GCPs are found in two ways:

- Using another data set that is in the destination coordinate system. GCPs are common points in both data sets.
- Using datums or other locations with printed coordinates or geocoded coordinates. The source locations are selected and the target coordinates are entered.

GCPs should be well distributed across the image. Create GCPs near the four corners of the image, plus several located in the middle of the image.

The GDAL georeferencer

Georeferencer GDAL plugin is a core QGIS plugin which you only need to enable. It is the only tool we will use to georeference scanned maps.

1. Go to `Plugins | Manage and install plugins ...` and then type “Georeferencer” in the search bar.
2. Check the little box next to the plugin’s name and close the plugin settings window.
3. The plugin is now enabled. You will find it under `Raster | Georeferencer | Georeferencer ...`
4. You can open an image using the checkerboard button. You can pan and zoom with the mouse. The GCP table is at the bottom of the image (and empty in the beginning).

The work flow

For referencing a grid or against another reference layer:

1. Load the source image into the Georeferencer GDAL plugin.
2. If you are georeferencing the raster against another reference layer, load the second layer into the main *QGIS* application.
3. Begin to enter GPCs with the `Add point` tool (seventh button from the left, or `Edit | Add point`). You need to click on the GCP point within the image window. Zoom in **a lot** so that you can precisely click on the GCP location.
4. Once you click on the input raster with the `Add point` tool, the `Enter map coordinates` dialog will pop up. Enter the coordinates you observe in the grid or see at the GCP in the reference layer. Pan and zoom if you cannot see them.
5. Repeat until you have enough GCPs, always save your GCPs!

Google or ESRI maps as a reference layer

In most cases, you will want a modern up-to-date map as a reference layer. Why not simply use Google maps or similar services?

The QuickMapServices plugin brings these maps into *QGIS*.

1. Select `Plugins | Manage and install plugins ...` and then search for "Quick", select `QuickMapServices` and press `Install plugin`. Make sure the check box is ticked.
2. Navigate to `Web | QuickMapServices` and select `Settings`. Then select `More services` and then `Get contributed pack`. Now you are all set.
3. Go to `Web | QuickMapServices | Google` and select a map type to try it.
4. Pan and zoom in the main *QGIS* window to see how it is loading the tiles from the web. It needs fast internet.

Historical CRSs

If you find an old map, it must have an older datum. Typical older ellipsoids are Bessel 1841, Clarke 1880, Clarke 1866, International 1924, Arc 1950, Arc 1960, or WGS72. Careful, the map may also be using a local transformation.

If you are lucky, then this information is printed somewhere on the map. If not, you will have to find it or simply go straight to WGS84 as the *target CRS*.

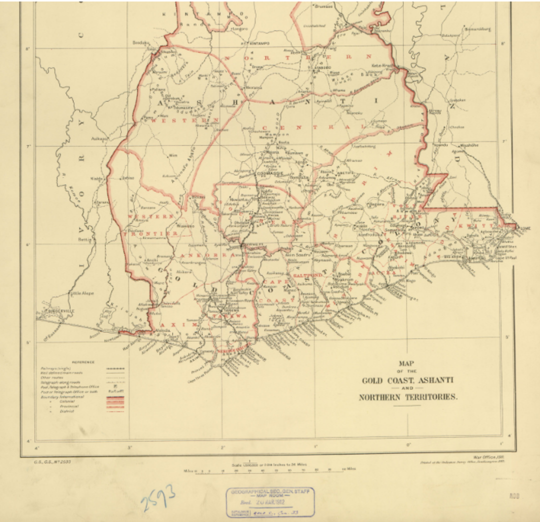
You can search for historical datums here:
`http://earth-info.nga.mil/GandG/coordsys/onlinedatum`

If you obtained a military or topographic map, it is likely to be in a UTM projection. Find the right reference UTM zone and the ellipsoid (they vary too) or use the degree graticules, if there are any.

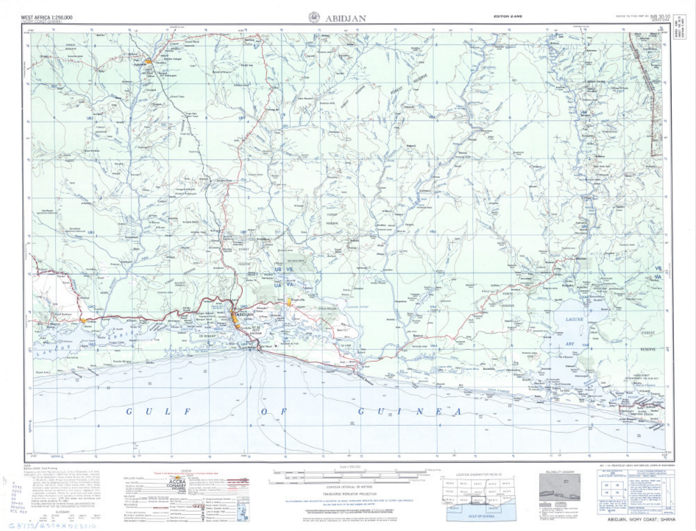
Transformation settings

1. Once all the desired GCPs have been created, click on the `Transformation settings` button.
2. Here, you can choose appropriate values for the transformation type, resampling method fields, and other output settings. For scanned maps you should use any of these:
   - Polynomial 2 or polynomial 3 should be used if the input raster needs to be bent or curved.
   - Thin plate spline allows for local deformations in the data.
3. There are five choices for the resampling method. Use nearest neighbor for scanned maps.
4. Specify the *target CRS* and check `load in QGIS when done`.
5. Verify the fit in places which are not GPCs on your scanned map to places on Google maps or another reference layer.

Example Map 1: Degrees



Example Map 2: UTM grid



Digitization work flow

1. Open a new vector layer with `Layer | Create layer | Shapefile Layer ...` and then select point, line or polygon. You can already specify the input CRS and create fields for the attribute table. Save the layer after clicking OK.
2. Go to `Settings | Snapping options` and then select `Advanced` in `Layer selection`. Change the mode to `vertex and segment` for all open vector layers, enter a tolerance of 20 and select pixels as units. Check `Enable snap on intersection`. Now your points will snap to the nearest polygon or line.
3. Start drawing new features by using the pen tool or navigating to `Layer | Toggle Editing` and then selecting `Edit | Add Feature`. Draw the feature, right-click when done and enter the attribute information.

Installing the topology checker

It is super easy to make mistakes while digitizing data. As you know, invalid geometries will give you a headache later on. Tackle the problem at its source instead.

High quality maps are always checked for topological errors using a topology checker. To install it:

1. Navigate to `Plugins` | `Manage and Install Plugins ...` and type "Topology" in the search bar.
2. Select the `Topology Checker` plugin and click on `Install plugin`.
3. Once enabled, the `Topology Checker` plugin can be found by navigating to `Vector | Topology Checker`.
4. When the `Topology Checker` window opens, it appears as a panel in *QGIS*.

Using the topology checker

The topology checker allows you to define rules for each vector layer. Click on the little tool item to open the settings dialog.

Common rules for a polygon layer are

1. “must not have invalid geometries”,
2. “must not have invalid multi-part geometries”,
3. “must not have gaps”,
4. “must not have duplicates”, and
5. “must not overlap”.

You should click the little check button after setting these rules for each desired layer to apply them.

I recommend to verify one layer at a time. You may add other layers, if you want no gaps against, say, a coastline or a country border.

Exercise

Download the prepared package for Ghana in 1911 (gold_coast.zip) from the course web site. Unpack the contents into a dedicated folder.

1. Open the Gold_Coast_1911_BNL_modified.tif which has already been georeferenced, add the ghana_neighbors.shp, and the ghana_ocean.shp. Make both polygon layers a little transparent, so that you can still see features on the map.
2. Create a new polygon layer, create a second field called name_1, and save the layer as ghana_adm1_1911.shp in the same directory as above. Set the appropriate snapping options for all relevant layers. Check that the project EPSG is 4326.
3. Starting from the country boundary or coastline, digitize three adjacent provinces by drawing the polygons. Enter the names of each and stop editing in between to save your work.
4. Use the topology checker to verify your work.