

Using GoogleMaps and GoogleEarth to Create and Use Map and Satellite Base Layers for Geospatial Field Data

Rick Reeves
FoxGrove Solutions
Santa Barbra, California

Abstract

Feature-rich open source scientific computing and geo-visualization software, hosted on powerful laptop and tablet computers equipped with high-bandwidth wireless connectivity, create an ideal platform for in-the-field data collection, visualization, analysis, and management. Ecologists and other scientists who collect geospatial data can further augment these platforms with digital environmental data and geospatial analysis and mapping software for a truly comprehensive data management solution.

This paper describes and then briefly demonstrates the software components for a portable field data management system which uses open-source geospatial and statistical software to perform an analysis. **GoogleMaps** and **GoogleEarth** are employed to present the incoming spatial data layers and outgoing results via the Internet.

The paper describes the software components and geo-spatial data sources, providing researchers new to this topic a comprehensive overview of the software available.

The complete analysis discussed in this paper is available (along with the most recent version of this paper) online:

<http://www.foxgrovesolutions.com/jsmPaper>

Toward a Portable Data Collection/ Management Platform

Scientists studying phenomena in the natural world (e.g., ecologists, geologists, soil scientists) collect, analyze, display, and archive geospatial data – information with associated geospatial attributes (longitude/latitude/altitude) occurring at points, along lines, and within polygons on the Earth surface.

Since the digital computing revolution of the 1980s, scientists have used Geographic Information System (GIS) software to manage, analyze, and display geospatial data in the form of vector maps and raster images. The high cost and steep learning curve of GIS software, combined with digital data storage and bandwidth transmission limitations, limited GIS use to well-funded research institutions and well-equipped field laboratories.

However, since the early years of this decade, advances in three digital technology aspects ease this constraint:

- 1) Increases in desktop computer processing and display power and storage capacity;
- 2) Extensive worldwide deployment of wireless broadband networks
- 3) Development of low-or-no cost open-source scientific computing software (including GIS applications) distributed and supported collaboratively over the Internet by groups of scientists and engineers.

Now, an initial investment of \$1500 - \$2000 purchases a desktop OR laptop workstation capable of significant scientific computing and scientific data storage (including GIS data). This workstation can then be configured with the Linux operating system and a suite of high-quality, feature-rich scientific computing software (see Table 1). New software package users immediately join the worldwide

collaborative ‘working group’ associated with each package; access to training material, case studies, and workshops created by the user community accelerate the software learning curve for each package.

Table 1 presents the most widely-used open-source scientific computing applications:

	RDB	Data Conve	Statisti Analys	GIS Ana	GIS Di	Web Di	Data Manage
MySql	X						
R			X	X	X		
OGR/GDAL		X		X			
Quantum GIS				X	X		
Morpho							X
Google Maps				X	X	X	

Table 1: Open Source Geospatial Computing Software Packages

GoogleMap GoogleEart Google Doc	Web Server-based <i>Interactive</i> Map and Data Display
Morpho	Heterogeneous Scientific Data Management/Documentation
R, QGIS	Map Construction and Display
R, QGIS	Analytical Computing: Statistics and Geoprocessing
GDAL/OG	(Geospatial) Data Format Conversion
MySQL/Spa Extension	Base Layer: Spatially-enabled RDBMS

A Sample Open Source Geospatial Computing Stack

The Open-Source Geospatial Computing Stack includes these layers:

The **Base Layer** consists of Relational Database Management (RDBMS) software enhanced to accommodate raster (e.g., satellite images or digital elevation maps) and geospatial data.

The Data Format Conversion layer converts raster (using GDAL) and vector (using OGR) data files from one standard geospatial format to another. Within the context of this paper, this layer is used to convert vector ESRI Shape Files and raster GeoTIFF image files into Google-compatible KML files.

The **Analytical Computing** layer performs geospatial and statistical calculations on Base Data Layer elements; in many cases, new Base Data Layers are generated and stored for subsequent display. In this system, **the R Programming Environment** and **Quantum GIS (QGIS) Geographic Information System** package.

Analysts use the **Map Construction and Display** layer to create and display digital maps on the client workstation, using layers of Base Layer datasets and sequences of Analytical Computing layer commands.

The **Data Management** layer enables scientists to document scientific data sets with descriptive metadata, which are stored along with the scientific data in the relational data base. The **Morpho Data Management** package, available at no cost from the Knowledge Network for Biocomplexity, is a good example of such a package:

<http://knb.ecoinformatics.org/morphoportal.jsp>

The Next Tier: Distributed interactive map display and analysis

Prior to the advent of Web-based computing, these map products were transmitted as static image files using standard formats (e.g., PDF and .JPEG). End users of the maps could not modify or interact with the maps; for example, change the map scale or viewing perspective, or ADD data layers to the map. These operations required a GIS software-enabled computer workstation. And while the cost of GIS software has dropped (to zero, in many cases) even faster than that of computer platforms, the steepness of the GIS software learning curve discourages many researchers from using them.

GoogleMaps / GoogleEarth: Web-Based Geospatial Data Display

Google Maps and Google Earth are Web-based map display tools allow users to view, modify and create *new* maps on their local desktop workstations using only a web browser, without installing GIS software on their local workstation.

Google Maps, an interactive web-based viewer, displays two-dimensional vector-based maps consisting of the widely-known, Google-supplied base layer stored in the ‘Google Cloud’. The scale and level of surface feature detail of the map display is controlled by the map user through a simple user interface (the Google ‘map scale control’).

GoogleEarth, another interactive geospatial data viewer, is installed on the client (user’s) computer. Users can display their own vector or raster-based data as overlays on top of Google-supplied base layers (located in the Google Cloud) created from raster satellite imagery. Unlike Google Maps, Google Earth uses a 3-dimensional viewing model that enables users to interactively change their viewing perspective - elevation and compass bearing - above the map ‘surface’. This allows them to ‘see’ topographical features as they would from an aircraft; in addition, users can create animated, simulated terrain ‘fly-bys’ between two points contained within their data set region. This feature of Google Earth’s creates a myriad of possibilities for interacting with their own geospatial data sets.

Crucial to Google Maps/Earth capabilities is the Google-designed Keyhole Markup Language (KML) an XML-based language for managing the display of 3D geospatial data. KML files contain geospatial metadata (and for vector datasets, the actual data) for user-supplied geospatial datasets that integrate the datasets with the Google-supplied base layer; Multiple KML-enhanced user-supplied data sets can simultaneously be displayed on top of the Google-supplied base layer.

Case Study: Terrain/Climate Impacts on Species Counts

To demonstrate the Geospatial Computing Stack’s capabilities, consider a sample terrain and climate effects study using Monarch Butterfly species counts collected within the state of Ohio during the year 2006 Monarch Butterfly census. The census is part of a long-term experiment performed by research scientists for a long-term study on the Monarch Butterfly migration patterns.

This study wcorrelates species counts obtained at 33 species counting stations with measurements of four physical environment parameters collected at each station locations. These parameter is listed below along with their data sources:

Surface Temperature	<i>NOAA’s Global Summary of the Day</i> http://gcmd.gsfc.nasa.gov/
Surface Precipitation	US / NWS Advanced Hydrologic Prediction Service http://water.weather.gov/precip/download.php
Terrain Elevation	<i>USGS ASTER Global DEM</i> http://gdex.cr.usgs.gov/gdex
NDVI (<i>Normalized Difference Vegetation Index: Vegetation Density measure</i>)	<i>MODIS MOD13A1 16 day Level 3 NDVI</i> http://modis.umiacs.umd.edu/

The surface temperature and precipitation values for each counting station are interpolated from two different networks of meteorological recording stations, using a script written within the R Scientific Programming Environment. The terrain elevation and NDVI values were extracted from digital terrain (ASTER GDEM) and NDVI (MODIS Terra) raster datasets using the vector/raster overlay features of the GRASS GIS geospatial analysis package. The four parameters were stored as point location attributes in an ESRI point shape file.

Figure 1 shows the United States temperature estimation stations, Ohio temperature stations (Red), and Monarch Butterfly count station locations (Yellow) for Ohio, displayed using the Quantum GIS (QGIS) geospatial analysis software.

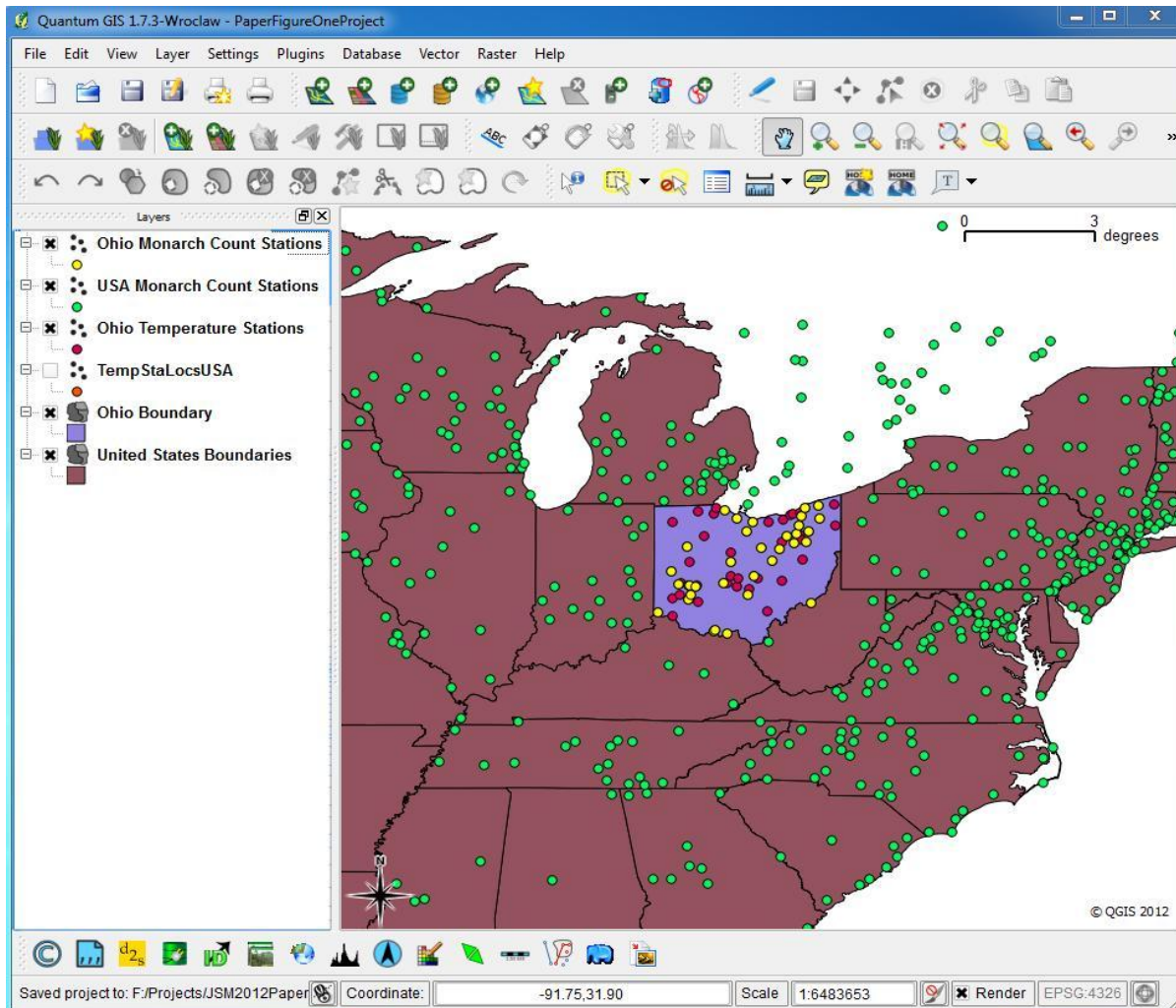


Figure 1: Surface Temperature Estimate / Monarch Butterfly Count Stations

Figures 2 and 3 display the Monarch Butterfly count stations over on the terrain elevation and NDVI raster images, using the GRASS GIS geospatial software.

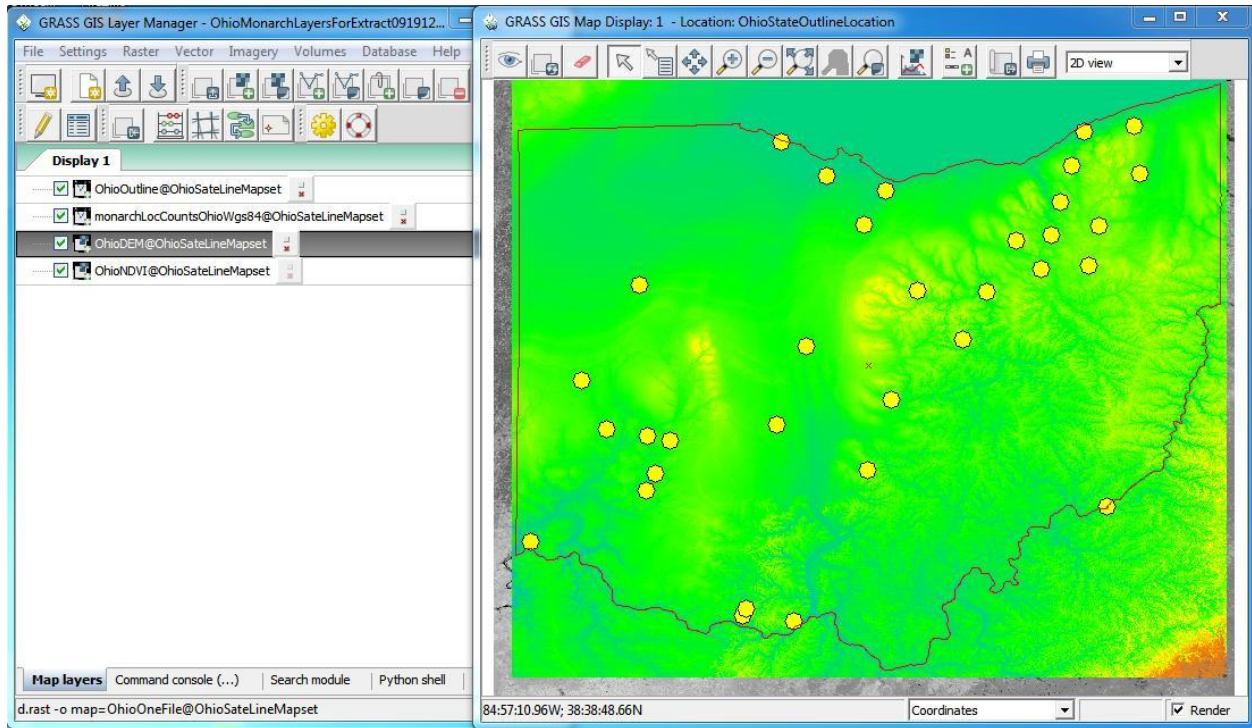


Figure 2: GRASS GIS: Monarch Butterfly Count Stations over DEM Elevation

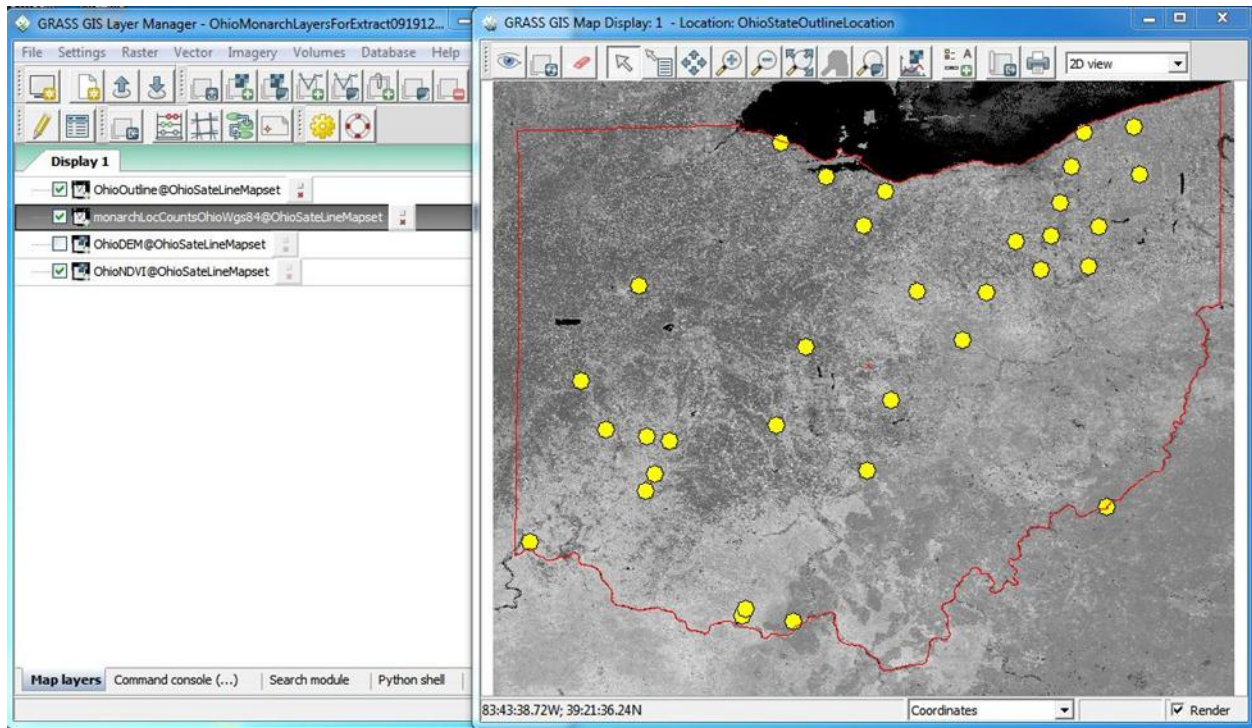


Figure 3: GRASS GIS: Monarch Butterfly Count Stations over NDVI Image

The *GoogleMaps/GoogleEarth* platform can be used to share scientific geospatial datasets via the Internet using a three-step process:

- 1) Convert the files to a standard geospatial format (ESRI Shape File format for *vector* (point/line/area) files, and PortableNetwork Graphics (PNG) format for *raster* (digital image) files.
- 2) Use a geospatial format translation utility such as *ogr2ogr* or the *GDAL library* (available as a ‘plugin’ within the R Programming Environment) to generate a corresponding *Keyhole Markup Language* (KML) file.
- 3) Start the *GoogleMaps* or *GoogleEarth*, application, specifying the KML files.

Figure 4 displays the Monarch Butterfly count stations within Ohio, using Google Maps. To create this map, a Keyhole Markup Language (KML) file was created for the ESRI Shape File using an R script (listed in the Appendix).

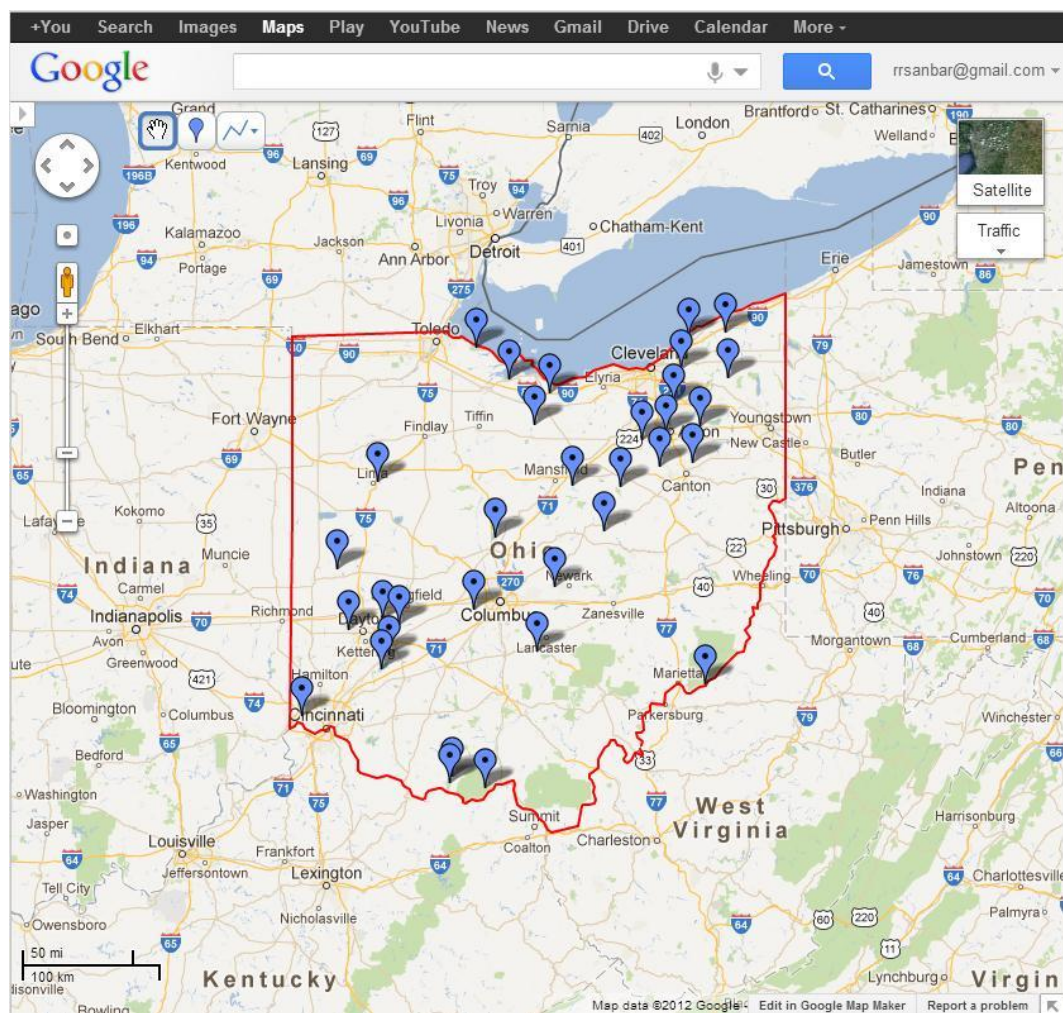


Figure 4: Monarch Butterfly Count Stations on Google Maps background

GoogleEarth displays combinations of vector and raster data layers over the default composite Earth surface satellite image. Simply create a KML file for each layer to be displayed, and then select them (using the File/Open

command) using the Google Earth toolbar. The data files can be located either on a web server, or on the local workstation.

Figures 5 and 6 display the Monarch Butterfly count stations within Ohio over the NDVI and DEM backgrounds, using GoogleEarth.

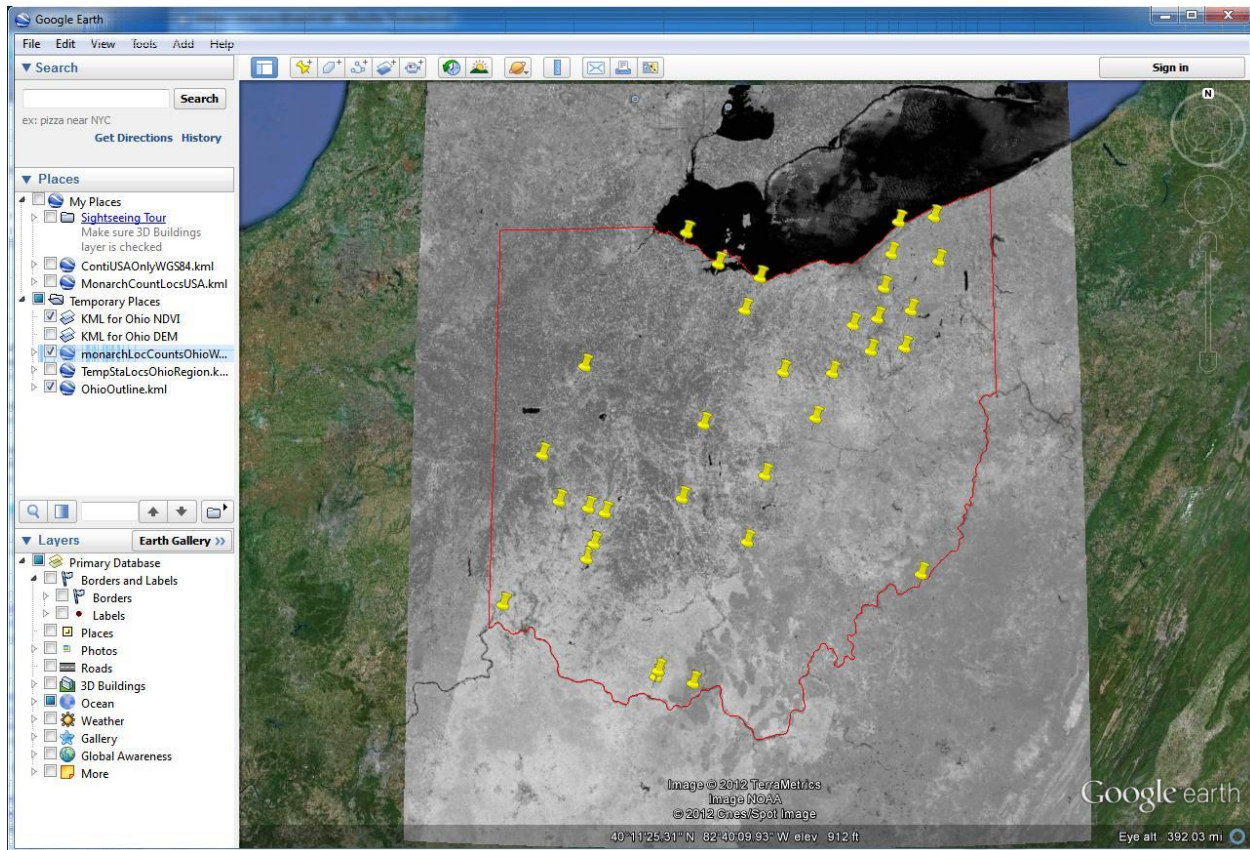


Figure 5: Monarch Butterfly Count Stations over NDVI Image: GoogleEarth

Tabular-format results of geospatial / statistical analyses can also be displayed within the Google environment, using the GoogleDocs application. In this case, the results file are exported from their native (e.g., Microsoft Excel) format into a 'shared' GoogleDocs Spreadsheet file, which is then stored 'in the Google Cloud' using a GoogleDocs account. From there, the data are available to the scientific community. **Figure 7** depicts a sample spreadsheet; the web site associated with this paper presents such a spreadsheet associated with the analysis performed for this study (which was not completed in time for publication).

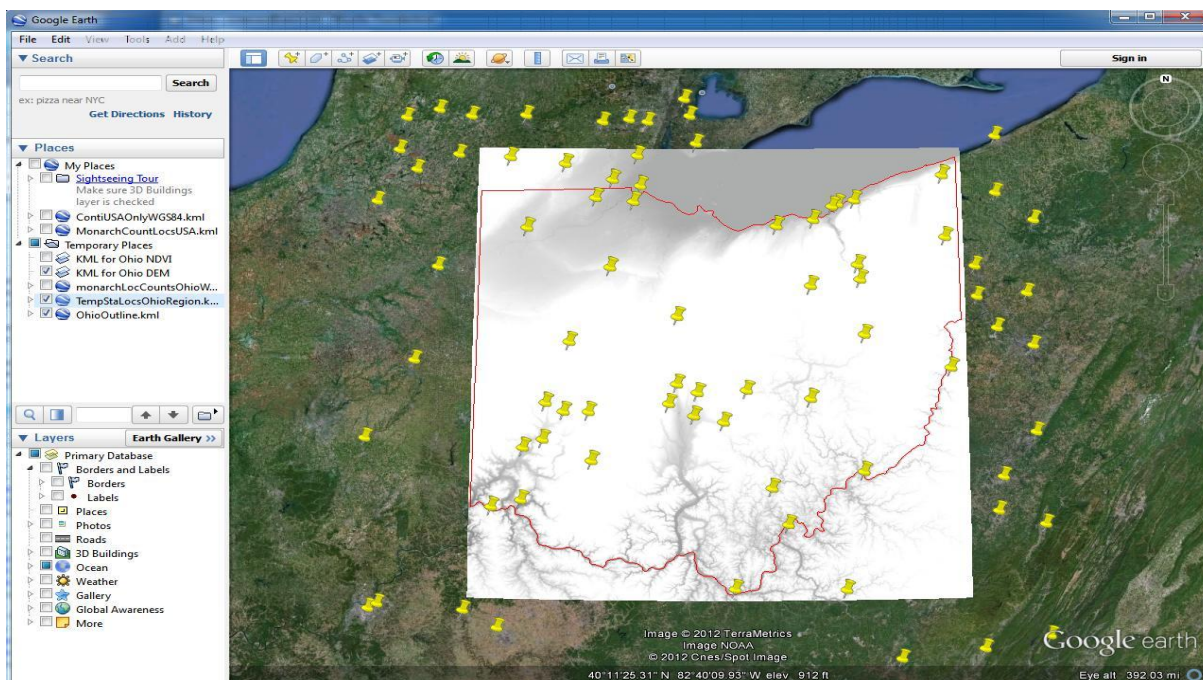


Figure 6: Monarch Butterfly Count Stations over DEM Image: GoogleEarth

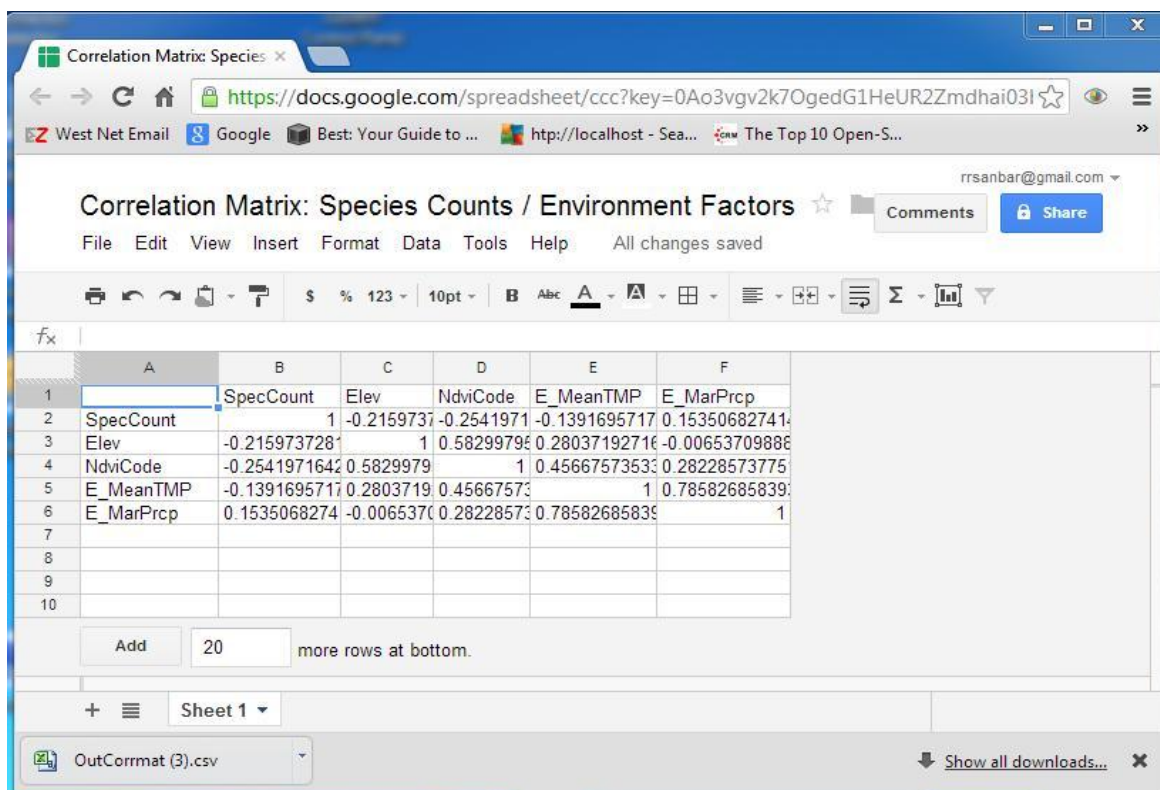


Figure 7 : Hypothetical correlation matrix presented via Internet using GoogleDocs spreadsheet.

The Next Step: Configuring a Field Data Collection/Management Platform

The data collection, extraction, analysis, and presentation stages are performed entirely using the open source geospatial software stack (configuration) described earlier in this paper. This configuration is compatible with all widely-used desktop and laptop computer operating systems. Therefore, it is straightforward to select a portable/ruggedized computer system that is designed for satellite/cellular broadband telephone reception in harsh physical environments, install the software ‘stack’, and begin using the workstation in the field research site.

Conclusion

The open-source geospatial software ‘stack’ described here provides research-grade spatial computing power at extremely low cost, placing this power within reach of any scientific research project with access to desktop computing platforms. This software can be integrated with modern low-cost computing hardware to create capable and inexpensive field data collection and management platforms.

GoogleMaps and *GoogleEarth* extend these tools onto inexpensive and ubiquitous web-enabled hand-held computing devices – netbook computers, tablets, and even smartphones. These new computing platforms create exciting and almost unlimited possibilities for field-based data collection and analysis programs.

The correlation analysis summarized in this paper, along with a complete specification for an open-source software-based field workstation, are presented at the site:

<http://www.foxgrovesolutions.com/jsmPaper>

Appendix: R Script Creates KML Files for Vector and Raster Geospatial Data Objects

```

# This program creates the KML files for JSM paper: vector
# Author: Rick Reeves, FoxGrove Solutions, September 2012

# Create KML for ndvi image

setwd("F:/Projects/JSM2012Paper/OhioLandcoverData/ndvi");
library(rgdal) # loads 'dependant' sp package
library(maptools)
ndviImage2 <- GDAL.open("Ohio.WGS84.2006097.ndvi.tif")

toPng <- copyDataset(ndviImage2,"ohioNdviImage2.png",driver="PNG")
saveDataset(toPng,"ohioNdviImage2.png")
ndviPng <- readGDAL("ohioNdviImage2.png")
ndviPngKML <- GE_SpatialGrid(ndviPng)
kmlOverlay(ndviPngKML,"ohioImageOverlay.kml","ohioNdviImage2.png",name="Ohio
NDVI KML object")

# Create KML for DEM image

setwd("F:/Projects/JSM2012Paper/OhioLandcoverData/OhioOneFile");
demImage2 <- GDAL.open("OhioOneFile.tif")
toPng <- copyDataset(demImage2,driver="PNG")
saveDataset(toPng,"ohioDemImg.png")
demPng <- readGDAL("ohioDemImg.png")
demPngGKML <- GE_SpatialGrid(demPng)
kmlOverlay(demPngGKML,"ohioDemImageOverlay.kml","ohioDemImg.png",name="Ohio
NDVI DEM object")

# now, KML files for three shape files:
# 1) Ohio Outline
# 2) Weather Stations
# 3) Monarch Sighting Stations

setwd("F:/Projects/JSM2012Paper/RicksDataSet");
monarchStaLocs <- readOGR("monarchLocCountsOhio.shp","monarchLocCountsOhio")
writeOGR(monarchStaLocs, "monarchLocCountsOhio.kml",
"monarchLocCountsOhio",driver="KML")
#
tempStaLocs <- readOGR("TempStaLocsOhioRegion.shp","TempStaLocsOhioRegion")
writeOGR(tempStaLocs,"TempStasOhioRegion.kml", "TempStasOhioRegion",driver="KML")
#
setwd("F:/Projects/JSM2012Paper/OhioLandcoverData");
ohioBoundary <- readOGR("OhioOutline.shp","OhioOutline")
writeOGR(ohioBoundary, "OhioOutline.kml", "OhioOutline", driver="KML") #end script

```