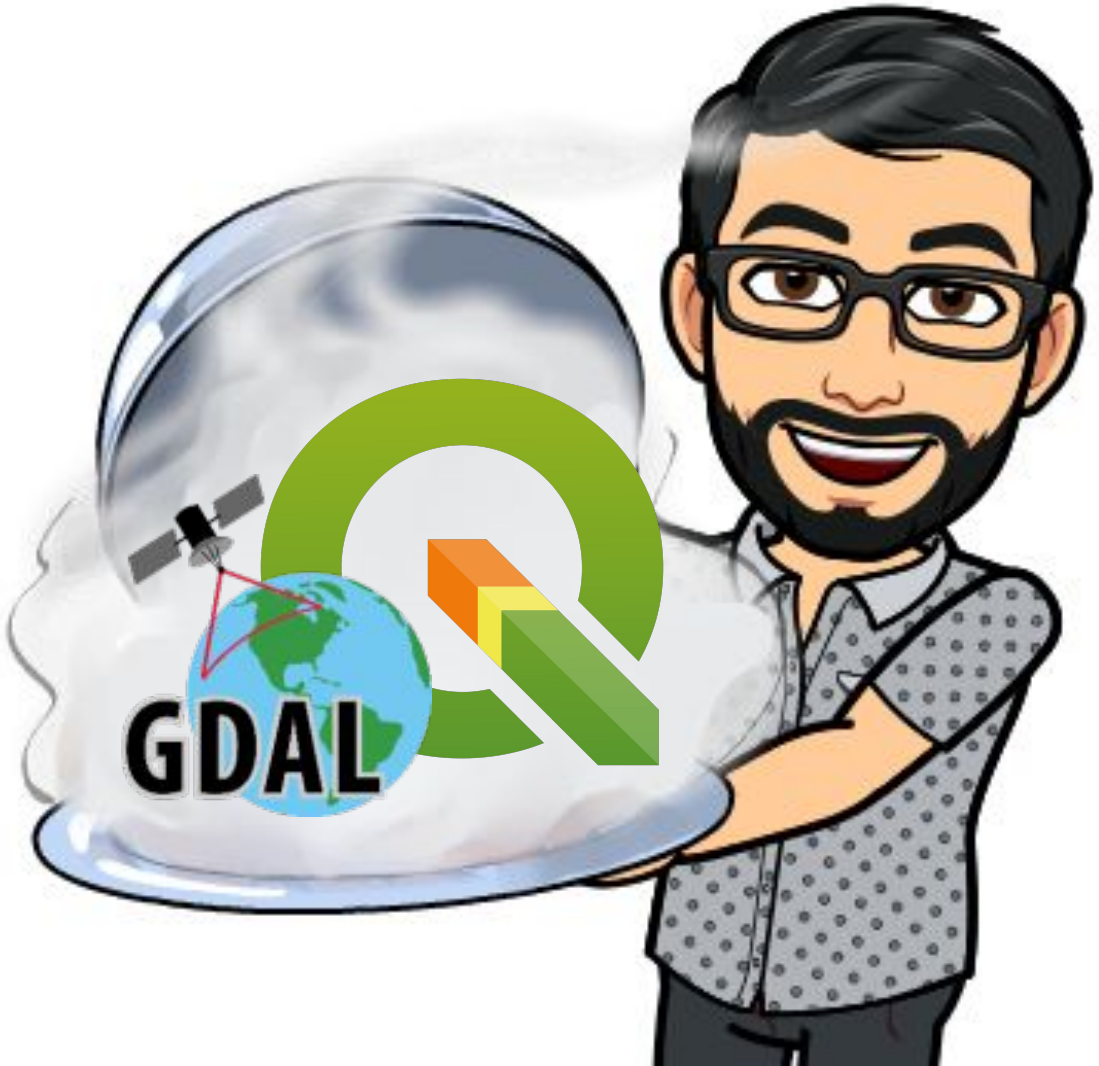
Introduction to Geospatial Computing for Pythonistas

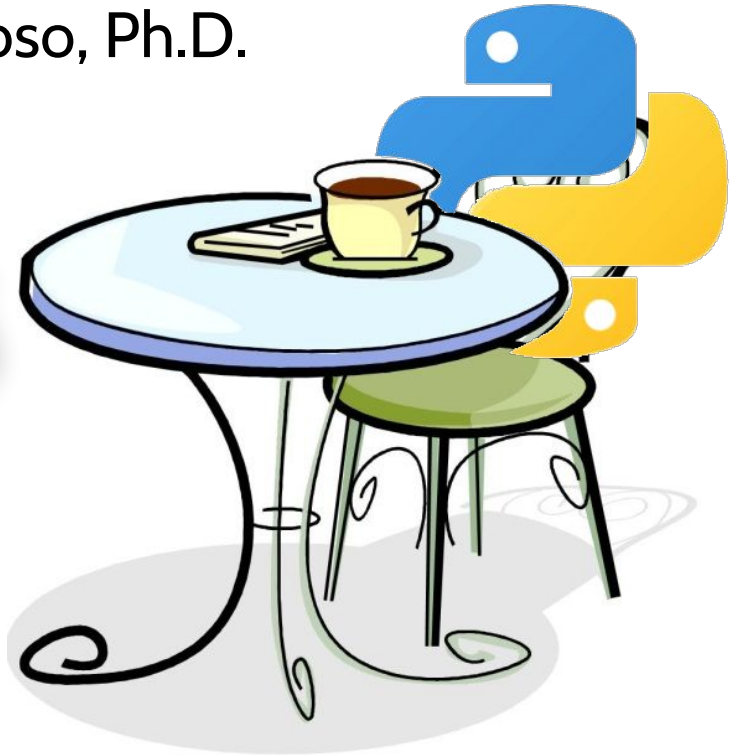


Paulo Raposo, Ph.D.

Sept 20th 2018

KnoxData

GEOGRAPHY

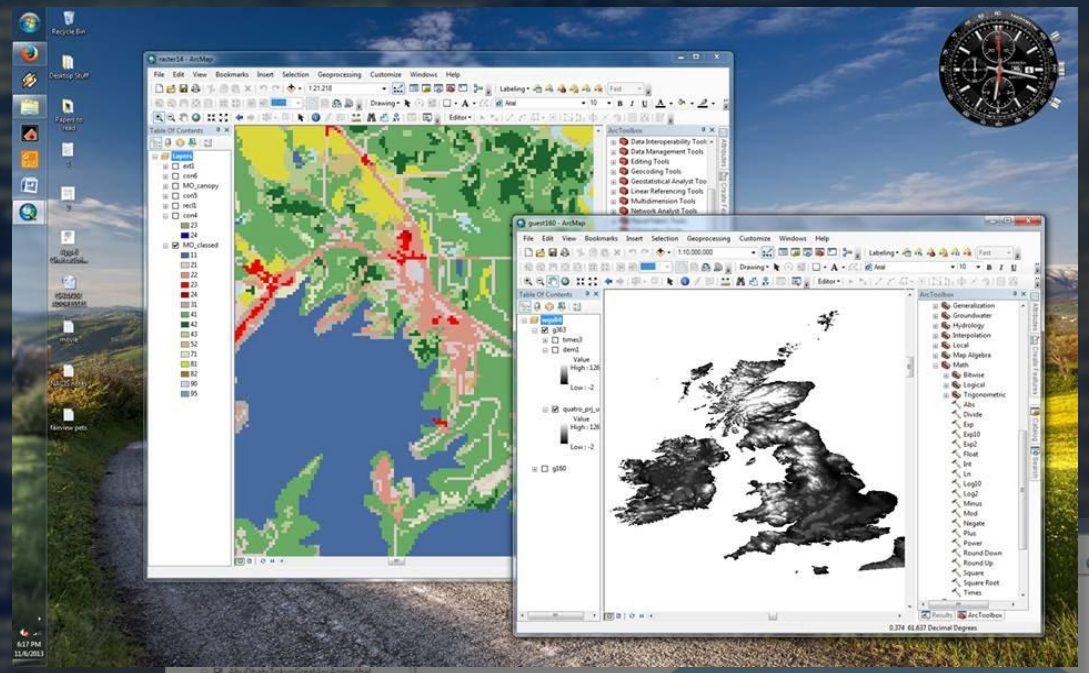


For the demo later,
you'll need these installed:

Anaconda (a nice Python distro)
QGIS (excellent free GIS)
git (versioning system)

Also, go to
http://www.naturalearthdata.com/downloads/10m-physical-vectors/
and download the Coastline data.


GIS

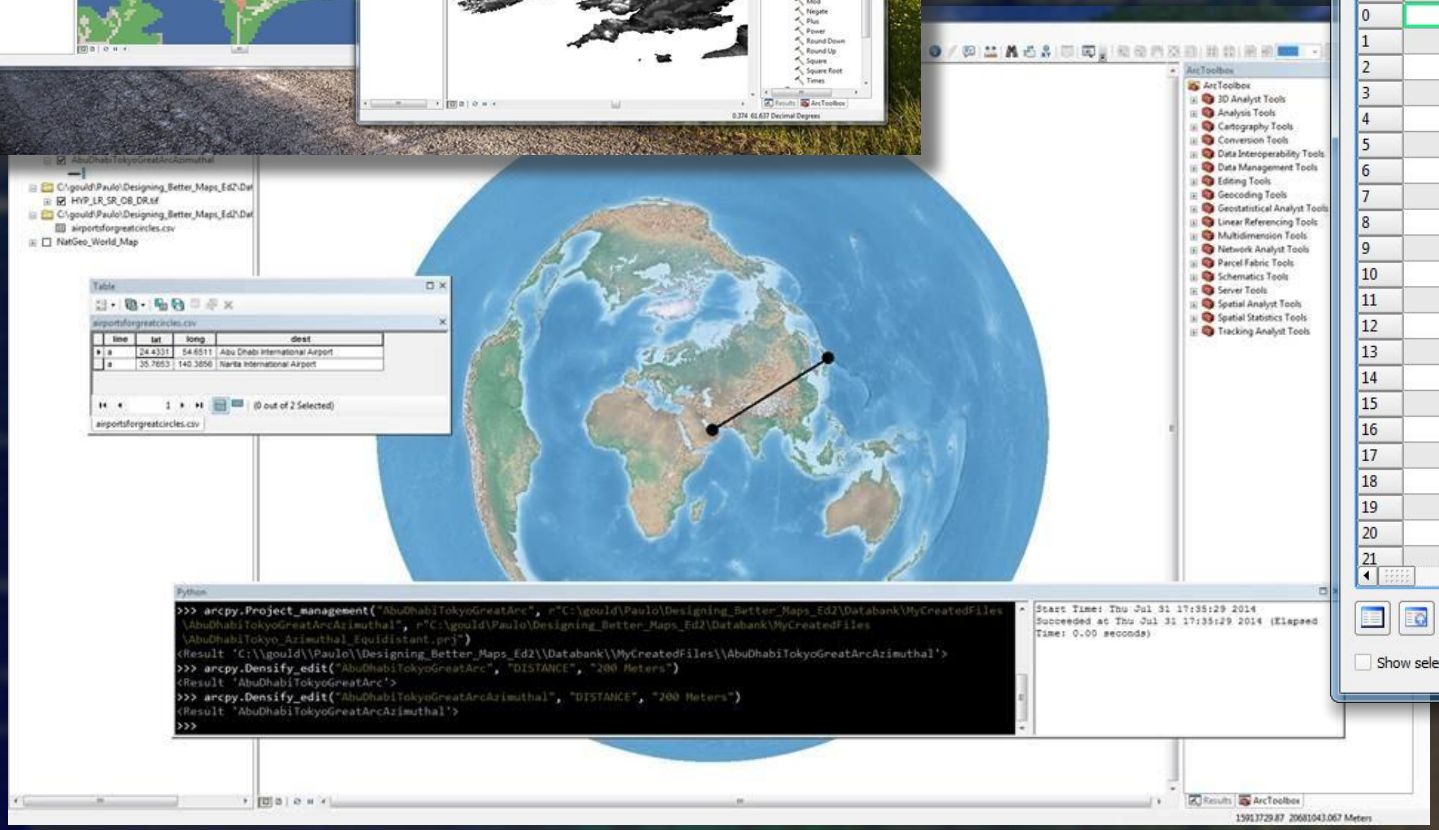


Attribute table - 10m_populated_places (7314 Feature(s))

	SCALERANK	NATSCALE	LABELRANK	FEATURECLA	NAME	NAMEPA
0	7	20	8	Populated place	Carmelo	NULL
1	6	30	8	Admin-1 capital	San Jose de Mayo	NULL
2	8	10	8	Admin-1 capital	Artigas	NULL
3	8	10	8	Populated place	Baltasar Brum	NULL
4	7	20	8	Populated place	Bella Union	NULL
5	7	20	8	Admin-1 capital	Mercedes	NULL
6	7	20	8	Admin-1 capital	Melo	NULL
7	8	10	8	Populated place	Tranqueras	NULL
8	7	20	8	Admin-1 capital	Rivera	NULL
9	8	10	8	Admin-1 capital	Tacuarembo	NULL
10	8	10	8	Populated place	Paso de los Toros	NULL
11	8	10	8	Populated place	Vergara	NULL
12	8	10	8	Admin-1 capital	Treinta y Tres	NULL
13	8	10	8	Populated place	Santa Lucia	NULL
14	8	10	8	Populated place	Jose Batlle y Or...	NULL
15	8	10	8	Admin-1 capital	Minas	NULL
16	8	10	8	Admin-1 capital	Maldonado	NULL
17	8	10	8	Admin-1 capital	Punta del Este	NULL
18	8	10	8	Populated place	Aigua	NULL
19	8	10	8	Populated place	La Paloma	NULL
20	7	20	8	Populated place	Lascano	NULL
21	7	20	8	Populated place	Castillos	NULL

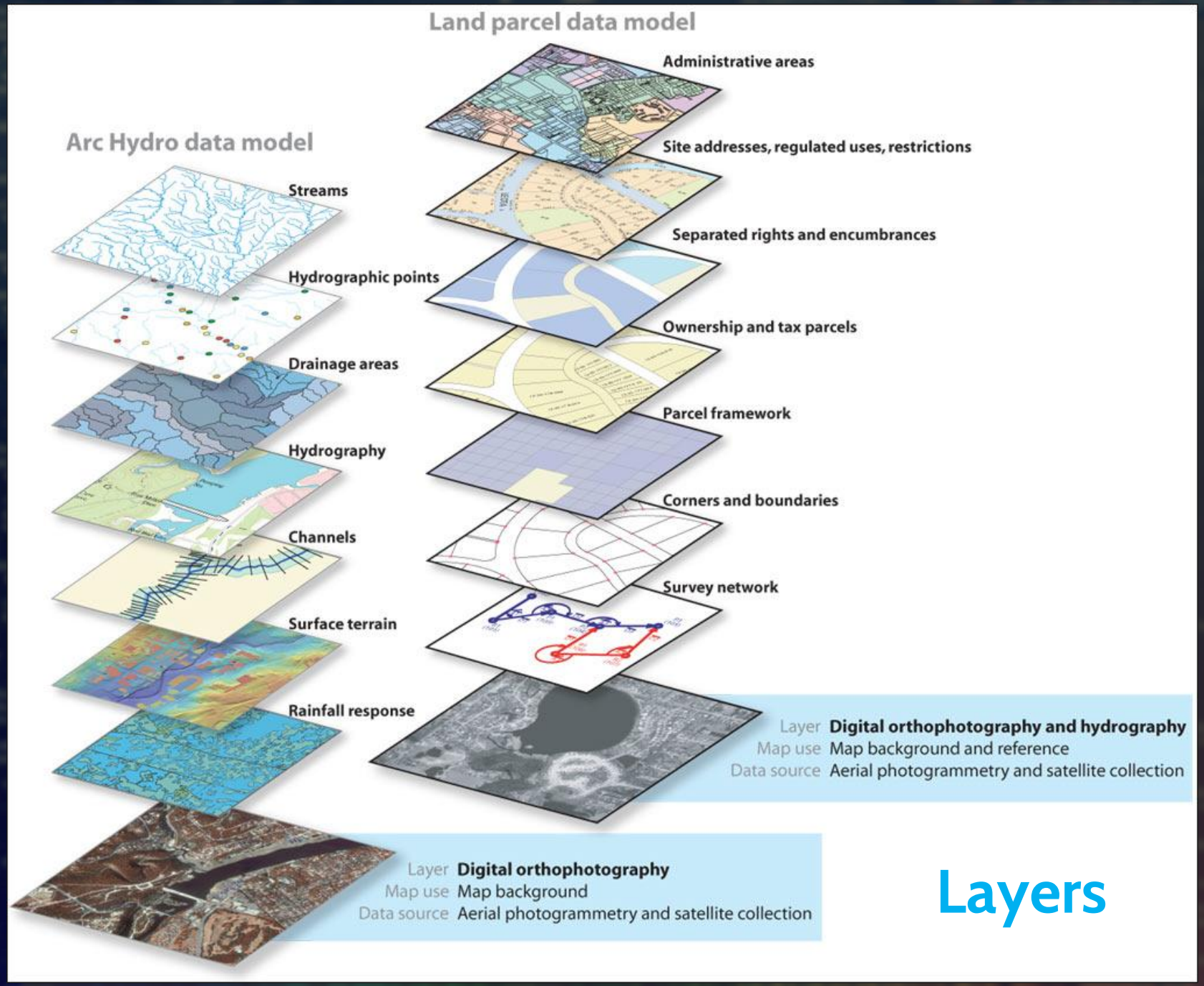
Look for in SCALERAI Search

Show selected records only Search selected records only Advanced search Help



```
>>> arcpy.Project_management("AbuDhabiTokyoGreatArc", r"C:\gould\Paulo\Designing_Better_Maps_Ed2\Databank\MyCreatedFiles
\AbuDhabiTokyoGreatArcAzimuthal", r"C:\gould\Paulo\Designing_Better_Maps_Ed2\Databank\MyCreatedFiles
\AbuDhabiTokyo_Azimuthal_Equidistant.prj")
<Result "C:\\gould\\Paulo\\Designing_Better_Maps_Ed2\\Databank\\MyCreatedFiles\\AbuDhabiTokyoGreatArcAzimuthal'>
>>> arcpy.Densify_edit("AbuDhabiTokyoGreatArc", "DISTANCE", "200 Meters")
<Result 'AbuDhabiTokyoGreatArc'>
>>> arcpy.Densify_edit("AbuDhabiTokyoGreatArcAzimuthal", "DISTANCE", "200 Meters")
<Result 'AbuDhabiTokyoGreatArcAzimuthal'>
>>>
```

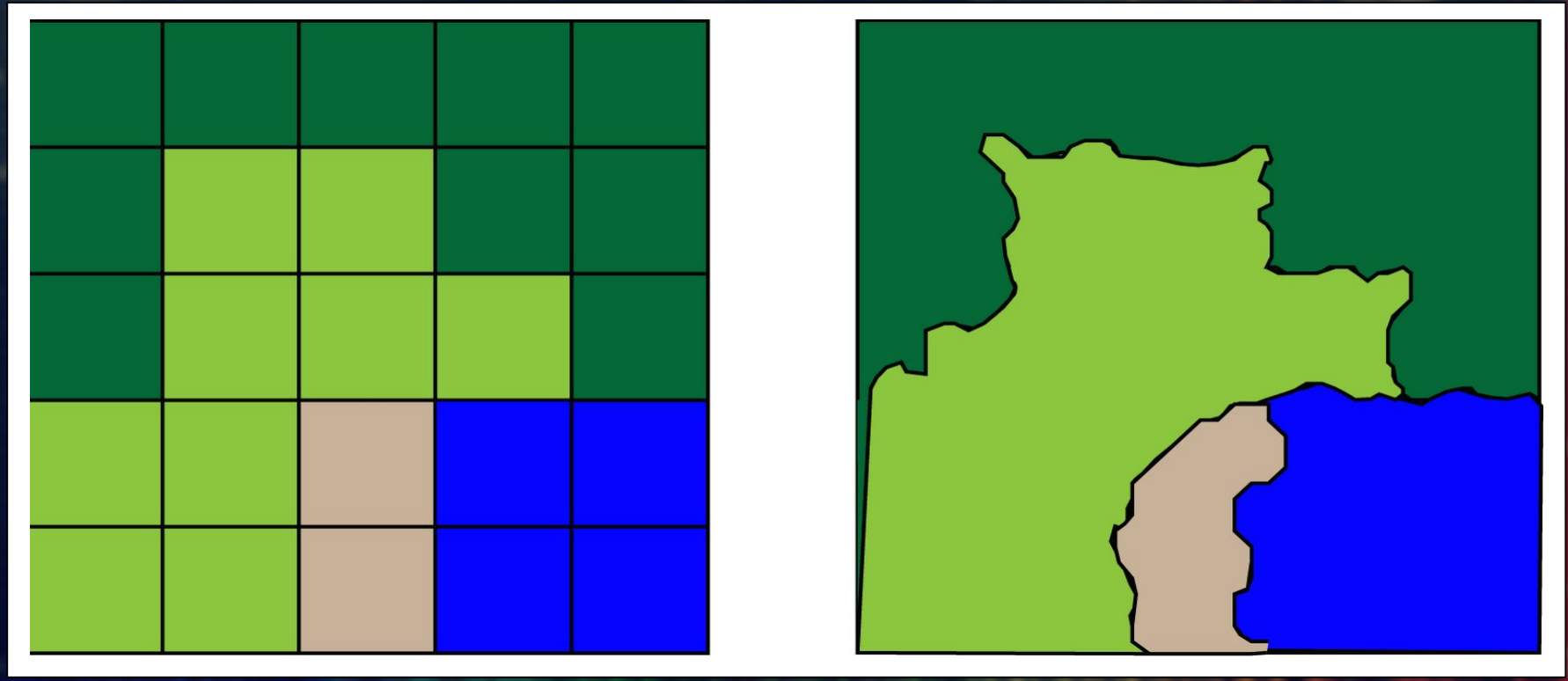
```
Start Time: Thu Jul 31 17:35:29 2014
Succeeded at Thu Jul 31 17:35:29 2014 (Elapsed
Time: 0.00 seconds)
```


Land parcel data model
Administrative areas
Site addresses, regulated uses, restrictions
Separated rights and encumbrances
Ownership and tax parcels
Parcel framework
Corners and boundaries
Survey network
Layer Digital orthophotography and hydrography
Map use Map background and reference
Data source Aerial photogrammetry and satellite collection
Arc Hydro data model
Streams
Hydrographic points
Drainage areas
Hydrography
Channels
Surface terrain
Rainfall response
Layer Digital orthophotography
Map use Map background
Data source Aerial photogrammetry and satellite collection

Layers

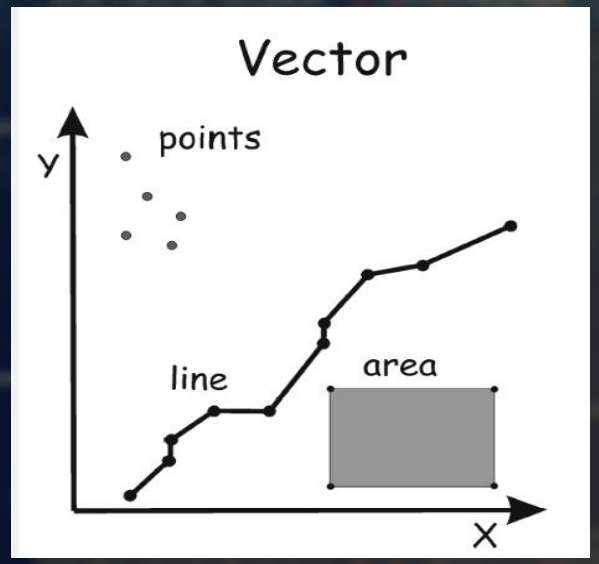
Raster

Vector

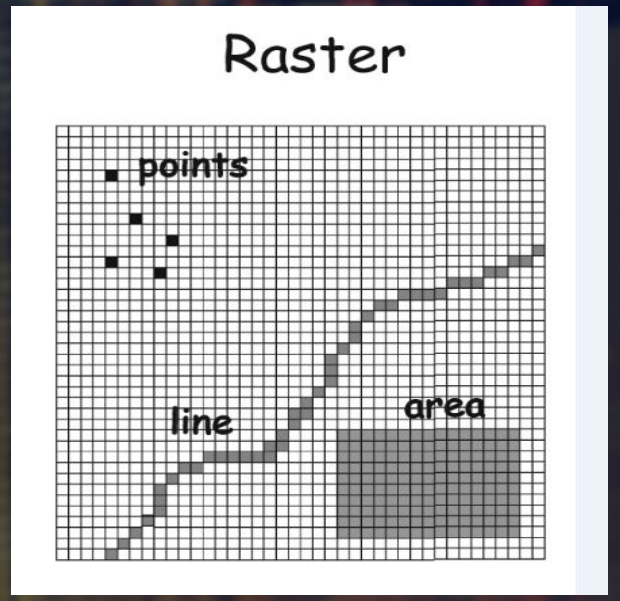


Typically in practice:

Object



Field



The text of the Madagascar.geojson file.

All the many coordinate pairs in the Betsiboka record are what tell the computer how to draw Betsiboka as a polygon - points, strung together by straight lines.

Each piece of data on this row is an "attribute" associated to the Betsiboka polygon.

Rapids in the Betsiboka River (courtesy of Wikipedia user Gloumouth1)

Betsiboka River estuary from orbit (courtesy of NASA)

GDAL

- A library (in C, with Python wrappers) and a set of executables (programs runnable from the command line) – Executable utilities are *very* versatile
- Includes OGR (vector) along with GDAL (raster)
- http://www.gdal.org/index.html
- Quickstart tutorial: https://live.osgeo.org/en/quickstart/gdal_quickstart.html
- General Docs: https://trac.osgeo.org/gdal/wiki/UserDocs
- Formats supported: 153 raster, 92 vector

WGS 84: EPSG Projection -- Spatial Reference - Mozilla Firefox

spatialreference.org/ref/epsg/4326/

Spatial Reference — epsg projection 4326 - wgs 84

Home | Upload Your Own | List user-contributed references | List all references

Search

Previous: EPSG:4324: WGS 72BE | Next: EPSG:4327: WGS 84 (geographic 3D)

Link to this Page

EPSG:4326

WGS 84 (Google it)

- **WGS84 Bounds**: -180.0000, -90.0000, 180.0000, 90.0000
- **Projected Bounds**: -180.0000, -90.0000, 180.0000, 90.0000
- **Scope**: Horizontal component of 3D system. Used by the GPS satellite navigation system and for NATO military geodetic surveying.
- **Last Revised**: Aug. 27, 2007
- **Area**: World

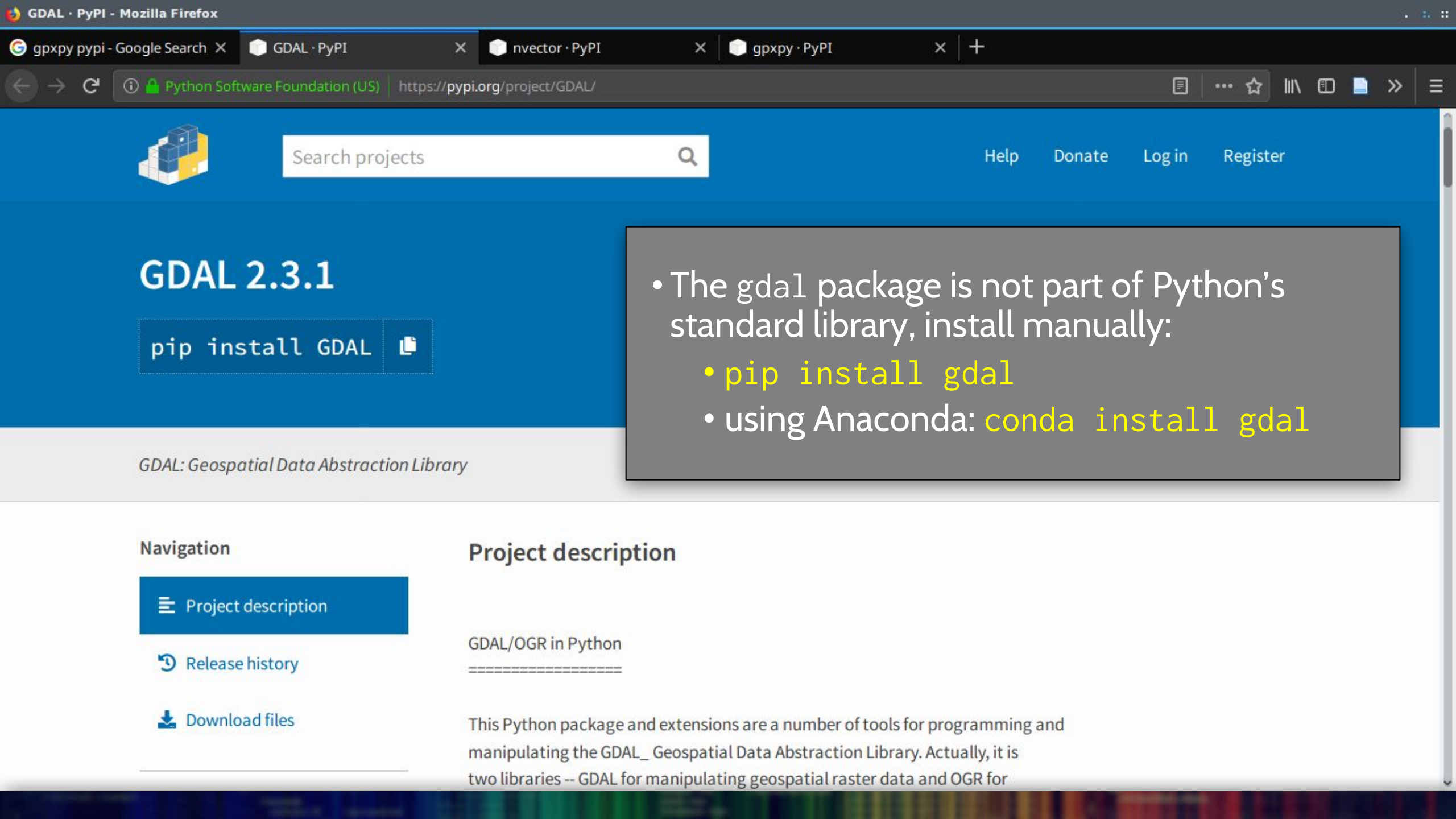
Input Coordinates: Output Coordinates:

- Well Known Text as HTML
- Human-Readable OGC WKT
- Proj4
- OGC WKT
- JSON
- GML
- ESRI WKT
- .PRJ File
- USGS
- MapServer Mapfile | Python
- Mapnik XML | Python
- GeoServer
- PostGIS spatial_ref_sys INSERT statement
- Proj4js format

About

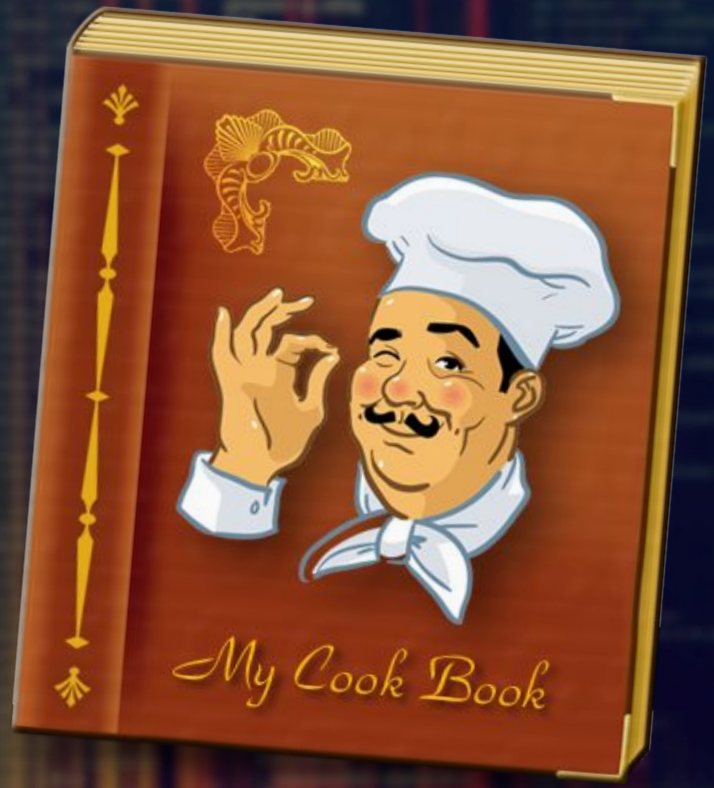
Proj4 String for WGS 84:

```
+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```


GDAL · PyPI - Mozilla Firefox
gpxpy pypi - Google Search
GDAL · PyPI
nvector · PyPI
gpxpy · PyPI
Python Software Foundation (US)
https://pypi.org/project/GDAL/
Search projects
Help
Donate
Log in
Register
GDAL 2.3.1
pip install GDAL
GDAL: Geospatial Data Abstraction Library
• The gdal package is not part of Python's standard library, install manually:
• pip install gdal
• using Anaconda: conda install gdal
Navigation
Project description
Release history
Download files
Project description
GDAL/OGR in Python
==========
This Python package and extensions are a number of tools for programming and manipulating the GDAL_ Geospatial Data Abstraction Library. Actually, it is two libraries -- GDAL for manipulating geospatial raster data and OGR for

GDAL ...with Python!

- The Cookbook! Lots of examples here: http://pcjericks.github.io/py-gdalogr-cookbook/
- Python API: http://gdal.org/python/

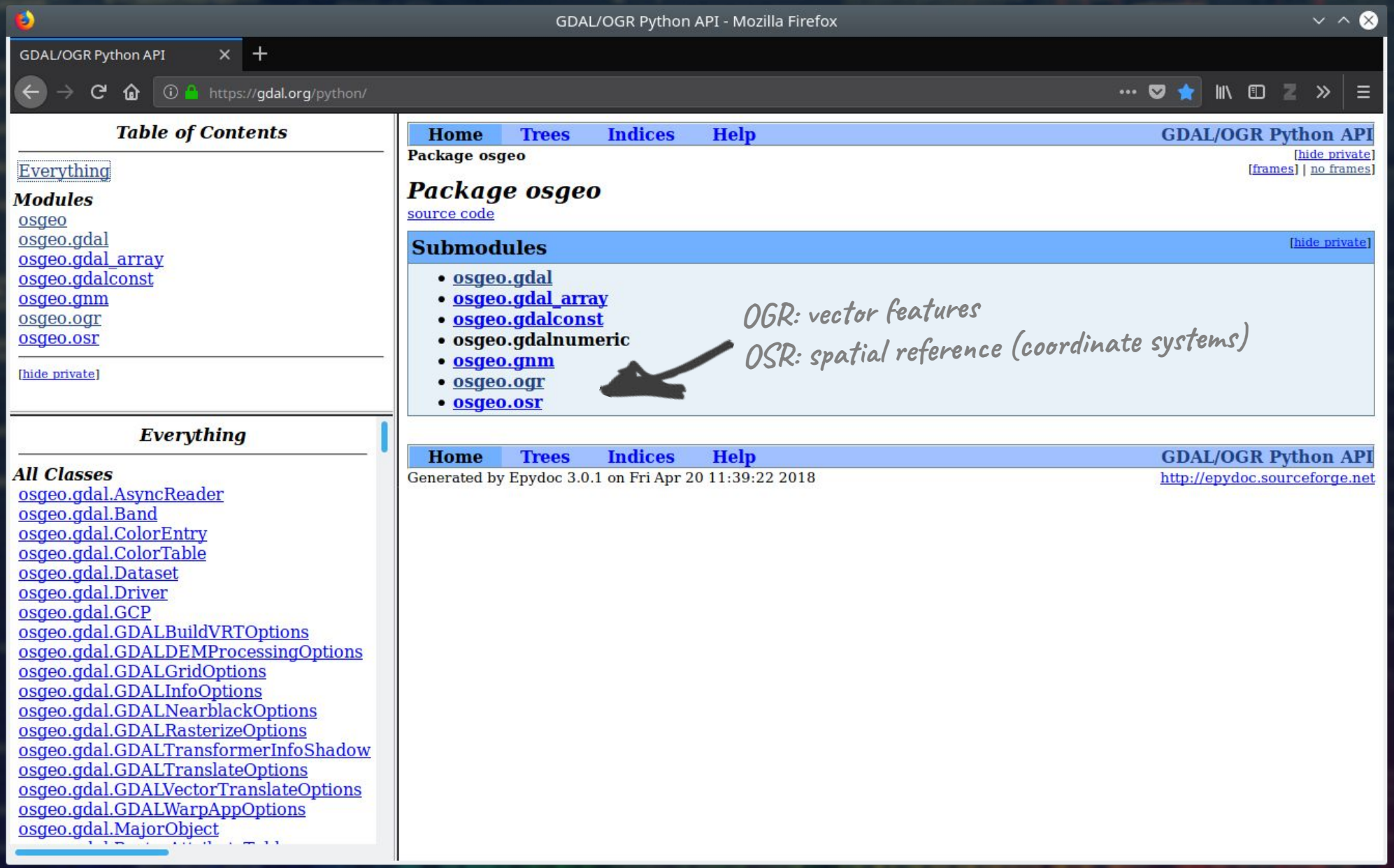


Welcome to the Python GDAL/OGR Cookbook!

This cookbook has simple code snippets on how to use the Python GDAL/OGR API. The web site is a project at GitHub and served by Github Pages. If you find missing recipes or mistakes in existing recipes please add an issue to the issue tracker.

For a detailed description of the whole Python GDAL/OGR API, see the useful API docs.

We heavily relied on Chris Garrard's excellent Geoprocessing with Python using Open Source GIS and the official GDAL/OGR Python documentation. Another great source of examples is OGR's autotest directory.


GDAL/OGR Python API - Mozilla Firefox
GDAL/OGR Python API
https://gdal.org/python/
Table of Contents
Everything
Modules
osgeo
osgeo.gdal
osgeo.gdal_array
osgeo.gdalconst
osgeo.gnm
osgeo.ogr
osgeo.osr
[hide private]
Everything
All Classes
osgeo.gdal.AsyncReader
osgeo.gdal.Band
osgeo.gdal.ColorEntry
osgeo.gdal.ColorTable
osgeo.gdal.Dataset
osgeo.gdal.Driver
osgeo.gdal.GCP
osgeo.gdal.GDALBuildVRTOptions
osgeo.gdal.GDALDEMProcessingOptions
osgeo.gdal.GDALGridOptions
osgeo.gdal.GDALInfoOptions
osgeo.gdal.GDALNearblackOptions
osgeo.gdal.GDALRasterizeOptions
osgeo.gdal.GDALTransformerInfoShadow
osgeo.gdal.GDALTranslateOptions
osgeo.gdal.GDALVectorTranslateOptions
osgeo.gdal.GDALWarpAppOptions
osgeo.gdal.MajorObject
Home Trees Indices Help GDAL/OGR Python API
Package osgeo
[hide private]
[frames] | no frames]
Package osgeo
source code
Submodules
[hide private]
osgeo.gdal
osgeo.gdal_array
osgeo.gdalconst
osgeo.gdalnumeric
osgeo.gnm
osgeo.ogr
osgeo.osr
OGR: vector features
OSR: spatial reference (coordinate systems)
Home Trees Indices Help GDAL/OGR Python API
Generated by Epydoc 3.0.1 on Fri Apr 20 11:39:22 2018
http://epydoc.sourceforge.net

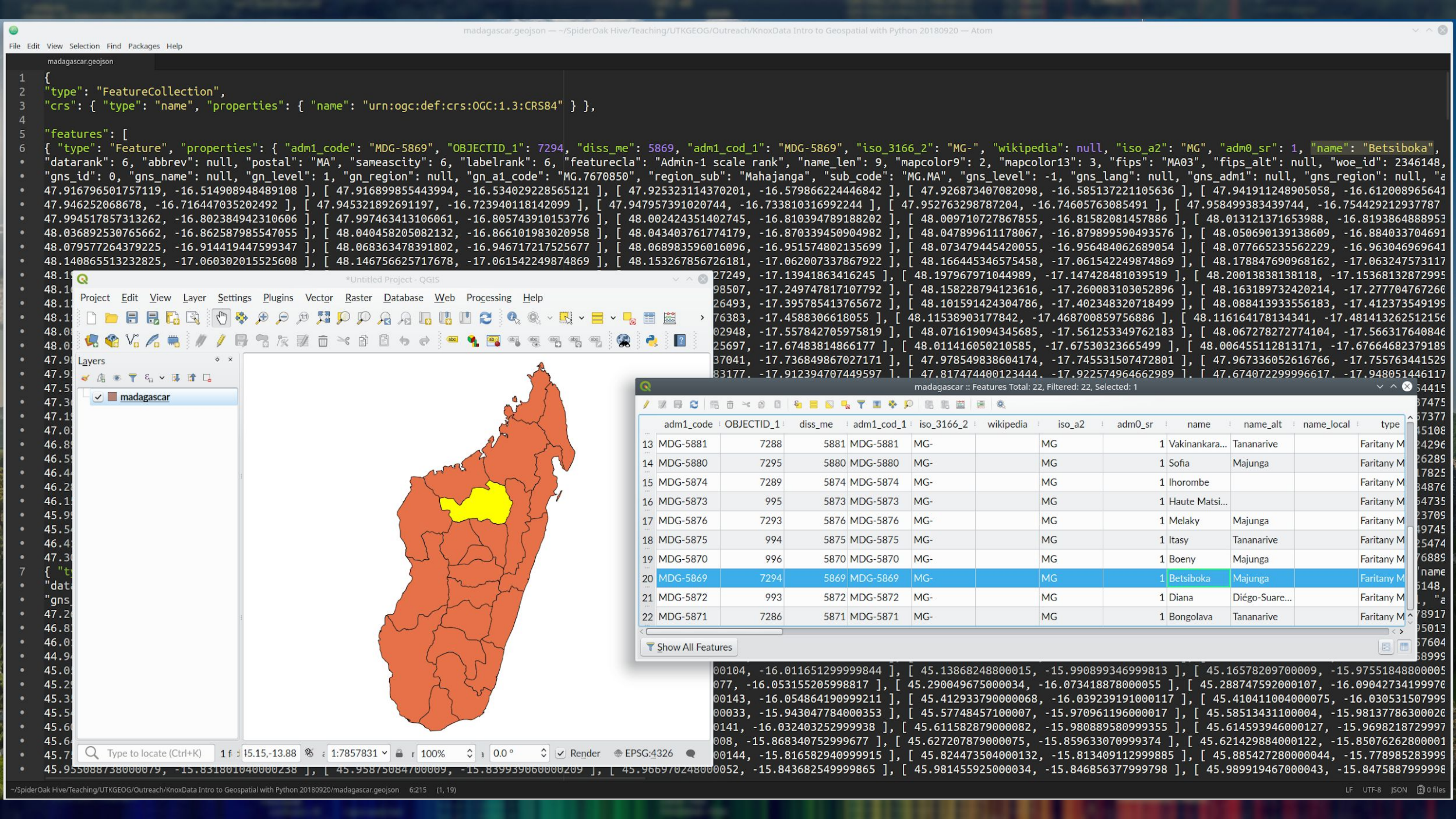
Concepts

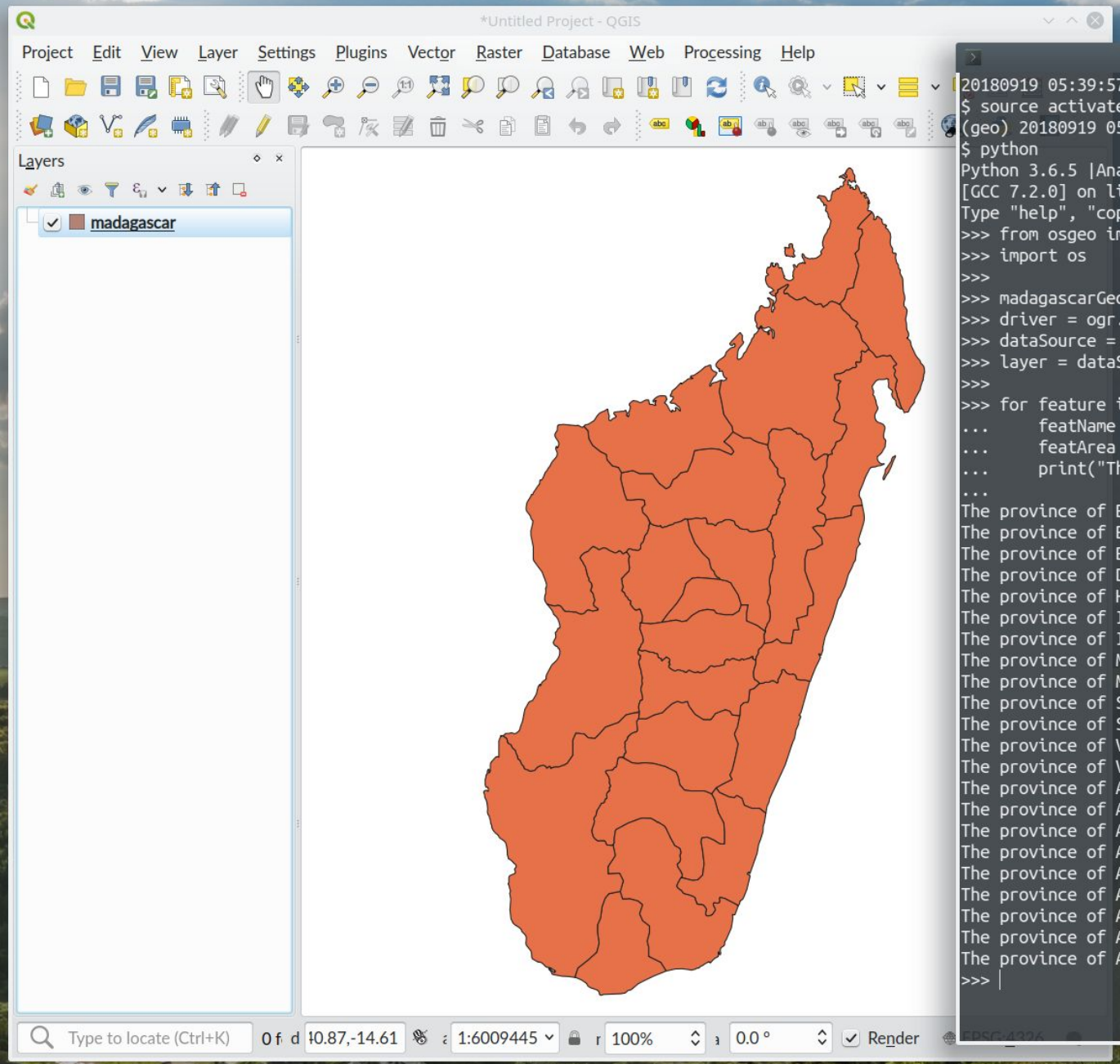
(For OGR Vector Data)

- A point is a type of geometry stored as a feature.
- A layer can have many features.
- A datasource can have many layers.
- The driver saves the datasource in a specific format.
- GDAL has drivers for different formats.

```
Driver
    Datasource
        Layer
            Feature
                Geometry
                    Point
```

http://invisibleroads.com/tutorials/gdal-shapefile-points-save.html


madagascar.geojson — ~/SpiderOak Hive/Teaching/UTKGEOG/Outreach/KnoxData Intro to Geospatial with Python 20180920 — Atom
File Edit View Selection Find Packages Help
madagascar.geojson
{
"type": "FeatureCollection",
"crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
"features": [
{ "type": "Feature", "properties": { "adm1_code": "MDG-5869", "OBJECTID_1": 7294, "diss_me": 5869, "adm1_cod_1": "MDG-5869", "iso_3166_2": "MG-", "wikipedia": null, "iso_a2": "MG", "adm0_sr": 1, "name": "Betsiboka",
"datarank": 6, "abbrev": null, "postal": "MA", "sameascity": 6, "labelrank": 6, "featurecla": "Admin-1 scale rank", "name_len": 9, "mapcolor9": 2, "mapcolor13": 3, "fips": "MA03", "fips_alt": null, "woe_id": 2346148,
"gns_id": 0, "gns_name": null, "gn_level": 1, "gn_region": null, "gn_a1_code": "MG.7670850", "region_sub": "Mahajanga", "sub_code": "MG.MA", "gns_level": -1, "gns_lang": null, "gns_adm1": null, "gns_region": null,
47.916796501757119, -16.514908948489108], [47.916899855443994, -16.534029228565121], [47.925323114370201, -16.579866224446842], [47.926873407082098, -16.585137221105636], [47.941911248905058,
*Untitled Project - QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Processing Help
Layers
madagascar
madagascar :: Features Total: 22, Filtered: 22, Selected: 1
adm1_code | OBJECTID_1 | diss_me | adm1_cod_1 | iso_3166_2 | wikipedia | iso_a2 | adm0_sr | name | name_alt | name_local | type
13 MDG-5881 7288 5881 MDG-5881 MG- MG 1 Vakinankara... Tananarive Faritany M
14 MDG-5880 7295 5880 MDG-5880 MG- MG 1 Sofia Majunga Faritany M
15 MDG-5874 7289 5874 MDG-5874 MG- MG 1 Ihorombe Faritany M
16 MDG-5873 995 5873 MDG-5873 MG- MG 1 Haute Matsi... Faritany M
17 MDG-5876 7293 5876 MDG-5876 MG- MG 1 Melaky Majunga Faritany M
18 MDG-5875 994 5875 MDG-5875 MG- MG 1 Itasy Tananarive Faritany M
19 MDG-5870 996 5870 MDG-5870 MG- MG 1 Boeny Majunga Faritany M
20 MDG-5869 7294 5869 MDG-5869 MG- MG 1 Betsiboka Majunga Faritany M
21 MDG-5872 993 5872 MDG-5872 MG- MG 1 Diana Diégo-Suare... Faritany M
22 MDG-5871 7286 5871 MDG-5871 MG- MG 1 Bongolava Tananarive Faritany M
Show All Features
Type to locate (Ctrl+K)
45.15,-13.88
1:7857831
100%
0.0 °
Render
EPSG:4326
~/SpiderOak Hive/Teaching/UTKGEOG/Outreach/KnoxData Intro to Geospatial with Python 20180920/madagascar.geojson 6:215 (1, 19)
LF UTF-8 JSON 0 files




```
paulo : python — Konsole
20180919 05:39:57 PM | paulo@OptiPlex-9010-PR | ~
$ source activate geo
(geo) 20180919 05:40:02 PM | paulo@OptiPlex-9010-PR | ~
$ python
Python 3.6.5 |Anaconda, Inc.| (default, Apr 29 2018, 16:14:56)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from osgeo import ogr
>>> import os
>>>
>>> madagascarGeoJSON = "/home/paulo/madagascar.geojson"
>>> driver = ogr.GetDriverByName("GeoJSON")
>>> dataSource = driver.Open(madagascarGeoJSON, 0)
>>> layer = dataSource.GetLayer()
>>>
>>> for feature in layer:
...     featName = feature.GetField("name")
...     featArea = feature.GetField("areasqkm")
...     print("The province of {} has an areas of {} sq km.".format(str(featName), str(featArea)))
...
The province of Betsiboka has an areas of 30676.9 sq km.
The province of Boeny has an areas of 30279.6 sq km.
The province of Bongolava has an areas of 18040.9 sq km.
The province of Diana has an areas of 19335.6 sq km.
The province of Haute Matsiatra has an areas of 21467.1 sq km.
The province of Ihorombe has an areas of 24512.8 sq km.
The province of Itasy has an areas of 6653.2 sq km.
The province of Melaky has an areas of 41630.6 sq km.
The province of Menabe has an areas of 49696.2 sq km.
The province of Sava has an areas of 23941.1 sq km.
The province of Sofia has an areas of 50971.4 sq km.
The province of Vakinankaratra has an areas of 17351.3 sq km.
The province of Vatovavy-Fitovinany has an areas of 21361.3 sq km.
The province of Alaotra-Mangoro has an areas of 27684.7 sq km.
The province of Amoron'i Mania has an areas of 17057.0 sq km.
The province of Analamanga has an areas of 16391.1 sq km.
The province of Analanjirofo has an areas of 22172.3 sq km.
The province of Androy has an areas of 15789.0 sq km.
The province of Anosy has an areas of 30573.7 sq km.
The province of Atsimo-Andrefana has an areas of 69058.2 sq km.
The province of Atsimo-Atsinanana has an areas of 16149.8 sq km.
The province of Atsinanana has an areas of 22187.7 sq km.
>>>
```

GDAL - command line

powershell

```
PS C:\Program Files\QGIS 2.18\bin> .\gdalwarp.exe --help
Usage: gdalwarp [--help-general] [--formats]
    [-s_srs srs_def] [-t_srs srs_def] [-to "NAME=VALUE"]
    [-order n | -tps | -rpc | -geoloc] [-et err_threshold]
    [-refine_gcps tolerance [minimum_gcps]]
    [-te xmin ymin xmax ymax] [-tr xres yres] [-tap] [-ts width height]
    [-ovr level|AUTO|AUTO-n|NONE] [-wo "NAME=VALUE"] [-ot Byte/Int16/...] [-wt Byte/Int16]
    [-srcnodata "value [value...]"] [-dstnodata "value [value...]"] -dstalpha
    [-r resampling_method] [-wm memory_in_mb] [-multi] [-q]
    [-cutline datasource] [-cl layer] [-cwhere expression]
    [-csql statement] [-cblend dist_in_pixels] [-crop_to_cutline]
    [-of format] [-co "NAME=VALUE"]* [-overwrite]
    [-nomd] [-cvmd meta_conflict_value] [-setci] [-oo NAME=VALUE]*
    [-doo NAME=VALUE]*
    srcfile* dstfile

Available resampling methods:
    near (default), bilinear, cubic, cubicspline, lanczos, average, mode,  max, min, med, Q1, Q3.
PS C:\Program Files\QGIS 2.18\bin>
```

Executable utility programs, and a few Python scripts, are installed by default with QGIS. Also given in Anaconda environment bin folders when you install the package **gdal**.

So many options!!

The two required inputs for gdalwarp; the things in square brackets are optional.

GDAL

A usage example

Out format is GeoTIFF. Format codes:
http://www.gdal.org/formats_list.html

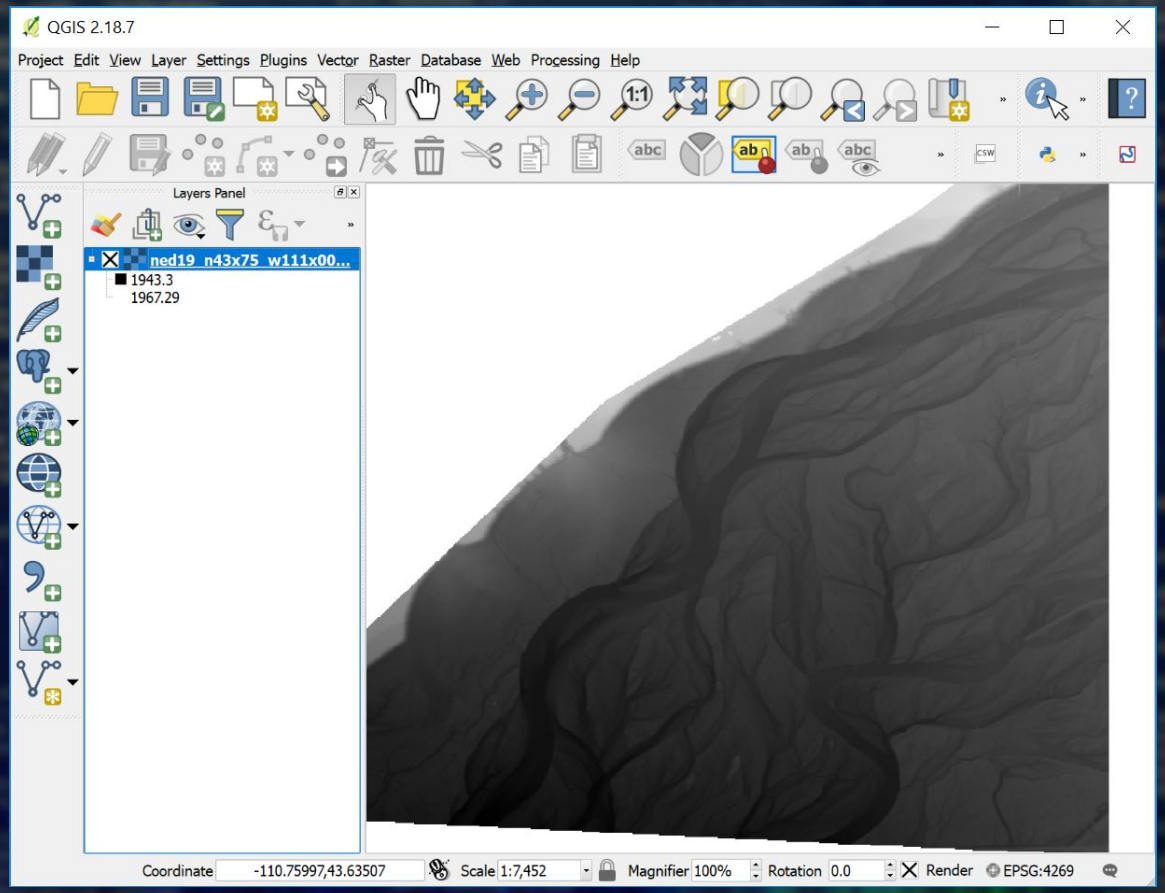
powershell

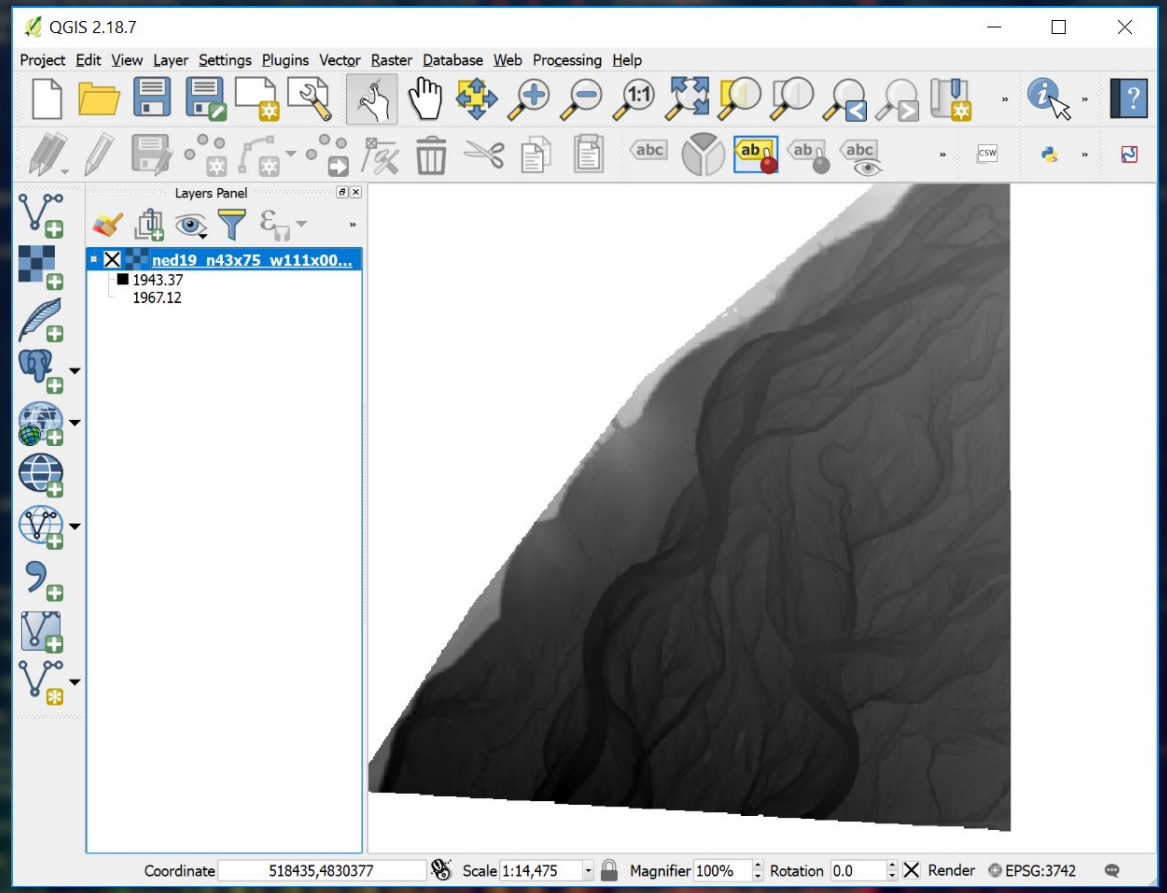
```
PS C:\Program Files\QGIS 2.18\bin> .\gdalwarp.exe -of GTIFF -t_srs EPSG:3742 C:\GISDATA\NED\ned19_n43x75_w111x00_wy_snakeriver_2007\ned19_n43x75_w111x00_wy_snakeriver_2007.img C:\GISDATA\NED\ned19_n43x75_w111x00_wy_snakeriver_2007\ned19_n43x75_w111x00_wy_snakeriver_2007_UTM12nHARN.tif
0...10...20...30...40...50...60...70...80...90...100 - done.
PS C:\Program Files\QGIS 2.18\bin>
```

Out spatial reference system is UTM 12 North
(aka EPSG:3742 from www.spatialreference.org)

Simple progress messages as the tool runs.

Before and after reprojection by gdalwarp



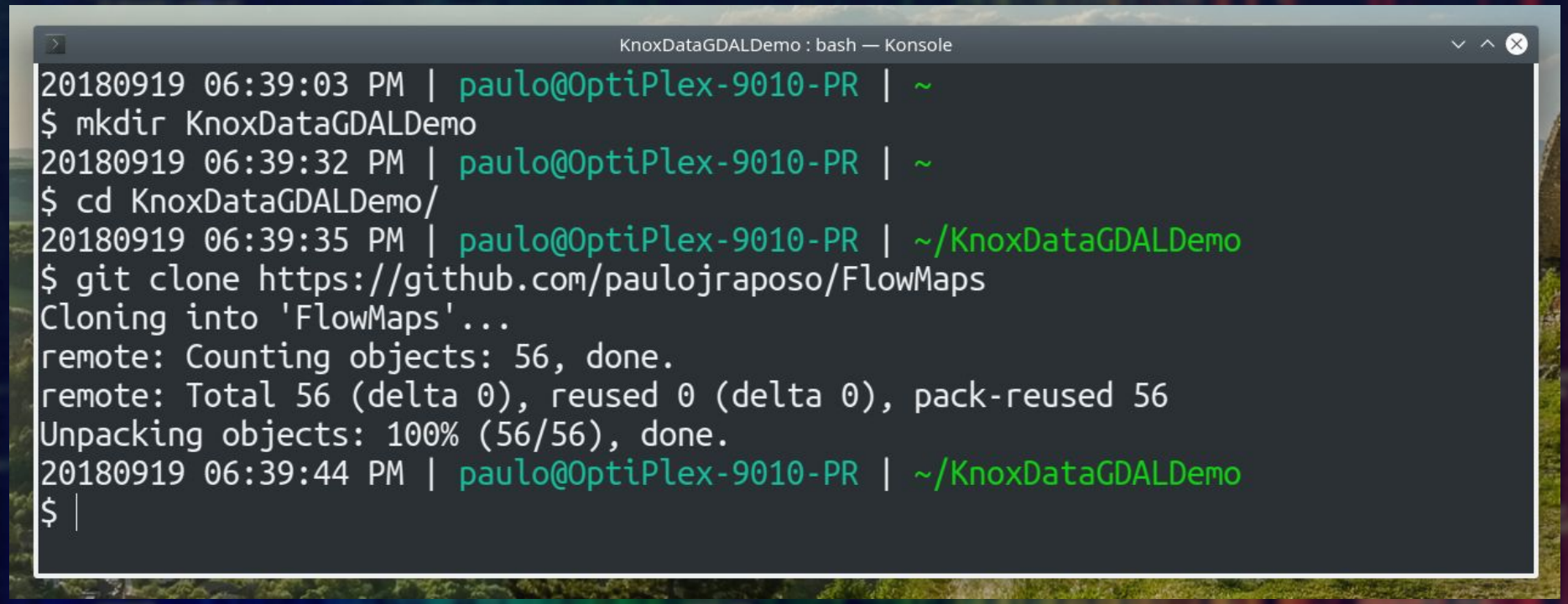


Demo: Flow Maps

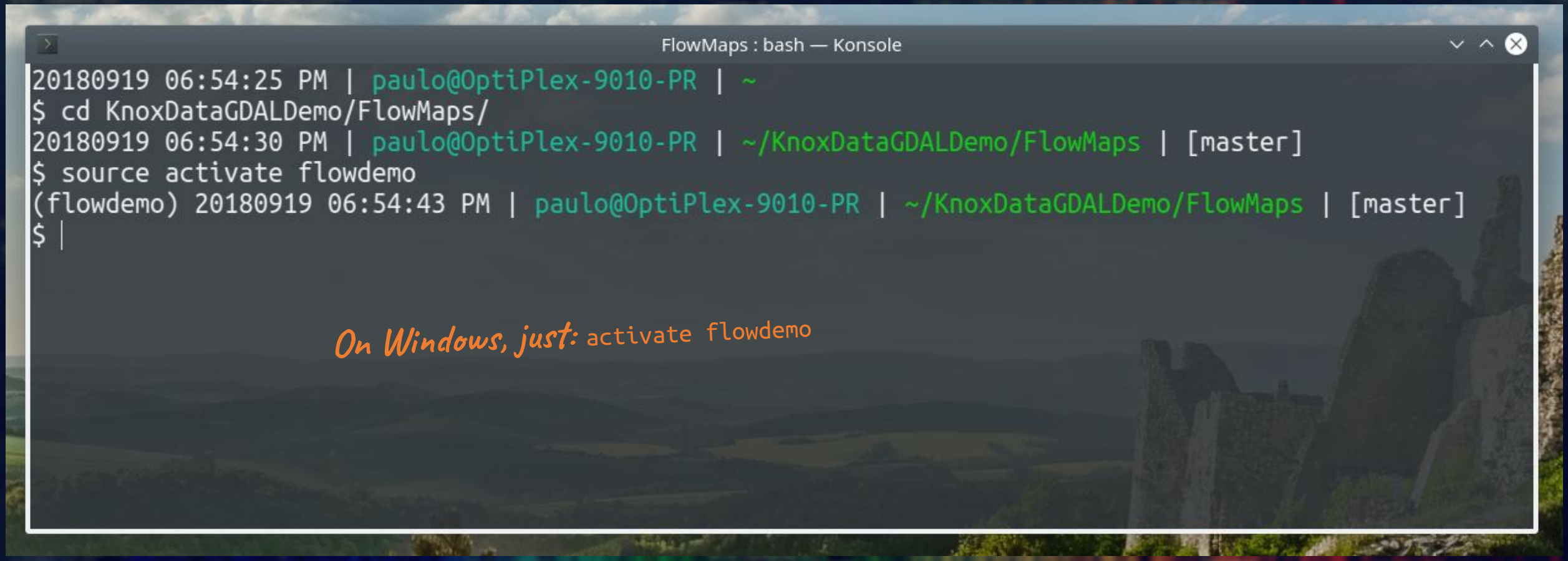
On the terminal:

```
conda create --name "flowdemo" python=3 gdal scipy shapely pyproj
```

Then clone this repo:

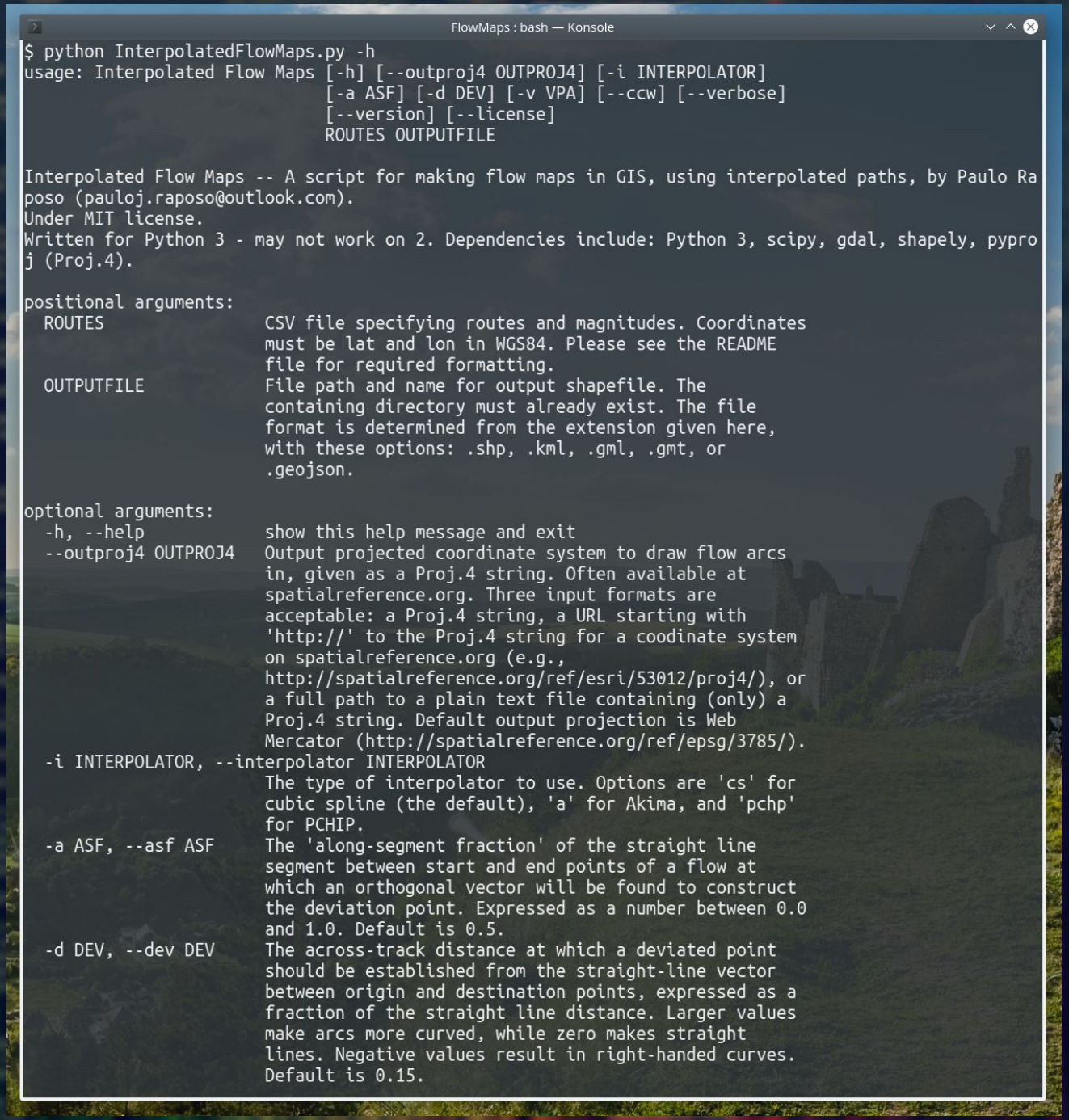


Move into the repo and activate your new conda environment:

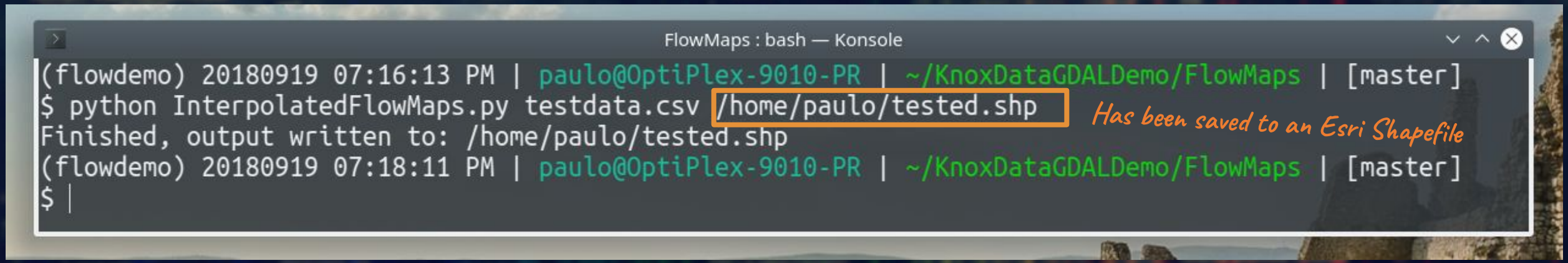


Start by reading the help screen:

`python InterpolatedFlowMaps.py -h`

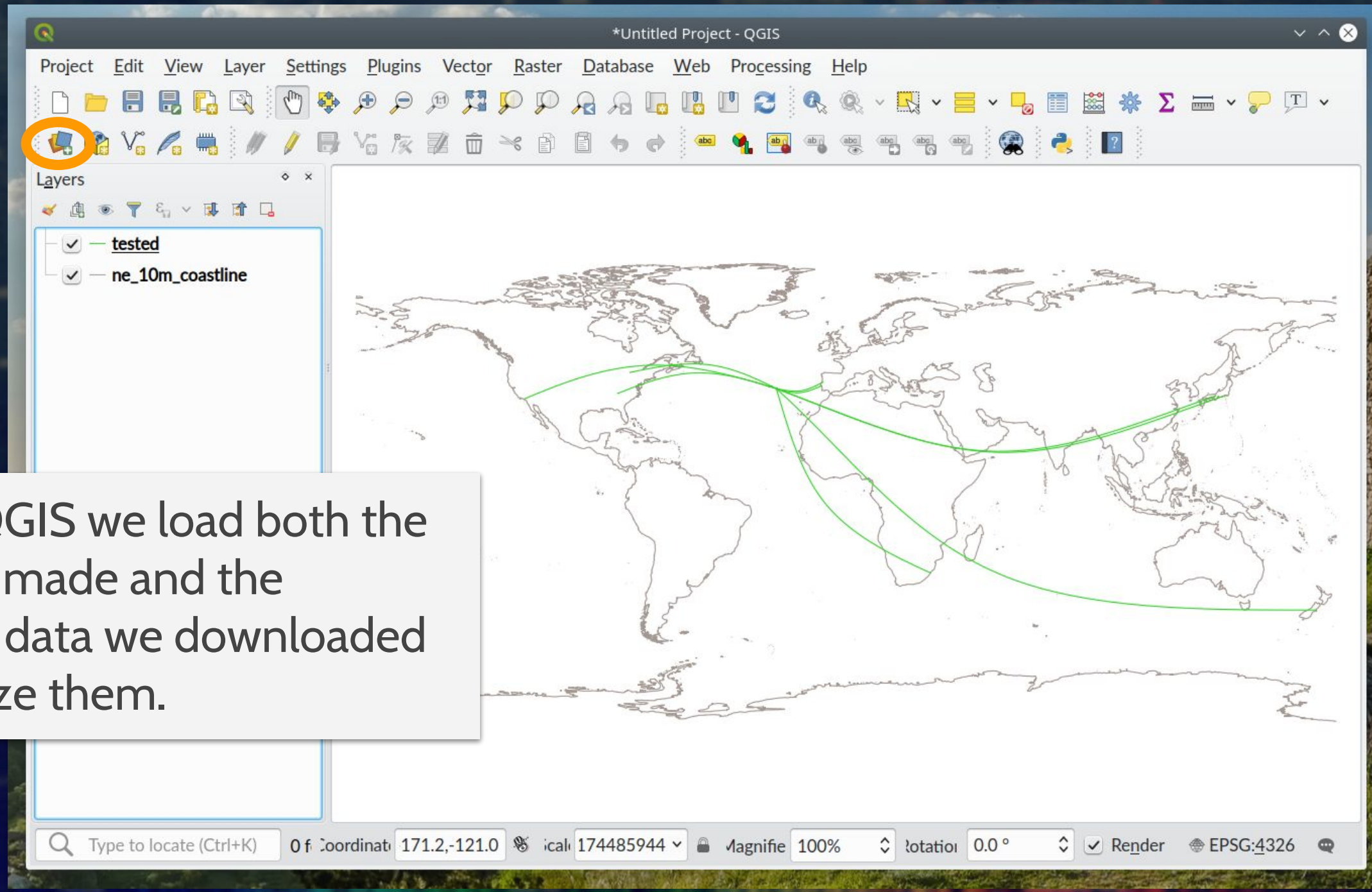


Run it against the provided testdata.csv file.

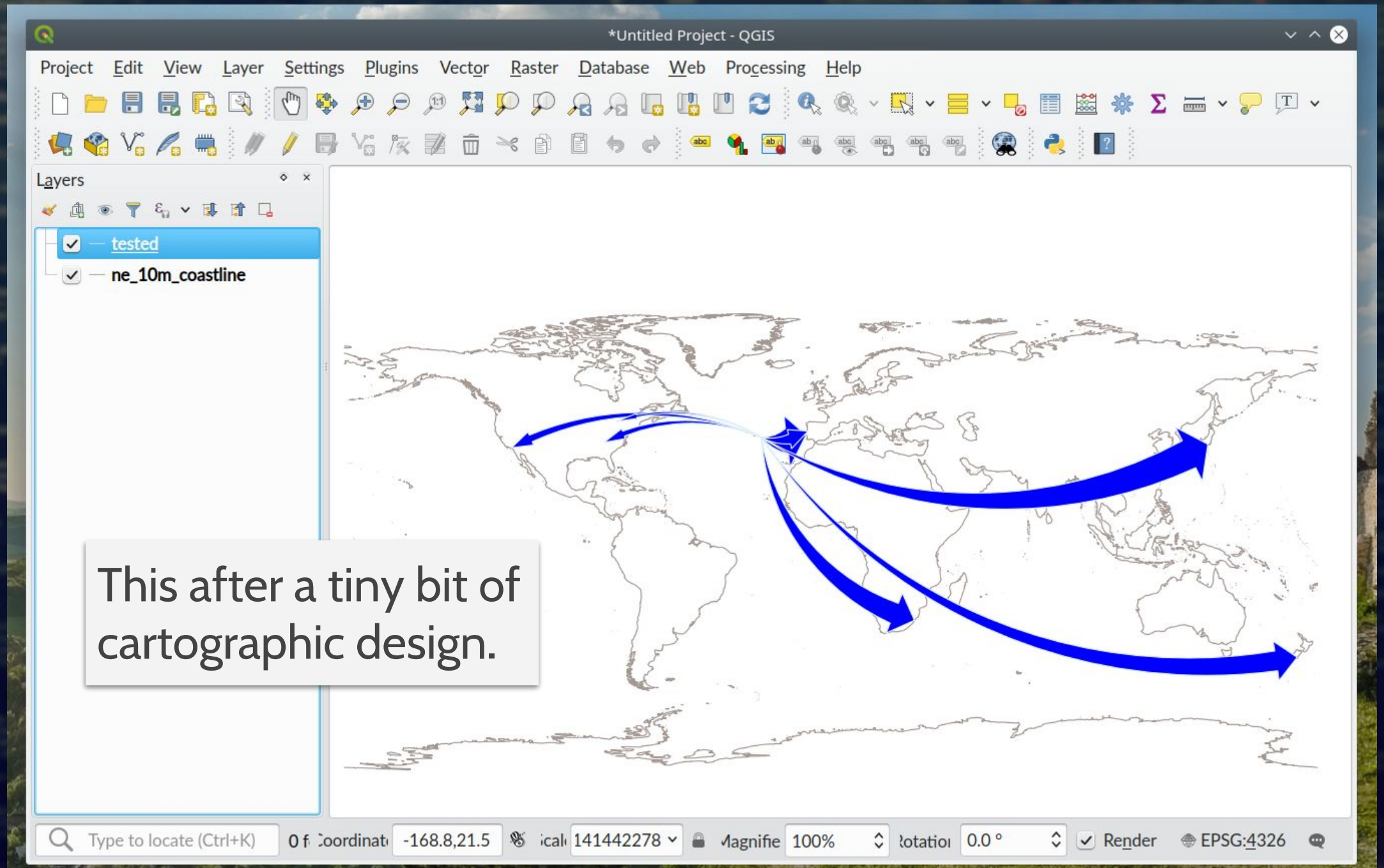


Load data here

Or drag-n-drop the .shp files into the map view.



Now in QGIS we load both the flows we made and the coastline data we downloaded to visualize them.


*Untitled Project - QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Processing Help
Layers
tested
ne_10m_coastline
This after a tiny bit of cartographic design.
Type to locate (Ctrl+K)
-168.8,21.5
141442278
100%
0.0 °
Render
EPSG:4326

Thanks!

pauloj.raposo@outlook.com
volweb.utk.edu/~praposo