Python for Open Source GIS

OSGeo Thailand Meetup 5/2019

Mr. Chaipat Nengcomma
Ayutthaya GIS Consultant.

Part I : Introduction

What Can I Do With Python + Open Source GIS ??

ตัวอย่างการนำไปใช้ของที่ Lab

Drone Mapping

Drone Mapping

Drone Mapping

DSM

Ortho Image

Drone Mapping

Drone Mapping

Land Cover classification

Object Detection

Land Cover Classification

Land Cover Classification

- Supervised Classification Multi-Sensor Data Integration Using Deep Learning
- ประมวลผลถ่ายดาวเทียมแบบ Supervised Classification โดยการพัฒนา complex model ที่วิเคราะห์ผลกับข้อมูลขนาดใหญ่ จากหลายแหล่ง แบบ Multi-Sensor มาทำการวิเคราะห์ร่วมกัน เพื่อสร้างข้อมูลผลลัพธ์การจำแนก ที่แม่นยำและมีประสิทธิภาพขึ้น

Mr. Chaipat @Ayutthaya GIS

Land Cover Classification

Mr. Chaipat @Ayutthaya GIS

Land Cover Classification

Land Cover Classification

Deep Neural Network

Mr. Chaipat @Ayutthaya GIS

Green Project

Mr. Chaipat @Ayutthaya GIS

Green Project

Mr. Chaipat @Ayutthaya GIS

Smart Farm

Crop Yield Estimation

Crop Yield Estimation

Crop Health Monitor

Crop Health Monitor

Automatic Plant Watering System

APWS

Mr. Chaipat @Ayutthaya GIS

APWS

Ayutthaya GIS Consultant Co., Ltd.

APWS

DS (Tn) <= 3

DS (Tn+x) > 8

Mr. Chaipat @Ayutthaya GIS

APWS

- Raspbian
- Python , Tensorflow + Keras , Scikit learn
- Postgresql
- Flask

python + RaspberryPi

TensorFlow™
Machine Learning
on Raspberry Pi with Tensorflow

Introduction

Introduction

- Python เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูง มากความสามารถ รองรับการทำงานข้ามแพลต์ฟอร์มระบบปฏิบัติการ, รองรับการทำงานหลากหลายประเภท
- GIS คือระบบสารสนเทศเชิงพื้นที่ +การบริหารจัดการข้อมูลตำแหน่ง
- Open Source GIS software คือ ซอฟต์แวร์ พัฒนาระบบ GIS แบบรหัสเปิด

Mr. Chaipat @Ayutthaya GIS

Introduction

- Open Source Geospatial Foundation (OSGEO) มูลนิธิ Geospatial Open Source เป็นองค์กรไม่แสวงหาผลกำไร
- ภารกิจคือการสนับสนุนและส่งเสริมการพัฒนาความร่วมมือของเทคโนโลยีและข้อมูลภูมิสารสนเทศแบบเปิด

OSGeo
Your Open Source Compass

Mr. Chaipat @Ayutthaya GIS

Open Source GIS

Desktop GIS	QGIS, GRASS GIS
Data Management API	GDAL , OGR , Proj4
Spatial Database	Postgresql+Postgis , Spatial lite (SQLite),
GIS Server	GeoServer , MapServer, Qgis Server
Web Application Framework	Openlayers, Leaflet

Open Source GIS

Python & GIS

- Python เป็นภาษาคอมพิวเตอร์ที่เป็น tools สำคัญในงานด้านการวิเคราะห์ข้อมูลเชิงพื้นที่ โดยเฉพาะลักษณะ กระบวนการทำงานแบบ อัตโนมัติ
- รองรับด้านการจัดการข้อมูล , การประมวลผล การแสดงผล รวมไปถึงความสามารถในการพัฒนา application ต่างๆ
- ภาษาผสานใน GIS Software เช่น QGIS , ArcGIS, GRASS ช่วยเพิ่มประสิทธิภาพในการทำงาน

Python & GIS

- Python เป็นเหมือนตัวเชื่อมประสาน การทำงานร่วมกันระหว่างนัก GIS กับ Developer
- มีการเติบโต และขยายของนักพัฒนา ทำให้มี opensource lib รองรับจำนวนมาก ส่วนหนึ่งต่อยอดจาก opensource gis เดิม รวมถึงการพัฒนาโมเดลใหม่ๆในลักษณะ GIS programming stack

Python & GIS

- ขยายขีดความสามารถในการเข้าถึงและวิเคราะห์ข้อมูล spatail data ในระบบงานที่ซับซ้อน
- งานด้าน Data science ,
- นำข้อมูลเชิงพื้นที่ มาสู่กระบวนการพัฒนา AI (Machine Learning , Deep Learning)
- ธุรกิจ / บริการ ที่ต่อยอดจากข้อมูลเชิงพื้นที่

Mr. Chaipat @Ayutthaya GIS

Spatial Data science

Spatial Data science

- Data science กระบวนการทางวิทยาศาสตร์เพื่อวิเคราะห์ เจาะลึก ทำความเข้าใจ เพื่อสกัดนำมูลค่าจากข้อมูลขนาดใหญ่ ข้อมูลที่มีโครงสร้างซับซ้อน
- Spatial Data = ข้อมูลเชิงปริภูมิ / ข้อมูลภูมิสารสนเทศ / ข้อมูลเชิงพื้นที่
- Key คือ ข้อมูลที่อ้างอิงกับตำแหน่งบนผิวโลก

Mr. Chaipat @Ayutthaya GIS

Data science

Mr. Chaipat @Ayutthaya GIS

Spatial Data Science

All they are combined for **data analysis** in order to ...

Support a better decision making

"The key word in data science is not data; it is science"
Jeff Leek, Data Science Specialization, Coursera.

Spatial Data science

Spatial Data science

Mr. Chaipat @Ayutthaya GIS

Data science

Data preparation *accounts for about 80% of the work of data scientists*

What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

Mr. Chaipat @Ayutthaya GIS

Data science

Machine Learning

Data science

Part II : Development

Python Geospatial Development Stack

Python Geospatial Development Stack

Layer	Package Name	Description	URL
Spatial Data IO	gdal	Interface to Geospatial data abstraction library	https://pypi.python.org/pypi/GDAL/ (link is external)
	fiona	API to OGR (Vector) layer of GDAL	http://toblerity.org/fiona/ (link is external)
	rasterio	Reading and writing geospatial raster data	https://github.com/mapbox/rasterio (link is external)

Python Geospatial Development Stack

Layer	Package Name	Description	URL
Geoprocessing	shapely	Deterministic spatial analysis	http://toblerity.org/shapely
	rasterstats	Summarizing rasters using vector geometries	https://github.com/perrygeo/python-rasterstats
	geopandas	Pandas-like spatial operations on geometric types	http://geopandas.org
	pyproj	PROJ4 Interface for cartographic transformations	https://github.com/jswhit/pyproj

Python Geospatial Development Stack

Layer	Package Name	Description	URL
Geo visualization	basemap	Plotting 2D data on maps	https://github.com/matplotlib/basemap
	cartopy	Cartographic tools	http://scitools.org.uk/cartopy
	folium	Visualization via interactive Leaflet maps	https://github.com/python-visualization/folium
	Bokeh (geo)	Interactive visualization library browsers	https://github.com/bokeh
	datashader	Big data visualization graphics pipeline	https://github.com/bokeh/datashader

Python Geospatial Development Stack

Layer	Package Name	Description	URL
Spatial Statistical Analysis	PySAL	Spatial data analysis	http://pysal.org
	pykriging	Geostatistics	http://pykriging.com/
	OSMnx + Networkx	network analysis , OpenStreetMap Data Analysis	https://osmnx.readthedocs.io/en/stable/ https://networkx.github.io/documentation/networkx-1.10/overview.html

Mr. Chaipat @Ayutthaya GIS

Python Geospatial Development Stack

Layer	Package Name	Description	URL
Web & Data Service	OWSlib	Client programming interface to OGC web service	https://geopython.github.io/OWSLib
	PyWPS	OGC WPS Framework for geospatial data	https://pywps.org/
	MapProxy	Open source proxy for geospatial data. , Tile cache & WMS server	https://mapproxy.org/

Mr. Chaipat @Ayutthaya GIS

Python Geospatial Installation

Installation

- ติดตั้งผ่าน pip ได้เกือบทั้งหมด
- บาง lib อาจจะต้องมีการ compile source code
- 99% รองรับการทำงานกับ Window OS รองรับ Anapip
- รองรับการทำงานบน Linux / Mac

Mr. Chaipat @Ayutthaya GIS

Installation

 # Install shapely
conda install -c conda-forge shapely
 # Install geopandas
conda install -c conda-forge geopandas
 # Install cartopy
conda install -c conda-forge cartopy
 # Install networkx
conda install networkx
 # Install osmnx
conda install -c conda-forge osmnx
 # Install statsmodels
conda install statsmodels
 # Install folium
conda install -c conda-forge folium
 # Install bokeh
conda install bokeh

 # Install shapely
pip install shapely
 # Install geopandas
pip install geopandas
 # Install cartopy
pip install cartopy
 # Install networkx
pip install networkx
 # Install osmnx
pip install osmnx
 # Install statsmodels
pip install statsmodels
 # Install folium
pip install folium
 # Install bokeh
pip install bokeh

Installation

- สามารถ import ตัว lib ได้แบบไม่มี error
- หรือ สามารถรัน print() ดูรายละเอียดของการติดตั้งและ package ได้แปล ว่าพร้อมจะทำงาน

import geopandas
import scipy
import shapely
import matplotlib.pyplot as plt
import pysal
import bokeh
import cartopy
import statsmodels

Mr. Chaipat @Ayutthaya GIS

Google Colaboratory

Google Colaboratory

- free Jupyter notebook environment
- Support Python 2 , 3
- Linux OS
- execute code, save and share code & result
- access powerful computing resources {CPU/GPU/TPU}
- Data storage : Google Drive

Mr. Chaipat @Ayutthaya GIS

GoogleColaboratory

Google Colaboratory

Mr. Chaipat @Ayutthaya GIS

Google Colaboratory

https://colab.research.google.com/notebooks/basic_features_overview.ipynb
https://colab.research.google.com

Google Colaboratory

Mr. Chaipat @Ayutthaya GIS

Google Colaboratory

Mr. Chaipat @Ayutthaya GIS

Google Colaboratory

Cloud Linux Server
-GPU: 1x Tesla K80 , compute 3.7, having 2496 CUDA cores , 12GB GDDR5 VRAM

-CPU: 1x single core hyper threaded Xeon Processors @2.3Ghz

-RAM: ~12.6 GB Available

-Disk: ~33 GB Available

Google Drive

```
from google.colab import drive
drive.mount('/content/gdrive/')
data_dir = os.path.join('/content/gdrive', 'My Drive','colab_notebooks')
os.listdir(data_dir)
```


Spatial Data Model

Spatial Data Model

- การสร้างแบบจำลองข้อมูลเชิงพื้นที่ในระบบสารสนเทศ
- Spatial Data Model
  a. Vector Data model
  b. Raster Data Model

Spatial Data Model

Spatial Data Model

- Vector Data Model คือ geometric objects รหัสเชิงเลข ที่ใช้นิยามตัวแทนวัตถุในระบบคอมพิวเตอร์
- Geometry Object โดยมีการระบุพิกัดตำแหน่งอ้างอิง และ index เชื่อมโยงกับข้อมูลบรรยายที่เกี่ยวข้อง
- สามารถแบ่ง Geometry Object พื้นฐาน ได้แก่ Point , Line และ Polygon
- ยังสามารถแยกย่อยตามระนาบทั้ง 2D และ 3D

Spatial Data Model

Spatial Data Model & GeoPandas

- Python มี module ในการจัดการข้อมูล geometry หลายตัว ใน workshop นี้จะเรียนรู้จักการจัดการด้วย *Geopandas*
- GeoPandas เป็น open source package สร้างเพื่อการทำงานกับข้อมูล geospatial ลักษณะของการผูกรวม
- โดยการประมวลผลด้าน data model เบื้องหลังมี shapely (GEOS C++ library) ดำเนินการ
-

Spatial Data Model & GeoPandas

Mr. Chaipat @Ayutthaya GIS

Point

- “จุด” ตัวแทน ประกอบด้วยตัวเลขพิกัด 2 มิติ หรือ 3 มิติ บน space
- ทดลองทำการสร้าง point จากกลุ่มข้อมูลพิกัด 2 มิติโดยใช้คลาส Geoseries
- จากนั้นทำการสร้าง geometry object สำหรับ Geodataframe บน pandas โดยทำการระบุระบบพิกัดอ้างอิง(CRS) เป็น EPSG:4326

Mr. Chaipat @Ayutthaya GIS

Point

```
#Point
gs =  geopandas.GeoSeries([Point(-120, 45), Point(-121.2, 46), Point(-122.9, 47.5)])
gs.crs = {'init': 'epsg:4326'}
print(gs)

#Plot geoseries
gs.plot(marker='*', color='red', markersize=100, figsize=(6,6))
plt.xlim([-123, -119.8])
plt.ylim([44.8, 47.7]);
```

Mr. Chaipat @Ayutthaya GIS

Point

```
0       POINT (-120 45)
1     POINT (-121.2 46)
2    POINT (-122.9 47.5)
dtype: object
```

Mr. Chaipat @Ayutthaya GIS

Line

- “เส้น” ตัวแทนประกอบด้วย set ของจุดเชื่อมต่อกัน
- ทำการสร้าง Linestring object จากกลุ่มข้อมูลของ point แบบ 2 มิติ ซึ่ง point จะเป็นตัวแทนของ node ที่อยู่บน Line ที่สร้างขึ้น

Line

```
# Linestring
mline1 = LineString([
    Point(0, 0),
    Point(0, 1),
    Point(1, 1),
    Point(1, 2),
    Point(3, 4),
    Point(5, 6),
])
mline2 = LineString([
    Point(6, 10),
    Point(10, 14),
    Point(11, 12),
    Point(12, 15),
])

#build line
gs_l = geopandas.GeoSeries([mline1, mline2])
print(gs_l)

#plot geoseries
gs_l.plot()
```


Line

```
0    LINESTRING (0 0, 0 1, 1 1, 1 2, 3 4, 5 6)
1    LINESTRING (6 10, 10 14, 11 12, 12 15)
dtype: object
<matplotlib.axes._subplots.AxesSubplot at 0x7f60c52b9358>
```

Polygon

- พื้นที่รูปปิด ตัวแทนของรูปทรงปิด ที่ประกอบด้วย จุดอ้างอิง ตัวแทนตำแหน่ง node และ edge ที่แน่นอน
- ตัวอย่าง ทำการสร้าง Polygon object จากกลุ่มข้อมูลของ Point 2 มิติ ที่เป็นตัวแทนของ node ของ polygon
- จากนั้นใช้ geoseries ในการสร้าง geometry object ประเภท polygon ขึ้นมาโดยจะได้รูปปิด ที่มีทั้ง node และ edge สุดท้ายสามารถนำไปใช้คำนวณคุณสมบัติประกอบอื่นๆได้ต่อไป

Polygon

```
#Polygon
p1 = Polygon([(1.5, 0), (1.75, 0), (1.75, 2),(1.5, 2)])
p2 = Polygon([(0, 0), (1, 0), (1, 1), (0, 1)])
p3 = Polygon([(2, 0), (3, 0), (3, 1), (2, 1)])
gs_po = geopandas.GeoSeries([p1, p2, p3])
print(gs_po)

#plot geoseries
gs_po.plot()
```

Mr. Chaipat @Ayutthaya GIS

Polygon

```
0    POLYGON ((1.5 0, 1.75 0, 1.75 2, 1.5 2, 1.5 0))
1              POLYGON ((0 0, 1 0, 1 1, 0 1, 0 0))
2              POLYGON ((2 0, 3 0, 3 1, 2 1, 2 0))
dtype: object
<matplotlib.axes._subplots.AxesSubplot at 0x7f60c4eae400>
```

Mr. Chaipat @Ayutthaya GIS

Polygon

- พื้นที่รูปปิดจะมีคุณสมบัติ เช่น พื้นที่, เส้นรอบรูป และจุดกึ่งกลาง
- ตัวอย่าง ทำการคำนวณคุณสมบัติเรขาคณิตของ polygon พื้นฐาน ได้แก่ Area, BBOX , Centroid

```
#Centroid calculation
cents = gs_po.centroid
print('Centroid \n',cents)

#Area calculation
Area = gs_po.area
print('Area \n',Area)

# Bounding box
bbox= gs_po.bounds
print('Bounding box \n',bbox)
```

Mr. Chaipat @Ayutthaya GIS

Polygon

```
Centroid
 0    POINT (1.625 1)
1    POINT (0.5 0.5)
2    POINT (2.5 0.5)
dtype: object
Area
 0    0.5
1    1.0
2    1.0
dtype: float64
Bounding box
    minx  miny  maxx  maxy
0  1.5   0.0  1.75   2.0
1  0.0   0.0  1.00   1.0
2  2.0   0.0  3.00   1.0
```

Mr. Chaipat @Ayutthaya GIS

Spatial Data Management

Spatial Data Management

- GeoPandas อ่าน/เขียน และจัดการไฟล์ GIS format ผ่าน Fiona (OGR + Proj4)
- Plot หรือแสดงผล geometry ด้วย descartes(เดการ์ต) (matplotlib)
- Pandas จัดการข้อมูล attribute data , table และ sql query
- GeoDataframe

Spatial Data format

- GeoPandas รองรับการทำงานกับ GIS Vector File เกือบทุกประเภท เช่นเดียวกับ OGR
- รองรับประเภทหลักเช่น Shapefile, GeoJSON, KML และ GPKG

Spatial Data Management

- ทำการเปิดและอ่านไฟล์ province.shp จากโฟเดอร์ data
- ทดสอบผลการอ่านไฟล์ด้วยการ แสดงข้อมูลบน table
- แสดงข้อมูล Geometry object แบบ WKT (polygon) พร้อมข้อมูลจาก database(.dbf)
- Plot แผนที่จาก ข้อมูล geometry พร้อมระบบพิกัด

Spatial Data Management

In [94]:
```
#load gis shp file
province= gpd.read_file('./data/province_.shp')
province.head(3)
```

Out[94]:

PROV_CODE	TOT_MALE	TOT_FEMALE	TOT_ALL	VOT_MALE	VOT_FEMALE	VOT_ALL	PROV_NAM_T	PROV_NAM_E	P_CODE	geometry
57	634959	639255	1274214	405515	419521	825036	จ.เชียงราย	Changwat Chiangrai	CHR	POLYGON ((597449.21680 2262033.98613
50	787591	808264	1595855	545188	582667	1127855	จ.เชียงใหม่	Changwat Chiang Mai	CHM	POLYGON ((553594.21266 2227842.91754
58	124295	115719	240014	71744	68597	140341	จ.แม่ฮ่องสอน	Changwat Mae Hong Son	MS	POLYGON ((398866.08340 2184810.10001

In [100]:
```
province.geometry.bounds
```

Out[100]:

	minx	miny	maxx	maxy
0	5.270823e+05	2.100870e+06	6.648943e+05	2.263225e+06
1	3.988433e+05	1.906465e+06	5.598969e+05	2.227888e+06
2	3.252167e+05	1.950139e+06	4.633613e+05	2.191310e+06
3	5.716720e+05	2.079831e+06	6.713170e+05	2.182879e+06
4	6.408247e+05	1.992243e+06	7.480584e+05	2.172110e+06
5	4.872655e+05	1.902434e+06	6.186460e+05	2.147156e+06
6	5.389370e+05	1.955443e+06	6.638243e+05	2.082903e+06
7	4.652020e+05	1.926472e+06	5.338147e+05	2.068751e+06
8	8.234012e+05	1.948806e+06	1.050002e+06	2.045524e+06
9	5.948195e+05	1.895027e+06	7.318766e+05	2.033675e+06

```
province.plot(figsize=(6,5));
```

Spatial Reference System

Spatial Reference System(SRS)

- Aka coordinate reference system (CRS)
- การนิยามระบบพิกัดอ้างอิงของ Geometry model ในระบบที่ใช้เป็นตัวแทนวัตถุบนพื้นโลก
- Map Projection = โมเดลการฉายผิวโลกทรงโค้ง 3D ให้กลายเป็นแผนที่(Map) 2 มิติ

Spatial Reference System(SRS)

- ระบบพิกัดอ้างอิง (coordinate system) ของข้อมูลเชิงพื้นที่
- การระบุพิกัดตำแหน่งอ้างอิงบนแผนที่
  >Geographic coordinate system (latitude, longitude)
  >Universal Transverse Mercator (UTM)

Geographic coordinate system

- การระบุพิกัดตำแหน่งอ้างอิงบนแผนที่แบบ latitude, longitude

Mr. Chaipat @Ayutthaya GIS

Geographic coordinate system

EPSG:4326

WGS 84 -- WGS84 - World Geodetic System 1984, used in GPS

Attributes

Unit: degree (supplier to define representation)
Geodetic CRS: WGS 84
Datum: World Geodetic System 1984
Ellipsoid: WGS 84
Prime meridian: Greenwich
Data source: OGP
Information source: EPSG. See 3D CRS for original information source.
Revision date: 2007-08-27

Scope: Horizontal component of 3D system. Used by the GPS satellite navigation system and for NATO military geodetic surveying.
Area of use: World.
Coordinate system: Ellipsoidal 2D CS. Axes: latitude, longitude. Orientations: north, east. UoM: degree

```
GEOGCS["WGS 84",
  DATUM["WGS_1984",
    SPHEROID["WGS 84",6378137,298.257223563,
      AUTHORITY["EPSG","7030"]],
    AUTHORITY["EPSG","6326"]],
  PRIMEM["Greenwich",0,
    AUTHORITY["EPSG","8901"]],
  UNIT["degree",0.0174532925199433,
    AUTHORITY["EPSG","9122"]],
  AUTHORITY["EPSG","4326"]]
```

Mr. Chaipat @Ayutthaya GIS

Universal Transverse Mercator (UTM)

- การระบุพิกัดตำแหน่งอ้างอิงบนแผนที่แบบ GRID
- เมืองไทยมีพิกัดตกบนกริด UTM โซน 47 และ 48 มีรหัสเลขช่วง N, P และ Q
- ตัวอย่าง 47P356482

Mr. Chaipat @Ayutthaya GIS

พื้นหลักฐานอางอิง (Datum)

- ระบบหลักฐานที่ใชอางอิงในการกำหนดคาทางแนวดิ่งและแนวราบ
- เพื่อใช้เป็นจุดบังคับทางความสูงและควบคุมตำแหนงบนแผนที่ สำหรับอ้างอิงในการหาตำแหน่งบนพื้นผิวโลก
- Horizontal Datum

  > Local Datum : ไทยใช้ Indian Datum 1975

  > Global Datum WGS 84 (World Geodetic System 1984)แ
- Vertical Datum :ไทยใช้ ค่าอ้างอิงจากระดับน้ำทะเลปานกลาง ณ เกาะหลัก จังหวัดประจวบ

Mr. Chaipat @Ayutthaya GIS

Universal Transverse Mercator (UTM)

- การระบุพิกัดตำแหน่งอ้างอิงบนแผนที่แบบ GRID
- เมืองไทยมีพิกัด SRS ประเภท UTM ที่พบได้แก่
- EPSG 32647 ,
- EPSG 24047
- EPSG 32648,
- EPSG 24048

Mr. Chaipat @Ayutthaya GIS

EPSG:32647 WGS 84 / UTM zone 47N

EPSG:32647

WGS 84 / UTM zone 47N

Attributes

Unit: metre

Geodetic CRS: WGS 84

Datum: World Geodetic System 1984

Ellipsoid: WGS 84

Prime meridian: Greenwich

Data source: OGP

Revision date: 2006-08-25

```
PROJCS["WGS 84 / UTM zone 47N",
  GEOGCS["WGS 84",
    DATUM["WGS_1984",
      SPHEROID["WGS 84",6378137,298.257223563,
        AUTHORITY["EPSG","7030"]],
      AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0,
      AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
      AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4326"]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",99],
  PARAMETER["scale_factor",0.9996],
  PARAMETER["false_easting",500000],
  PARAMETER["false_northing",0],
  UNIT["metre",1,
    AUTHORITY["EPSG","9001"]],
  AXIS["Easting",EAST],
  AXIS["Northing",NORTH],
  AUTHORITY["EPSG","32647"]]
```


EPSG:24047 Indian 1975 / UTM zone 47N

Attributes

Unit: metre

Geodetic CRS: Indian 1975

Datum: Indian 1975

Ellipsoid: Everest 1830 (1937 Adjustment)

Prime meridian: Greenwich

Data source: OGP

Revision date: 1995-06-02

```
PROJCS["Indian 1975 / UTM zone 47N",
  GEOGCS["Indian 1975",
    DATUM["Indian_1975",
      SPHEROID["Everest 1830 (1937 Adjustment)",6377276.345,300.8017,
        AUTHORITY["EPSG","7015"]],
      TOWGS84[210,814,289,0,0,0,0],
      AUTHORITY["EPSG","6240"]],
    PRIMEM["Greenwich",0,
      AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
      AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4240"]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",99],
  PARAMETER["scale_factor",0.9996],
  PARAMETER["false_easting",500000],
  PARAMETER["false_northing",0],
  UNIT["metre",1,
    AUTHORITY["EPSG","9001"]],
  AXIS["Easting",EAST],
  AXIS["Northing",NORTH],
  AUTHORITY["EPSG","24047"]]
```


EPSG:32648 WGS 84 / UTM zone 48N

EPSG:32648

WGS 84 / UTM zone 48N

Attributes

Unit: metre

Geodetic CRS: WGS 84

Datum: World Geodetic System 1984

Ellipsoid: WGS 84

Prime meridian: Greenwich

Data source: OGP

Revision date: 2006-08-25

```
PROJCS["WGS 84 / UTM zone 48N",
  GEOGCS["WGS 84",
    DATUM["WGS_1984",
      SPHEROID["WGS 84",6378137,298.257223563,
        AUTHORITY["EPSG","7030"]],
      AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0,
      AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
      AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4326"]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",105],
  PARAMETER["scale_factor",0.9996],
  PARAMETER["false_easting",500000],
  PARAMETER["false_northing",0],
  UNIT["metre",1,
    AUTHORITY["EPSG","9001"]],
  AXIS["Easting",EAST],
  AXIS["Northing",NORTH],
  AUTHORITY["EPSG","32648"]]
```

EPSG:900913 Google Maps Global Mercator

EPSG:900913

Google Maps Global Mercator -- Spherical Mercator
(unofficial - used in open source projects / OSGEO)

The unofficial code for Spherical Mercator (Google Maps, OpenStreetMap, ...
OpenLayers and other open-source projects.

See the official coordinate system EPSG:3857

Alternatives codes : 3857

Attributes

Data source: PROJ.4

Information source: other

Description: +proj=merc +a=6378137
+lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=
+nadgrids=@null +wktext +no_defs

```
PROJCS["Google Maps Global Mercator",
  GEOGCS["WGS 84",
    DATUM["WGS_1984",
      SPHEROID["WGS 84",6378137,298.257223563,
        AUTHORITY["EPSG","7030"]],
      AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0,
      AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
      AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4326"]],
  PROJECTION["Mercator_2SP"],
  PARAMETER["standard_parallel_1",0],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",0],
  PARAMETER["false_easting",0],
  PARAMETER["false_northing",0],
  UNIT["Meter",1],
  EXTENSION["PROJ4","+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null
  AUTHORITY["EPSG","900913"]]
```


Geocode

Geocode

- การแปลงจากชื่อ ที่อยู่ (address) ให้กลายเป็น ระบบพิกัดบนโลก
- รวมถึงข้อมูล IP address
- โดยมีบริการ(service) ในการแปลง ทำหน้าที่ค้นหาที่อยู่จากฐานข้อมูล sptial database แล้วคำนวณพิกัดตำแหน่ง ส่งกลับมายังปลายทาง
- เช่น Google Geocoding API , OSM Map API, Bing เป็นต้น
- Python modules ได้แก่ Geopy, Geocoder , Geopandas

Mr. Chaipat @Ayutthaya GIS

OGC Web Feature Service

WFS

- Open Geospatial Consortium (OGC)
- Web feature service โปรโตคอลมาตรฐานของ OGC พัฒนาขึ้นเพื่อบริการข้อมูล spatial data
- เชื่อมโยงการทำงานระหว่างระบบ GIS แบบ interoperability ในการเข้าถึงข้อมูลผ่าน webservice
- ข้าม platform ,ข้าม software , ข้าม data format
- อ้างอิงเพิ่มเติม

https://www.opengeospatial.org/standards/wfs

WFS

WFS

Web Feature Service (WFS) returns data.

WFS

WFS

- โปรโตคอลการทำงานเบื้องต้นของ OGC WFS
- ประกอบด้วย 3 Operation หลักคือ

  1.GetCapabilities

  2.DescribeFeatureType

  3.GetFeature

WFS - GetCapabilities

- Operation เพื่อส่งคำร้องสืบค้าไปยัง WFS server เพื่อตรวจสอบสถานะและข้อมูลที่ให้บริการ
- *http://localhost/geoserver/ows?service=WFS*

  &version=1.0.0&request=GetCapabilities

WFS - GetCapabilities


```
- <WFS_Capabilities version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wfs
  http://localhost:8080/geoserver/schemas/wfs/1.0.0/WFS-capabilities.xsd">
  - <Service>
      <Name>My GeoServer WFS</Name>
      <Title>GeoSolutions GeoServer workshop WFS</Title>
    - <Abstract>
        This is the reference implementation of WFS 1.0.0 and WFS 1.1.0, supports all WFS operations
        including transactions.
      </Abstract>
      <Keywords>WFS, WMS, GEOSERVER</Keywords>
      <OnlineResource>http://localhost:8080/geoserver/wfs</OnlineResource>
      <Fees>NONE</Fees>
      <AccessConstraints>NONE</AccessConstraints>
    </Service>
  - <Capability>
    - <Request>
      - <GetCapabilities>
        - <DCPType>
          - <HTTP>
              <Get onlineResource="http://localhost:8080/geoserver/wfs?request=GetCapabilities"/>
            </HTTP>
          </DCPType>
        - <DCPType>
          - <HTTP>
              <Post onlineResource="http://localhost:8080/geoserver/wfs"/>
            </HTTP>
          </DCPType>
        </GetCapabilities>
      - <DescribeFeatureType>
        - <SchemaDescriptionLanguage>
            <XMLSCHEMA/>
          </SchemaDescriptionLanguage>
        - <DCPType>
          - <HTTP>
              <Get onlineResource="http://localhost:8080/geoserver/wfs?request=DescribeFeatureType"/>
```


WFS - DescribeFeatureType

- Operation เพื่อส่งคำร้องสืบค้าไปยัง WFS server เพื่อเข้าถึงข้อมูลรายละเอียดและคำอธิบายของข้อมูลแต่ละ layer
- http://localhost/geoserver/ows?service=WFS

  &version=1.0.0

  &request=DescribeFeatureType

  &typename=geosolutions:Counties

WFS - DescribeFeatureType


```
− <xsd:schema elementFormDefault="qualified" targetNamespace="http://www.geo-solutions.it
  /workshop">
    <xsd:import namespace="http://www.opengis.net/gml" schemaLocation="http://localhost:8080
    /geoserver/schemas/gml/2.1.2/feature.xsd"/>
  − <xsd:complexType name="CountiesType">
    − <xsd:complexContent>
      − <xsd:extension base="gml:AbstractFeatureType">
        − <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" name="the_geom" nillable="true"
            type="gml:MultiPolygonPropertyType"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="STATEFP10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="COUNTYFP10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="COUNTYNS10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="GEOID10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="NAME10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="NAMELSAD10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="LSAD10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="CLASSFP10" nillable="true"
            type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="MTFCC10" nillable="true"
            type="xsd:string"/>
```


WFS - GetFeature

- Operation เพื่อส่งคำร้องสืบค้าไปยัง WFS server เพื่อเข้าถึงข้อมูล Spatial data ที่ต้องการ โดยการสืบค้น ต้องระบุพารามิเตอร์ สำคัญ เช่น ชื่อชั้นข้อมูล และสามารถกำหนด parameter เพิ่มเติมในการกรองเช่น bbox,
- ผลลัพธ์ ปกติจะเป็น GML3 แต่มีทางเลือกอื่นๆขึ้นกับ server เช่น geojson , wkb เป็นต้น

Mr. Chaipat @Ayutthaya GIS

WFS - GetFeature

- Operation
- http://localhost/geoserver/ows?service=WFS
- &version=1.0.0
- &request=GetFeature
- &typeName=geosolutions:Counties
- &maxFeatures=50
- &outputFormat=GML3

Mr. Chaipat @Ayutthaya GIS

WFS - GetFeature


```
                    -105.038097,40.349251 -105.038739,40.349251 -105.039041,40.349252 -105.040939,40.349256
                    -105.040274,40.349257 -105.040417,40.349257 -105.040802,40.349258 -105.042632,40.349261
                    -105.044923,40.349257 -105.046481,40.349257 -105.045742,40.349258 -105.047122,40.349257
                    -105.047122,40.349257 -105.047122,40.349274 -105.047122,40.351793,40.349273 -105.052039,40.349273
                    -105.052078,40.349272 -105.052942,40.349272 -105.054221,40.349271 -105.056537,40.34928
                    -105.056628,40.34928 -105.056672,40.349281
                  </gml:coordinates>
                </gml:LinearRing>
              </gml:outerBoundaryIs>
            </gml:Polygon>
          </gml:polygonMember>
        </gml:MultiPolygon>
      </geosolutions:the_geom>
      <geosolutions:STATEFP10>08</geosolutions:STATEFP10>
      <geosolutions:COUNTYFP10>003</geosolutions:COUNTYFP10>
      <geosolutions:COUNTYNS10>00196150</geosolutions:COUNTYNS10>
      <geosolutions:GEOID10>08003</geosolutions:GEOID10>
      <geosolutions:NAME10>Larimer</geosolutions:NAME10>
      <geosolutions:NAMELSAD10>Larimer County</geosolutions:NAMELSAD10>
      <geosolutions:LSAD10>06</geosolutions:LSAD10>
      <geosolutions:CLASSFP10>H1</geosolutions:CLASSFP10>
      <geosolutions:MTFCC10>G4020</geosolutions:MTFCC10>
      <geosolutions:CSAFP10/>
      <geosolutions:CBSAFP10>22860</geosolutions:CBSAFP10>
      <geosolutions:METDIVFP10/>
      <geosolutions:FUNCSTAT10>A</geosolutions:FUNCSTAT10>
      <geosolutions:ALAND10>6723011406</geosolutions:ALAND10>
      <geosolutions:AWATER10>98295230</geosolutions:AWATER10>
      <geosolutions:INTPTLAT10>+40.6620012</geosolutions:INTPTLAT10>
      <geosolutions:INTPTLON10>-105.4821309</geosolutions:INTPTLON10>
    </geosolutions:Counties>
  </gml:featureMember>
− <gml:featureMember>
  − <geosolutions:Counties fid="Counties.2">
    − <geosolutions:the_geom>
      − <gml:MultiPolygon srsName="http://www.opengis.net/gml/srs/epsg.xml#4269">
        − <gml:polygonMember>
          − <gml:Polygon>
            − <gml:outerBoundaryIs>
              − <gml:LinearRing>
```


Mr. Chaipat @Ayutthaya GIS

WFS - Server

WFS in thailand (gov)

http://fgds-backend.ngis.go.th/geoserver/geonode/ows?

http://gservices.gistda.or.th/geoserver/wfs

http://mapservices.energy.go.th:80/geoserver/ows?

Spatial Data Analysis

Spatial Data Analysis

- การวิเคราห์ข้อมูลเชิงพื้นที่ ประกอบด้วย

  1. Geoprocessing

  คือ การประมวลผล เพื่อสร้างข้อมูล geometry ใหม่จากข้อมูล geometry ตั้งต้น

  และส่วนการประมวลผล ระหว่าง geometry เพื่อสร้างผลลัพธ์ใหม่

  > function การแปลง geometry data และสร้างผลลัพธ์ข้อมูล attribute ใหม่

  > Buffer , Merge , Difference, Intersection, Union เป็นต้น

  > ศึกษาเพิ่มเติมจาก http://geopandas.org/reference.html

- **2. Spatial Relation**

  คือความสัมพันธ์ระหว่าง geometry ตัวบ่งบอกคุณลักษณะ และคุณสมบัติข้อมูล สำหรับการวิเคราะห์

Spatial Data Analysis

- การวิเคราะห์ข้อมูลเชิงพื้นที่ ยังรวมไปถึงการวิเคราะห์เชิงลึกอื่นๆ
- เช่น Spatial temporal analysis, Terrain analysis, Surface Analysis: Spatial Interpolation หรือ spatial statistic analysis เป็นต้น

Mr. Chaipat @Ayutthaya GIS

Geoprocessing

saylordotorg.github.io

Spatial Query

Spatial Query

- กระบวนการสืบค้นข้อมูลเชิงตำแหน่ง ที่ใช้เงื่อนไขเชิงตำแหน่งในการวิเคราะห์ข้อมูลเพื่อให้ได้ผลลัพธ์ตามต้องการ
- นำ geometry data มาร่วมในการสร้างเงื่อนไขเพื่อทำการวิเคราะห์ หรือสืบค้นหาข้อมูล จากฐานข้อมูล
- ค้นหาด้วยเงื่อนไข จาก spatial relation function ได้แก่
- contains(), crosses(), .intersects(),overlaps(),touches(), disjoint(), within()

Mr. Chaipat @Ayutthaya GIS

Spatial relation

Mr. Chaipat @Ayutthaya GIS

Spatial Query

Mr. Chaipat @Ayutthaya GIS

Thematic map

Thematic map

- การสร้างแผนที่เฉพาะเรื่อง จากการกำหนดเงื่อนไข ในฐานข้อมูล GIS
- นำผลลัพธ์จากการวิเคราะห์ข้อมูล มาสร้างเป็นแผนที่เฉพาะทาง
- อธิบายเล่าเรื่อง โดยกำหนด ลักษณะการแสดงผล สี สัญลักษณ์ ตามค่า(value)
- จากเงื่อนไขบน logical operation และ spatial operation

Mr. Chaipat @Ayutthaya GIS

Thematic map

Mr. Chaipat @Ayutthaya GIS

Thematic map

Web Mapping Application

- การพัฒนาระบบบริการข้อมูล GIS แบบออนไลน์ ผ่านอินเตอร์เน็ต
- รองรับการแสดงผล และการสืบค้นข้อมูล
- Python มีหลา่ยโมดูลที่สามารถเลือกใช้งานได้
- ตัวอย่างใช้ Geopandas + Bokeh
- นอกจากนี้ มีการเชื่อมโยงและผสานการแสดงผลข้อมูลกับ Map service ทั้ง WMS, WMST

Mr. Chaipat @Ayutthaya GIS

Web Mapping Application

Mr. Chaipat @Ayutthaya GIS

Question ???