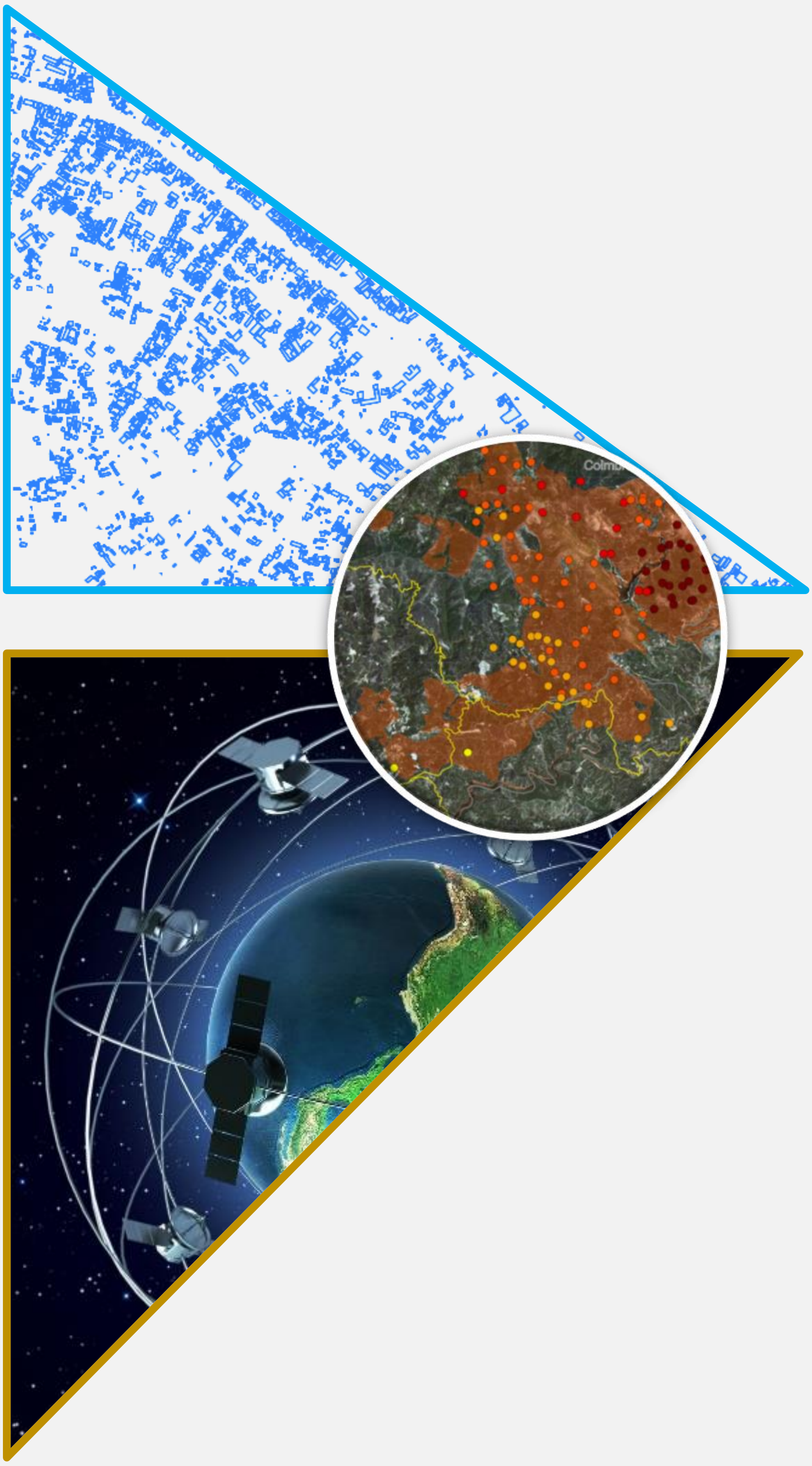
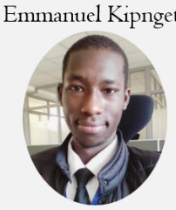
Vulnerability Assessment of Human Settlement to Floods



Integrating **Google's Open Buildings Geospatial Data** with **Earth Observation Systems** For Floods Preparedness and Response

Report Submitted for Zindi Competition on
"Hack the Continent Open Buildings Challenge"

Emmanuel Kipngetich



Student - B.Sc. GIS and Remote Sensing,
Institute of Geomatics, GIS & Remote Sensing,
Dedan Kimathi University of Technology,
Nyeri, Kenya.
emmanuel.kipngetich19@students.dkut.ac.ke

Floods and Human Settlement

Floods impact on both individuals and communities, and have social, economic, and environmental consequences.

Water-Related Disaster Management (SDG 11.5) targets to significantly reduce the number of deaths and the number of people affected and substantially decrease the direct economic losses relative to global gross domestic product caused by disasters, including water-related disasters, with a focus on protecting the poor and people in vulnerable situations.

This report, through analyzing the flood prone-zone along River Nzoia-Budalangi Kenya in Digital Earth Africa Analysis Sandbox, focuses on The United Nations Sustainable Development Goal (SDG) 6: CLEAN WATER AND SANITATION, and directly addresses Indicator 6.6.1 - Change in the extent of water-related ecosystems

The immediate impacts of flooding include loss of human life, damage to property, destruction of crops, loss of livestock, and deterioration of health conditions owing to waterborne diseases, as sewers overflow and fresh water mix with polluted water.

As communication links and infrastructure such as power plants, roads and bridges are damaged and disrupted, some economic activities may come to a standstill, people are forced to leave their homes and normal life is disrupted.

Flooding in key agricultural production areas can lead to widespread damage to crops and fencing and loss of livestock. Crop losses through rain damage, waterlogged soils, and delays in harvesting are further intensified by transport problems due to flooded roads and damaged infrastructure. The flow-on effects of reduced agricultural production can often impact well outside the production area as food prices increase due to shortages in supply.

On the other hand, flood events can result in long-term benefits to agricultural production by recharging water resource storages, especially in drier, inland areas, and by rejuvenating soil fertility by silt deposition.

Damage to public infrastructure affects a far greater proportion of the population than those whose homes or businesses are directly inundated by the flood. In particular, flood damage to roads, rail networks and key transport hubs, such as shipping ports, can have significant impacts on regional and national economies.

Flooding of urban areas can result in significant damage to private property, including homes and businesses. Losses occur due to damage to both the structure and contents of buildings. Insurance of the structure and its contents against flooding can reduce the impacts of floods on individuals or companies.

Budalang'i Constituency, Busia- Kenya

Budalang'i Constituency is an electoral constituency in Kenya. It is one of seven constituencies in Busia County (others includes; Teso North, Teso South, Nambale, Matayos, Butula and Samia). The constituency was established for the 1997 elections.

The physical setting of Budalangi at the floodplain of the Nzoia river, the neighboring Lake Victoria and increased runoff from degraded catchments are contributory factors to the flooding in this area

The Nzoia River is a 257-kilometre-long Kenyan river, rising from Cherangany hills. It passes through Kapsara, Springer , Moi's bridge then crosses to Kakamega county. It flows south and then west, eventually flowing into Lake Victoria near the town of Port Victoria, in Budalangi Subcounty. The river has a discharge of about 118 m3/s or about 3,721 million cubic meters annually, making it the second biggest river in the country by discharge.

Like many rivers across the world, the Nzoia River in western Kenya pushes over its banks annually. This results in flooding in the regions of Bunyala South.

Floods Situation in Budalang’i Constituency

On March 05 2020, Musoma primary and secondary schools in Bunyala South, Budalang'i constituency, were likely to be closed by the public health department due to flooding.

The floodwater ware from the rains currently falling in most parts of the country that time.

Pit latrines in the two learning institutions were flooded, putting the health of the children at risk.

Pupils said were forced to wade through the floodwater to reach their respective schools.

Early childhood education pupils are most affected as they are unable to wade through the floodwater and need their parents and guardians to take them to school

March 05 2020 - Flood victims in Budalangi, Busia
The Star News

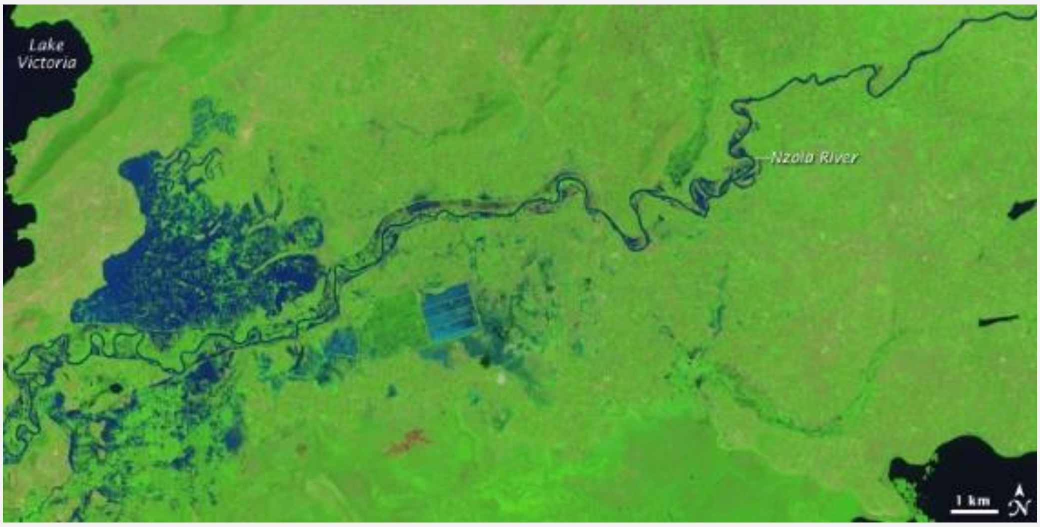
Budalangi region , residents have had to carry their belongings away from their submerged houses using boats and motorbikes, after the River Nzoia burst its banks, spilling over the land for miles around.*01:12, 07-May-2020* *Sky News*

Earth Observation Systems in Flood Monitoring

Spaceborne Earth observation is a key technology for flood response, offering valuable information to decision makers on the ground.

Through analysis from the satellite data, extent, damage, predictions and preparedness with regards to floods can be achieved

Natural Color view of the River Nzoia and its surroundings. The river, brown with runoff, snakes west across the image, emptying into Lake Victoria on the left. Muddy brown flood water spreads across the land north of the river, but it is difficult to know what is mud or bare ground, and what is water.

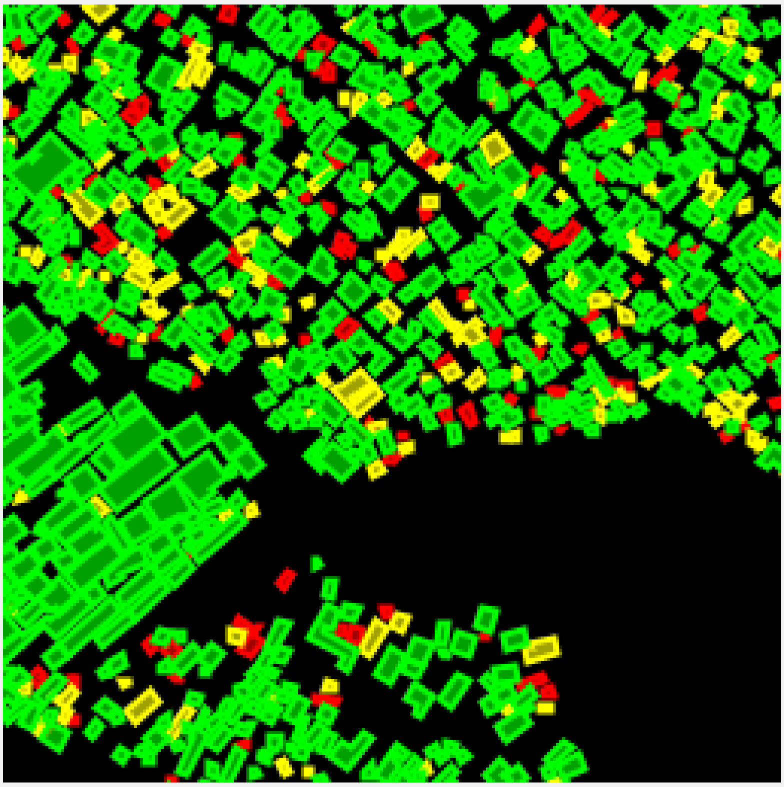


False Color View. To differentiate between silt-laden water and earth, Image Math (MNDWI) uses infrared light (short-wave infrared), which water absorbs and plants and bare earth tend to reflect. For this reason, water is black in the lower, infrared-enhanced image. Sediment in the water reflects infrared light, giving the water a blue tint. The surrounding plant-covered land is bright green, and bare earth is pale pink-tan.

Because water and land are so different in this type of image, it is useful for flood mapping.

Open Buildings for Flood Humanitarian Response

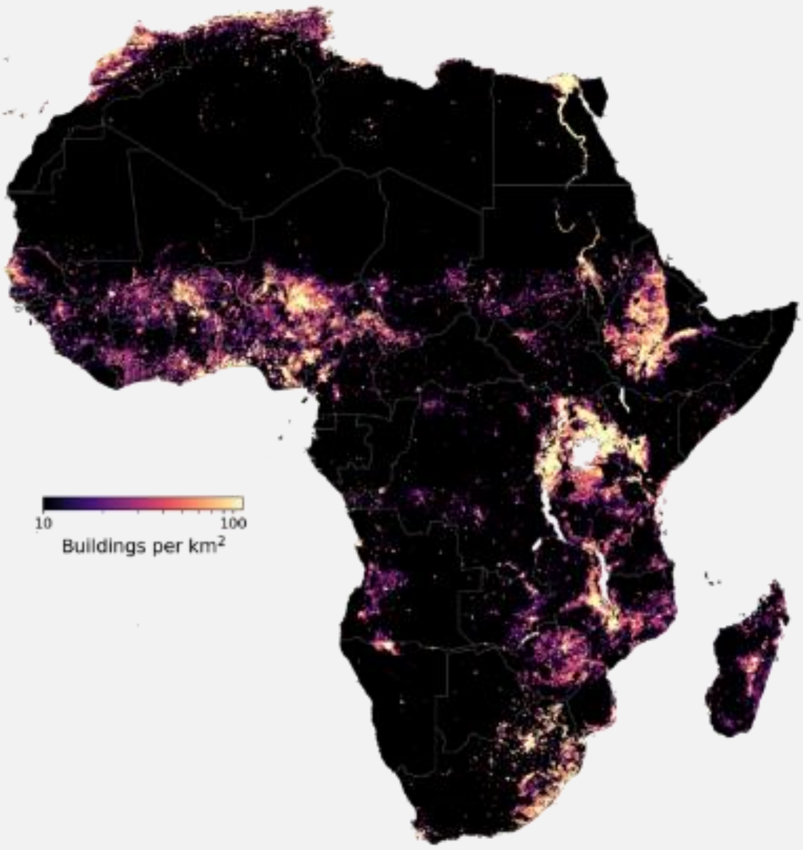
Open Buildings



A dataset of building footprints to support social good applications.

Building footprints are useful for a range of important applications, from population estimation, urban planning and humanitarian response, to environmental and climate science.

This large-scale open dataset contains the outlines of buildings derived from high-resolution satellite imagery in order to support these types of uses.



The dataset contains 516M building detections, across an area of 19.4M km^2 (64% of the African continent).

For each building in this dataset, it includes the polygon describing its footprint on the ground and a confidence score indicating how sure that it is a building.

The dataset also includes a [Plus Code] corresponding to the center of the building.

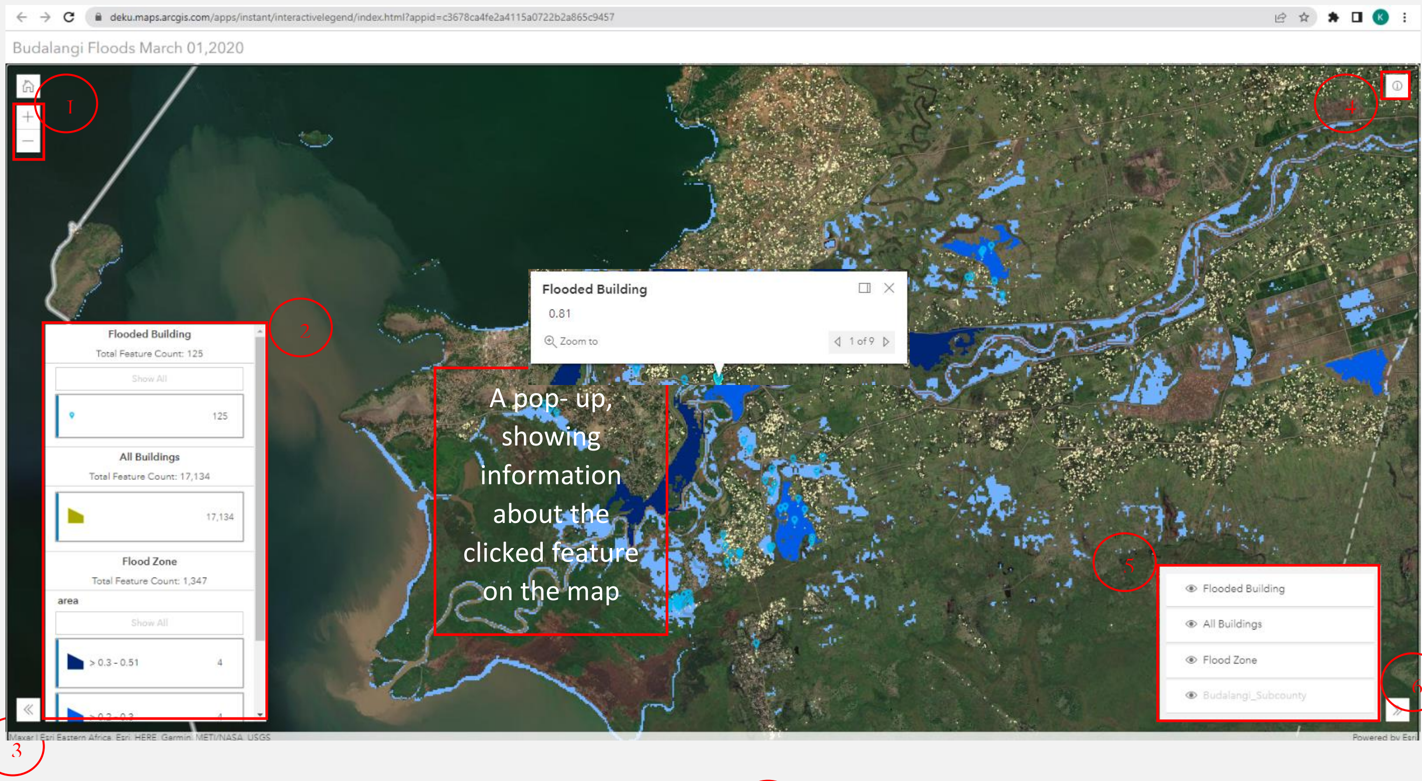
Open Buildings and Earth Observation Systems Integration for Flood Mapping

An analysis done in **Digital Earth Africa Sandbox** (a cloud-based computational platform, accessible to users for open-source scientific notebook development, that operates through a **Jupyter Lab environment.**), provided for delineation of flood prone zones in Budalangi Constituency as per the flood occurrence on 01 March ,2020 in this area

Through overlaying with Open Building Dataset, the affected built-up regions and potential hazardous built regions neighboring the flood zones are able to be pin pointed, thus facilitating measures to curb adverse effects of floods

To effectively communicate the results, an interactive GIS Web App showing the flood zone s(grouped according to area coverage), affected Buildings and all other buildings in Budalangi Constituency.

https://deku.maps.arcgis.com/apps/instant/interactivelegend/index.html?appid=c3678ca4fe2a4115a0722b2a865c9457



1	Home and Zoom Tabs	4	Introduction Panel
2	Interactive Legend	5	Layer List
3	Interactive Legend Toggle Tab	6	Layer List Toggle tab

Platform, Data and Methodology

Digital Earth Africa Sandbox

DE Africa Platform includes infrastructure and tools that support data visualization, discovery and analysis and enable users to interface with DE Africa data and services.

The **DE Africa Sandbox** is a cloud-based computational platform, accessible to users for open-source scientific notebook development, that operates through a Jupyter Lab environment. It provides users with access to data and analysis tools, democratizing access to remote-sensing data to allow for ad-hoc report generation and rapid development of new algorithm

Digital Earth Africa (DE Africa) aims to improve the lives of Africans by providing planners and policy makers with tailored Earth observation information to support better decision making and enhance sustainable development outcomes.

The **Mission** of DE Africa is to provide a routine, reliable and operational service, using Earth observations to deliver decision-ready products enabling policy makers, scientists, the private sector and civil society to address social, environmental and economic changes on the continent and develop an ecosystem for innovation across sectors

I have tailored the skills and knowledge gained upon completion of 'Introduction to the Digital Earth Africa Sandbox' online course' to generate informative insights from the remotely sensed data. These insights provide for flood preparedness and response through delineating the flood prone zones in Budalang'i, Busia, Kenya, mapping of affected buildings and the neighboring buildings potential to flooding.

Being a script-based approach, it is highly reproducible in other regions. The feature of performing analysis without downloading satellite data, gives a chance to the user to apply it in different regions, only changing the analysis parameters (time and location)

Data

- **Administrative Boundaries**

The Kenya level 3 (Country, County and Subcounty) administrative boundaries, source : IEBC, Contributor: OCHA Regional Office for Southern and Eastern Africa (ROSEA) , updated : 28 September 2021 , were obtained from The Humanitarian Data Exchange, and uploaded to DEA Sandbox https://data.humdata.org/dataset/cod-ab-ken

- **Satellite Data**

Surface Reflectance Sentinel 2 Level 2 A are pre-loaded together with other earth observation data in DE Africa enabling users to immediately start performing interactive analysis on the data without downloading the raw data.

- **Building's Dataset**

Open Buildings V1 Polygons dataset of the area of study was obtained through Google Earth Engine GEE (https://code.earthengine.google.com/12209646ee7b17a7d36cb885bd2810d2) and uploaded to DE Africa Sandbox

Methodology

This Analysis was conducted in **DE Africa Sandbox**

The **DE Africa Sandbox** is a cloud-based computational platform, accessible to users for open-source scientific notebook development, that operates through a Jupyter Lab environment and uses Python Programming language.

Step 01: Preparing the Analysis Notebook

- **Loading Packages** Time taken to run: 5.078315734863281e-05 seconds

```
%matplotlib inline

import datacube
import matplotlib.pyplot as plt
import numpy as np
import xarray as xr
import geopandas as gpd
import pandas as pd

from IPython.display import Image
from matplotlib.colors import ListedColormap
from matplotlib.patches import Patch
from odc.ui import image_aspect
from datacube.utils import geometry

from deafrica_tools.datahandling import load_ard
from deafrica_tools.bandindices import calculate_indices
from deafrica_tools.plotting import xr_animation, map_shapefile
from deafrica_tools.dask import create_local_dask_cluster
from deafrica_tools.spatial import xr_rasterize
from deafrica_tools.spatial import xr_vectorize, xr_rasterize
from datacube.utils import geometry

import json
from shapely.geometry import shape ,Point, Polygon

from odc.algo import xr_reproject

from datacube.utils.aws import configure_s3_access
configure_s3_access(aws_unsigned=True, cloud_defaults=True)
```

The datacube allows creation of an object that can retrieve data from the datacube

The package deafrica_tools contain several modules that facilitates loading, analyses and output data from Digital Earth Africa.

rgb utility function to plot a true colour image for a series of timesteps

load_ard function load all the images from Sentinel-2 and apply a cloud mask. The returned xarray.Dataset will contain analysis ready images with the cloudy and invalid pixels masked out

- **Connecting to Data cube** Time taken to run: 3.886222839355469e-05 seconds

The Open Data Cube (ODC) is an Open-Source Geospatial Data Management and Analysis Software project that helps harness the power of Satellite data.

The Africa Regional Data Cube (ARDC) currently supports open data cube (ODC) access and capacity building for five countries: Ghana, Kenya, Senegal, Sierra Leone and Tanzania

A business case for DE Africa was developed in 2018 based on a consensus and understanding across a range of stakeholders across Africa including the ARDC countries

```
dc = datacube.Datacube(app='Budalangi_Floods_Mapping')
```

datacube.Datacube class provides access to the datacube
app parameter is a unique name for the analysis which is based on the notebook file name

- **Setting up a Dask Cluster** Time taken to run: 6.103515625e-05 seconds

Dask is a tool used when working with large analyses (either in space or time) as it breaks data into manageable chunks that can be easily stored in memory.
It can also use multiple computing cores to speed up computation thus it is used to better manage memory use down and conduct the analysis in parallel

```
create_local_dask_cluster()
```

```
/env/lib/python3.8/site-packages/distributed/node.py:151: UserWarning: Port 878
7 is already in use.
Perhaps you already have a cluster running?
Hosting the HTTP server on port 40535 instead
  warnings.warn(
```

Client	Cluster
Scheduler: tcp://127.0.0.1:39595	**Workers:** 1
Dashboard: /user/emmanuel_kipngetich/proxy/40535/status	**Cores:** 4
	Memory: 28.14 GB

Step 02: Setting Up Analysis Parameters

- **Defining area of study**

Reading the 'ken_admbnda_adm2_iebc_20191031.shp' Time taken to run: 0.08147215843200684 seconds

```
Kenya_Subcounties = gpd. read_file('ken_admbnda_adm2_iebc_20191031.shp')
```

Extracting Budalangi subcounty Time taken to run: 0.002190113067626953 seconds

```
Budalangi_Subcounty = Kenya_Subcounties[Kenya_Subcounties.ADM2_EN == "Budalangi"]
```

Reprojecting Budalangi subcounty crs from WGS 86 (EPSG:4326) to WGS 84 UTM Zone 36 N (EPSG: 32636) Time taken to run: 0.0038440227508544922 seconds

WGS 84 / UTM zone 36N is a projected coordinate reference system (Cartesian 2D CS. Axes: easting, northing (E, N). Orientations: east, north. UoM: m)

```
Budalangi_Subcounty.to_crs(epsg=32636, inplace=True)
```

- **Specifying time range, Satellite data products, measurements, and resolution** Time taken to run: 0.0001571178436279297 seconds

Time range is the time period between the start and the end of nominated years of water extent analysis.

Products are the satellite data to be loaded. Landsat 8 and sentinel 2 level 2 are the satellite data availed in Digital Earth Africa Sandbox. Sentinel 2 leve 2 were used in this analysis

Measurements are the bands in the sentinel 2 level 2 to be loaded

Resolution is the ground resampling power. Since the analysis uses both 10m and 20m, (-20, 20) was defined

```
products = ['s2_l2a']

time_range = ('2018', '2020')


measurements = ['green','swir_1', 'SCL']

resolution = (-20,20)
```

Step 03: Operations on Satellite Data

- **Creating a reusable query for loading satellite data** Time taken to run: 0.005937099456787109 seconds

  The "query dictionary " syntax places the query parameters used to load data from multiple timesteps into a python dictionary object that is re-useable

```
#create a geom object from the vector file
geom = geometry.Geometry(Budalangi_Subcounty.iloc[0].geometry.__geo_interface__,
                         geometry.CRS(f'EPSG:{Budalangi_Subcounty.crs.to_epsg()}'))

water_query = {'time': time_range,
               'measurements': measurements,
               'resolution': resolution,
               'output_crs': 'EPSG:32636',
               'dask_chunks': dask_chunks,
               'geopolygon': geom,
               'group_by':'solar_day'
              }
```

- **Loading Cloud-masked Sentinel 2 level 2 data** Time taken to run: 1.0564384460449219 seconds

  Digital Earth Africa provides Sentinel-2, Level 2A (processed to Level 2A using the Sen2Cor algorithm) surface reflectance data. Surface reflectance provides standardized optical datasets by using robust physical models to correct for variations in image radiance values due to atmospheric properties, as well as sun and sensor geometry, resulting an Analysis Ready Data (ARD) product. ARD allows you to analyses surface reflectance data as is without the need to apply additional corrections. The resulting stack of surface reflectance grids are consistent over space and time, which is instrumental in identifying and quantifying environmental change

```
filters=[('erosion', 5),("closing", 2),("opening", 2),("dilation", 1)]

ds = load_ard(dc=dc,
              products=products,
              min_gooddata=min_gooddata,
              mask_filters=filters,
              **water_query
                  )
print(ds)
```

- **Masking to the area of study** Time taken to run: 0.04294919967651367 seconds

  Masking the satellite data to the area of study ensures that only imageries to the area of study are added to the notebook. This reduces the processing time and provides for statistical values within the area jurisdictions

```
#create mask
mask = xr_rasterize(Budalangi_Subcounty,ds)

#mask data
ds = ds.where(mask)

#remove SCL since we don't need it anymore
ds = ds.drop('SCL')

#convert to float 32 to conserve memory
ds=ds.astype(np.float32)
```

- **Calculating the Modified Normalized Difference Water Index (MNDWI)** Time taken to run: 0.014151811599731445 seconds

  The Modified Normalized Difference Water Index (MNDWI) uses green and SWIR bands for the enhancement of open water features. It also diminishes built-up area features that are often correlated with open water in other indices. MNDWI = (Green - SWIR) / (Green + SWIR)

  The function `calculate_indices(ds[, index, collection, …])` Takes an xarray dataset containing spectral bands, calculates one of a set of remote sensing **indices**, and adds the resulting array as a new variable in the original dataset.

```
# Calculate the chosen vegetation proxy index and add it to the loaded data set
ds = calculate_indices(ds=ds, index='MNDWI', collection='s2')

# drop green and swir since we don't need it
ds = ds.drop(['green','swir_1'])
```

- **Resampling time** series Time taken to run: 142.78148102760315 seconds

  Due to many factors (e.g., cloud obscuring the region, missed cloud cover in the fmask layer) the data will be gappy and noisy.

  One of the key reasons for generating a composite is to replace pixels classified as clouds with realistic values from the available data. This results in an image that doesn't contain any clouds. In the case of a median composite, each pixel is selected to have the median (or middle) value out of all possible values

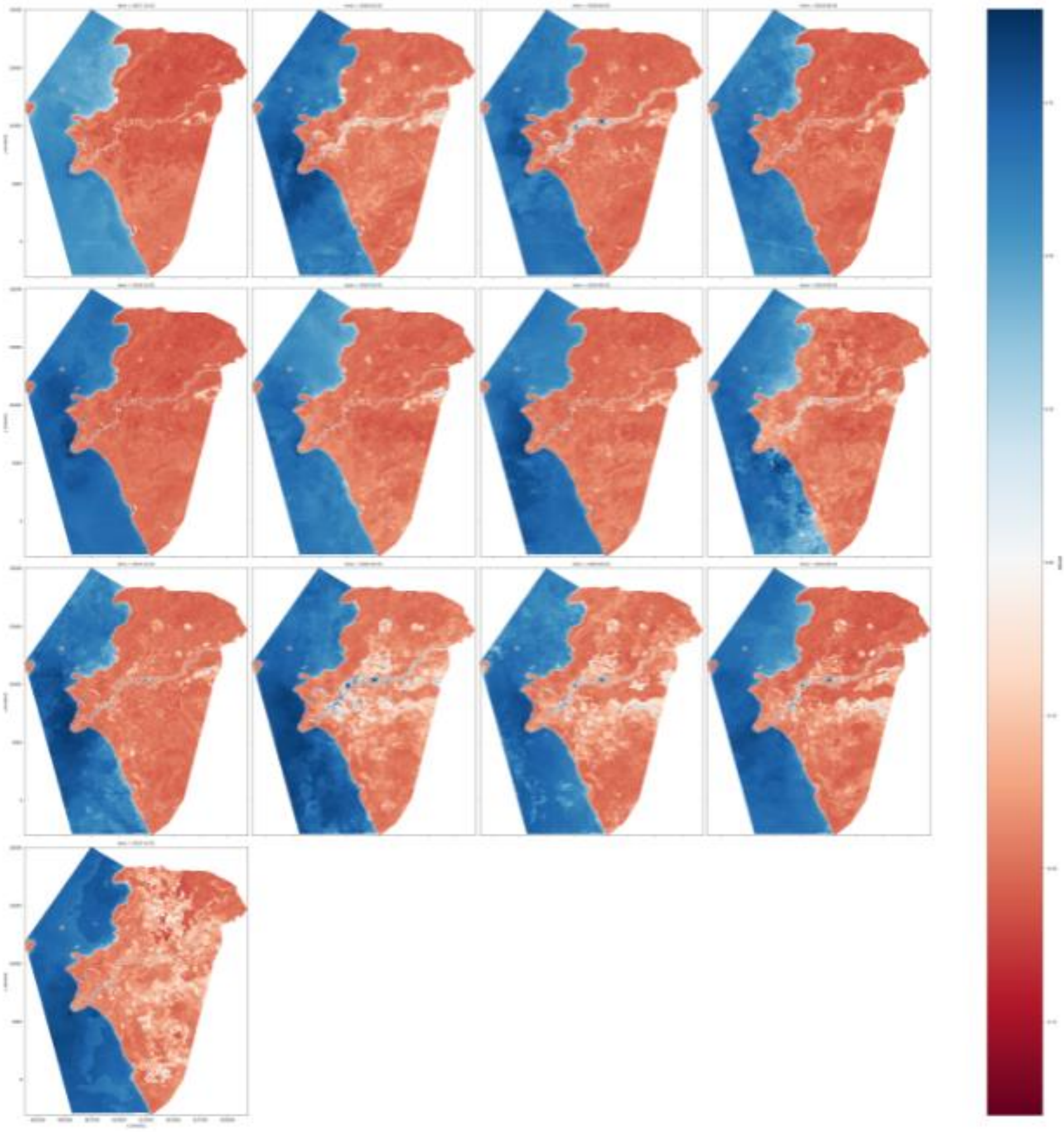
  Resampling the data to seasonal timestamps using medians ensures consistent time-series

```python
sample_frequency="QS-DEC"  # quatrley starting in DEC, i.e. seasonal

#resample MNDWI using medians
print('calculating MNDWI medians')
mndwi = ds['MNDWI'].resample(time=sample_frequency).median().compute()
```

```
calculating MNDWI medians
```

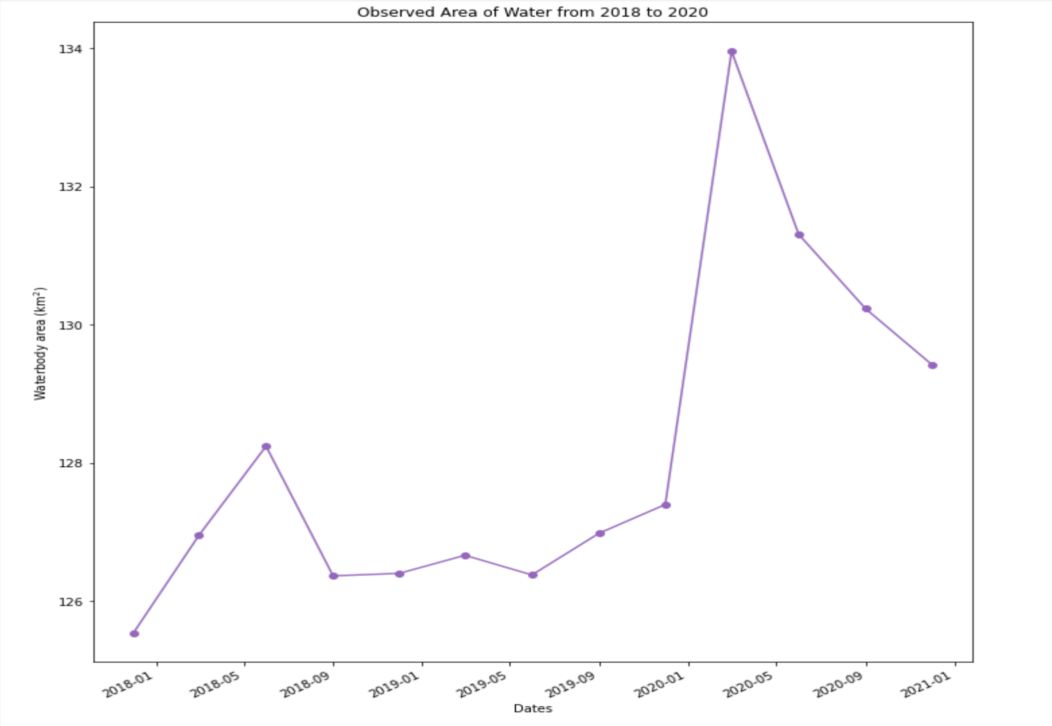
Plotting the median MNDWI timesteps

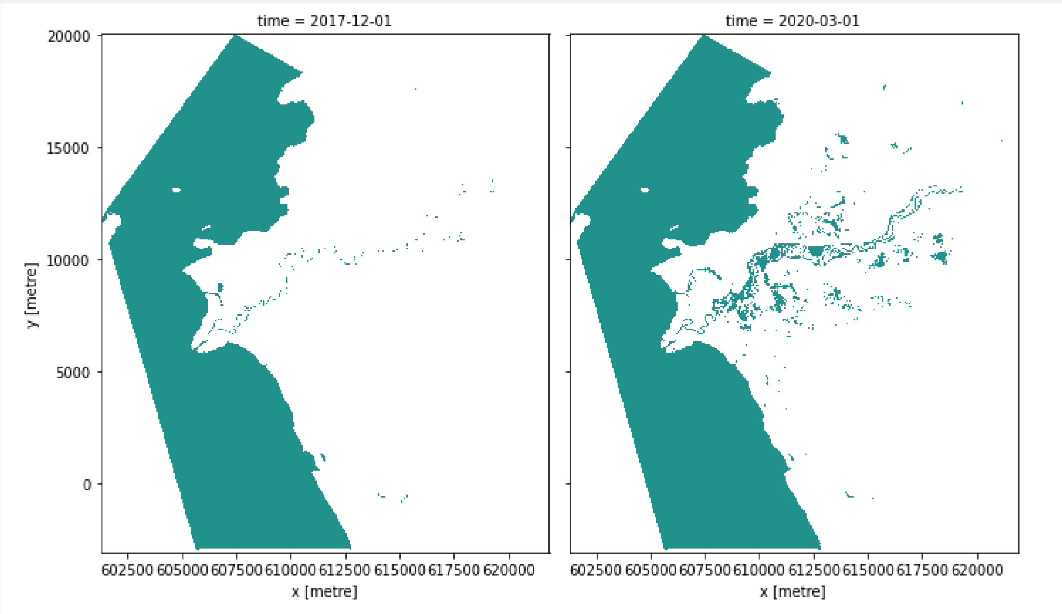
```python
mndwi.plot(col='time', col_wrap=4, cmap='RdBu', size = 10);
```



- **Determining minimum and maximum water extent and calculating the change from the two nominated periods** Time taken to run: aprox 60 seconds

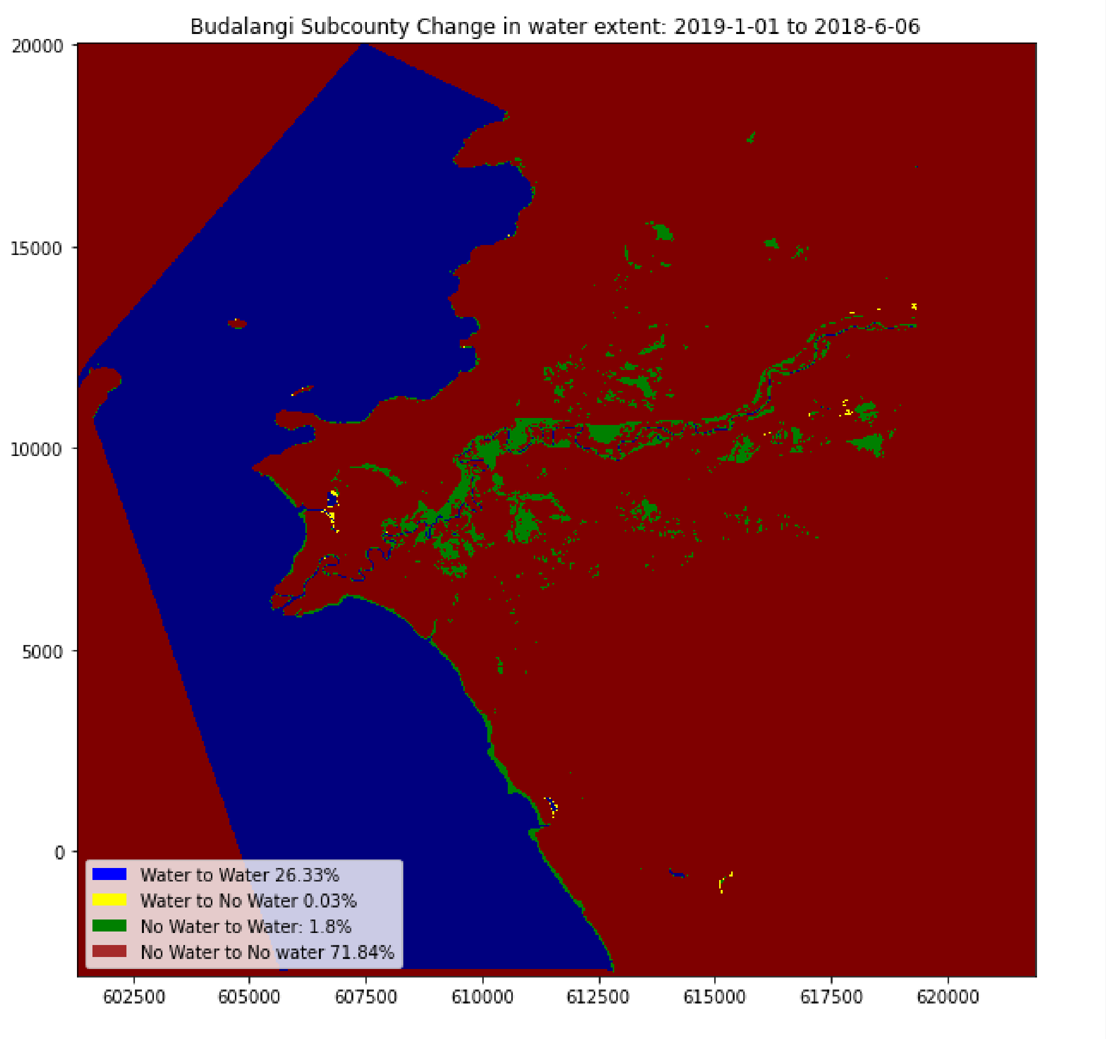
Calculating the water extent per time-step and plotting the time series of open water area exhibited the maximum and the minimum water extent. An average extent was used as baseline time and the maximum extent was used as analysis time.





- **Calculating the Change from the two nominated periods**

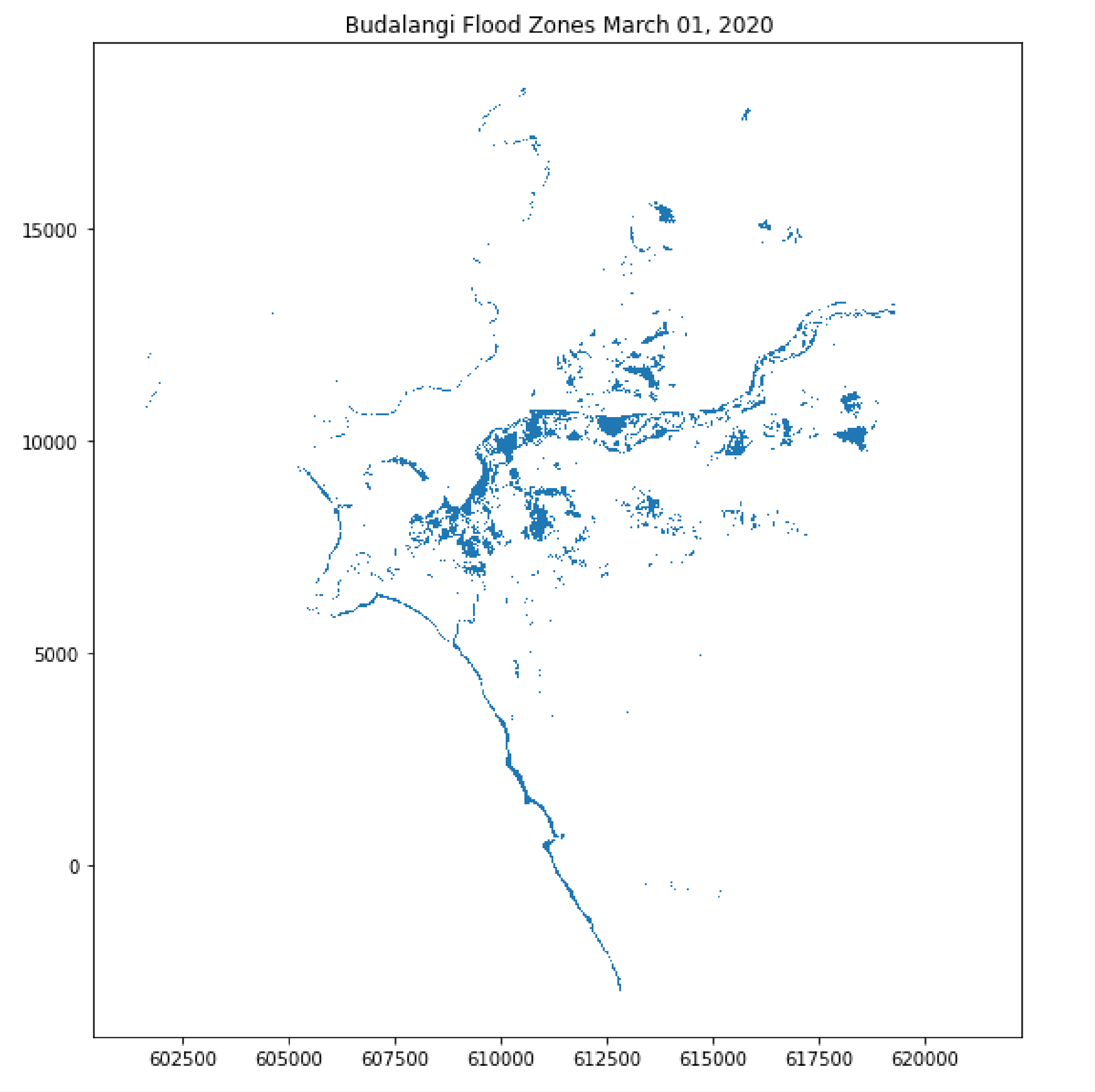
```
# calculating thhr ammount of water gain, loss and no change


# The two period Extract the two periods(Baseline and analysis) dataset from
ds_selected = area_ds.where(area_ds == 1, 0).sel(time=time_xr)

#The dataset array is transform to 1 and 0 using the `astype(int)` function.
analyse_total_value = ds_selected[1]
change = analyse_total_value - ds_selected[0]

#The various scenarios are extracted from the `change` variable for the two years
#Water gain  is 1
water_appeared = change.where(change == 1)

#Permanent water = 0
permanent_water = change.where((change == 0) & (analyse_total_value == 1))

#Permanent Land = 0
permanent_land = change.where((change == 0) & (analyse_total_value == 0))

#Water loss = -1
water_disappeared = change.where(change == -1)
```



- **Extracting and vectorizing the flood prone zone layer** Time taken to run: 5.984306335449219e-05 seconds

For easy geometry operations, the flood zone raster was vectorized using the xr_vectorize in deafrica_tools.spatial.

```
water_appeared.plot.imshow(cmap=ListedColormap([water_appeared_color]), add_colorbar=False, add_labels=False)

flood_zone = xr_vectorize(water_appeared,
                   crs='epsg:32636',

                   mask=water_appeared.values==1)

print(flood_zone.head())
flood_zone.plot( )
```



Step 04: Operations on Open Buildings data

- Importing the building dataset from file

  To cater for errors that might occur in persing the column values of a comma separated values (csv) to a geojson geometry, the buildings data wer exporten to drive in Google earth engine in geojson file format (. GeoJson is a format for encoding a variety of geographic data structures that supports point, linestring, multipoint, multilinestring and multipolygon.
  Working with geojson in python is facilitated by json package

```python
builtUp = gpd.read_file('Budalangi_buildings.geojson')
builtUp.crs
```

```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

- Setting coordinate reference system for analysis

  The downloaded buildings data from Google Earth Engine is referenced to WGS 84 , which has to be projected to a 2D crs , common to the other datasets in this analysis

```python
builtUp.to_crs(epsg=32636, inplace=True)
builtUp.plot(color= 'red',alpha=1, figsize=(15,15))
builtUp.crs
```

```
<Projected CRS: EPSG:32636>
Name: WGS 84 / UTM zone 36N
Axis Info [cartesian]:
- E[east]: Easting (metre)
- N[north]: Northing (metre)
Area of Use:
- name: Between 30°E and 36°E, northern hemisphere between equator and 84°N, onshor
e and offshore. Belarus. Cyprus. Egypt. Ethiopia. Finland. Israel. Jordan. Kenya. L
ebanon. Moldova. Norway. Russian Federation. Saudi Arabia. Sudan. Syria. Turkey. Ug
anda. Ukraine.
- bounds: (30.0, 0.0, 36.0, 84.0)
Coordinate Operation:
- name: UTM zone 36N
- method: Transverse Mercator
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

Step 04: Integrating the Two

- Adding area column to flood zone layer

  To facilitate classification of flood zones in terms of area coverage, a column of area in square kilometer was added to the flood zone layer.

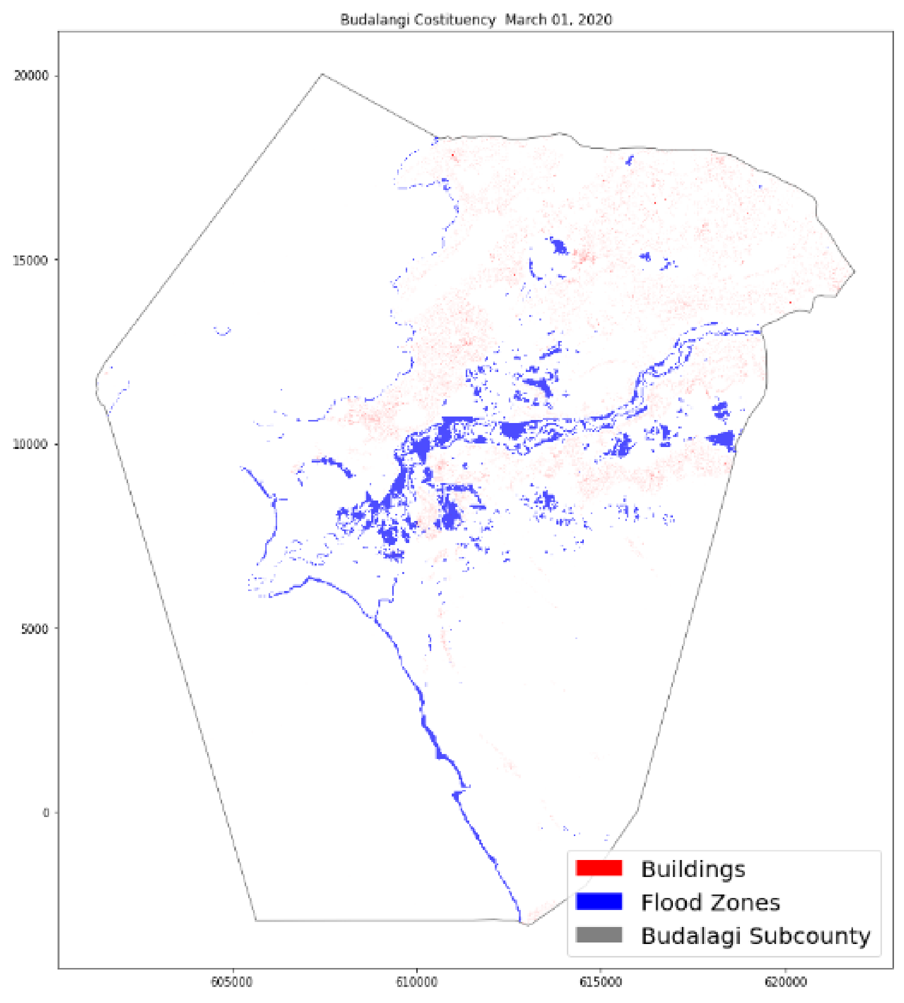
```
flood_zone["area"] =  flood_zone['geometry'].area/ 10**6
```

```
flood_zone
```

	attribute	geometry	area
0	1.0	POLYGON ((610520.000 18320.000, 610560.000 183...	0.0052
1	1.0	POLYGON ((610460.000 18240.000, 610460.000 182...	0.0008
2	1.0	POLYGON ((610560.000 18220.000, 610600.000 182...	0.0040
3	1.0	POLYGON ((610440.000 18160.000, 610460.000 181...	0.0016
4	1.0	POLYGON ((610040.000 17980.000, 610040.000 179...	0.0004
...	...	...	...

- **Plotting the two datasets in same axes**

  To overlay the two datasets, a common axis,ax1 was defined with a figure size of 15*15. The two datasets being of a same region and in same coordinate reference system, overlayed successfully with some building polygons within the flood zone layer. These buildings are of great concern as it facilitates estimation of flood consequences, hence preparedness and response

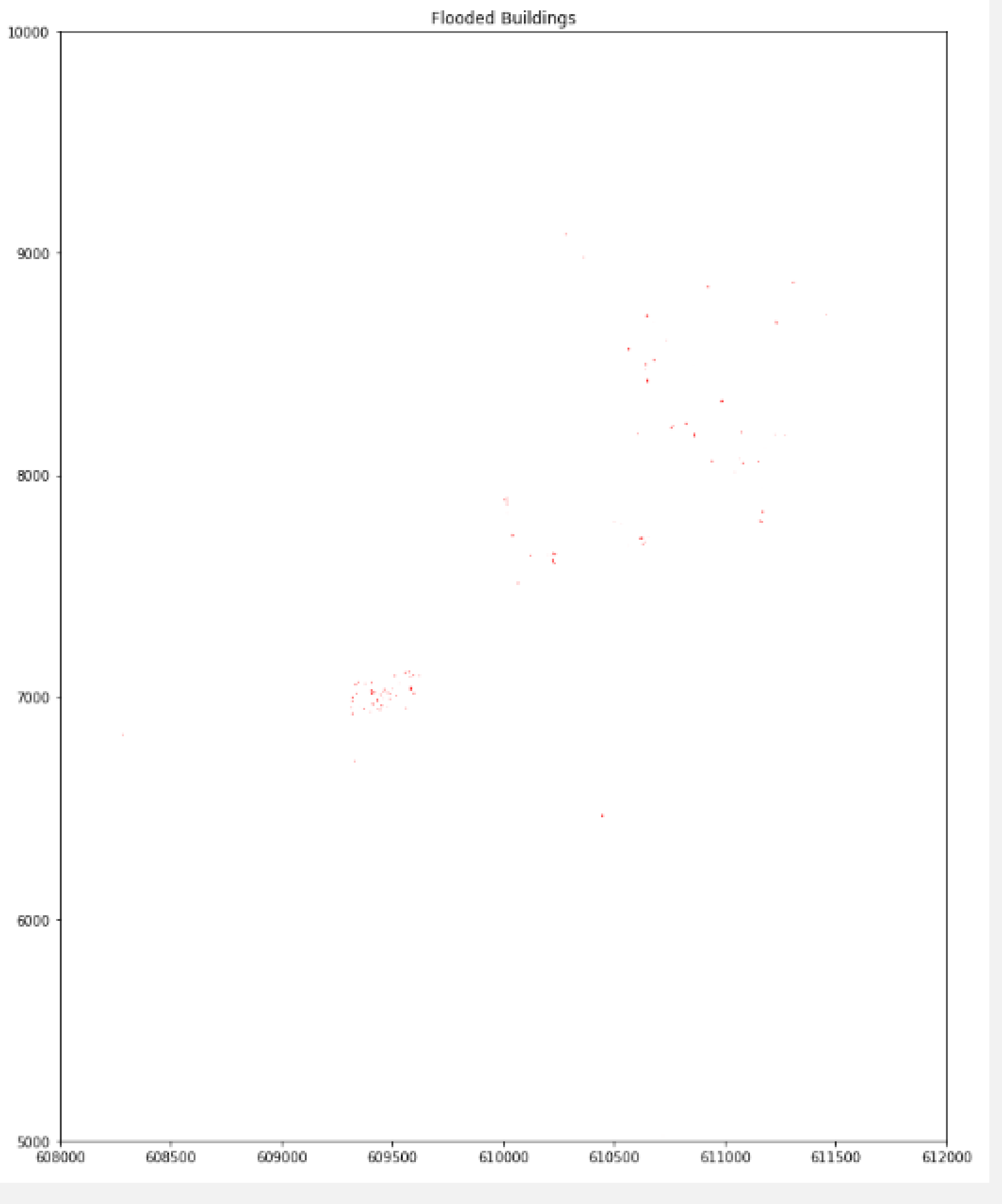


- **Generation of flood affected buildings**

Geopandas.clip clip points, lines or polygon geometries to the mask extent. Both layers must be in the same Coordinate Reference System (CRS). The *gdf* is clipped to the full extent of the clip object.

If there are multiple polygons in mask, data from *gdf* will be clipped to the total boundary of all polygons in mask.

```
affected_builtup = gpd.clip(builtUp, flood_zone)


affected_builtup.plot( color= 'red',alpha=1, figsize=(15,15))
plt.title('Flooded Buildings')
plt.ylim((5000, 10000))
plt.xlim((608000, 612000))
```



Step 06: Exporting and Interactive Communication of the Results *runtime =*

Layers of Flood zone areas, all buildings and flooded buildings were exported to DE Africa Sandbox file as shapefiles (digital vector storage format for storing geographic location and associated attribute information), which were then downloaded to PC storage for interactive mapping

```
flood_zone.to_file("flood_zone_area.shp")

builtUp.to_file("All_buildings.shp")

affected_builtup.to_file("flooded.shp")
```

The interactive communication of the results was done in ESRI's ArcGIS Online.

https://storymaps.arcgis.com/stories/ac7c462a809f4938acfad4f888a3b67d

https://deku.maps.arcgis.com/apps/instant/interactivelegend/index.html?appid=c3678ca4fe2a4115a0722b2a865c9457