Qr code

Description automatically generated

# Vulnerability

A screenshot of a computer

Description automatically generated with medium confidenceThe Intergovernmental Panel on Climate Change defines Vulnerability as “the degree to which a system is susceptible to, or unable to cope with, adverse effects of climate change, including climate variability and extremes.” It is a function of (1) the degree of the system’s Exposure to climate hazards; (2) its Sensitivity to such hazards; and (3) its Adaptive Capacity (IPCC, 2001)

# Adaptive Capacity

These are the layers used for calculating the adaptive capacity.

## Political Category

This layer divides the data into the following categories. The shapefile was derived from the Phil-LiDAR Project of the Department of Science and Technology. Most cities would have more money to provide aid, thus higher adaptive capacity.

5 – City

3 - Municipality

## Regional GDP

The regional gross domestic product was accessed from PSA. The values were then divided into five classes as shown in the layer properties below.

* <https://psa.gov.ph/grdp/grdp-id/167338>

Graphical user interface, application

Description automatically generated

## Provincial Score

This is a score derived from the Department of Trade and Industry. Rankings of Cities and Municipalities are based on the sum of their scores on 4 Pillars: Economic Dynamism, Government Efficiency, Infrastructure, and Resiliency. Provincial rankings are based on population and income weighted average of the Overall scores of cities and municipalities under a province.

* Graphical user interface, application

  Description automatically generated<https://cmci.dti.gov.ph/rankings-data.php?unit=Provinces>

def reclass(x):

if x <= 20.793699 :

return 1

elif x >= 20.793700 and x <= 28.793699 :

return 2

elif x >= 28.793700 and x <= 32.643101 :

return 3

elif x >= 32.643102 and x <= 36.168098 :

return 4

else:

return 5

reclass( !Prov\_Score! )

## Land Cover

The land cover data was retrieved from below:

* <https://data.humdata.org/dataset/philippines-other-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0>
* Graphical user interface, text, application

  Description automatically generated

"DESCRIPT" in ( 'Arable land, crops mainly cereals and sugar' , 'Closed canopy, mature trees covering > 50 percent', 'Coconut plantations', 'Crop land mixed with coconut plantation', 'Crop land mixed with other plantation', 'Cultivated Area mixed with brushland/grassland', 'Grassland, grass covering > 70 percent', 'Mangrove vegetation', 'Marshy area and swamp', 'Mossy forest', 'Open canopy, mature trees covering < 50 percent', 'Other plantations', 'Pine forest')

## Mean Green

Graphical user interface, text

Description automatically generatedThe value was extracted from ESRI RGB green band.

def reclass(x):

if x <= 58.453701 :

return 1

elif x >= 58.453702 and x <= 69.981697 :

return 2

elif x >= 69.981698 and x <= 80.967003:

return 3

elif x >= 80.967004 and x <= 95.461304 :

return 4

else:

return 5

reclass( !Green\_Val!)

## [Change in Global Tree Cover on Agricultural Land – 2000-2010](http://worldagroforestry.org/global-tree-cover/Data/Global_Tree_Cover_Database_Tiff_Format/Tree_Cover_Ag_Land_Diff_2000-2010.zip)

Graphical user interface, application

Description automatically generatedThis data was retrieved from: <http://apps.worldagroforestry.org/global-tree-cover/data-download.html>

def reclass(x):

if x <= -2.793330 :

return 1

elif x >= -2.793329 and x <= 0.785714 :

return 2

elif x >= 0.785715 and x <= 3.898620 :

return 3

elif x >= 3.898621 and x <= 7.75:

return 4

else:

return 5

reclass( !TreeCover! )

## [Change in Global Biomass Carbon on Agricultural Land – 2000-2010](http://worldagroforestry.org/global-tree-cover/Data/Global_Tree_Cover_Database_Tiff_Format/Biomass_Carbon_Ag_Land_Diff_2000-2010.zip)

Graphical user interface, application

Description automatically generatedThis data was retrieved from: <http://apps.worldagroforestry.org/global-tree-cover/data-download.html>

def reclass(x):

if x <= -624.411987 :

return 1

elif x >= -624.411986 and x <= 175 :

return 2

elif x >= 175.000001 and x <= 866.585022 :

return 3

elif x >= 866.585023 and x <= 1716.239990:

return 4

else:

return 5

reclass( !Biomass! )

## Global Human Modification

Graphical user interface, application

Description automatically generatedThe data was retrieved from: <https://sedac.ciesin.columbia.edu/data/set/lulc-human-modification-terrestrial-systems>

def reclass(x):

if x <= 0.245240 :

return 5

elif x >= -0.245241 and x <= 0.330327 :

return 4

elif x >= 0.330328 and x <= 0.412 :

return 3

elif x >= 0.4120001 and x <= 0.516910:

return 2

else:

return 1

reclass( !GHM!)

## Normalized DifferenceVegetation Index

Using google earth engine, calculate for NDVI before and after Odette to estimate the change in cover.

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi)

}

var visPaaramsTrue = {bands: ['B4', 'B3', 'B2'], min: 0, max: 0.3, gamma: 1.4};

var before\_odette = ee.ImageCollection("LANDSAT/LC08/C02/T1\_TOA")

.filterDate('2021-01-01', '2021-12-15')

.filterBounds(roi)

.min();

var after\_odette = ee.ImageCollection("LANDSAT/LC08/C02/T1\_TOA")

.filterDate('2021-12-17', '2022-03-30')

.filterBounds(roi)

.min();

var ndvi\_before\_odette = addNDVI(before\_odette)

var ndvi\_after\_odette = addNDVI(after\_odette)

Map.addLayer(ndvi\_before\_odette, {min:0, max:0.5}, 'NDVI Before Odette')

Map.addLayer(ndvi\_after\_odette, {min:0, max:0.5}, 'NDVI After Odette')

Map.addLayer(before\_odette, visPaaramsTrue, 'Before Odette')

Map.addLayer(after\_odette, visPaaramsTrue, 'After Odette')

Map.centerObject(roi, 8);

var area = ee.FeatureCollection('users/kalyejenner/odette\_affected\_municipalities');

Map.addLayer(area, {}, 'Affected Areas')

Export.image.toDrive({

image: before\_odette.float(),

description: 'BeforeOdette',

scale: 30,

region: roi

});

Export.image.toDrive({

image: after\_odette.float(),

description: 'AfterOdette',

scale: 30,

region: roi

});

# Impact

These are the layers used for calculating the impact. Most of the values on this layer was retrieved from the National Disaster Risk Reduction’s monitoring dashboard.

* https://monitoring-dashboard.ndrrmc.gov.ph/page/situation/situational-report-for-tc-odette-2021

## Casualties

Table

Description automatically generated

### Number of Deaths

Graphical user interface, application

Description automatically generated

def reclass(x):

if x >= 72:

return 5

elif x >= 27 and x <= 71:

return 4

elif x >= 9 and x <= 26:

return 3

elif x >= 2 and x <= 8:

return 2

else:

return 1

reclass( !Death! )

### Number of Injured

Graphical user interface, application

Description automatically generated

def reclass(x):

if x >= 174:

return 5

elif x >= 113 and x <= 173:

return 4

elif x >= 29 and x <= 112:

return 3

elif x >= 3 and x <= 28:

return 2

else:

return 1

reclass( !Injured! )

### Number of Missing People

Graphical user interface

Description automatically generated

def reclass(x):

if x >= 8:

return 5

elif x >= 6 and x <= 7:

return 4

elif x >= 5 and x <= 5:

return 3

elif x >= 1 and x <= 4:

return 2

else:

return 1

reclass( !Missing! )

## Roads and Bridges

Table

Description automatically generated

### Damaged Roads

Graphical user interface, application

Description automatically generated

def reclass(x):

if x >= 56:

return 5

elif x >= 28 and x <= 55:

return 4

elif x >= 24 and x <= 27:

return 3

elif x >= 1 and x <= 23:

return 2

else:

return 1

reclass( !Blk\_Roads! )

### Damaged Bridges

Graphical user interface

Description automatically generated

def reclass(x):

if x >= 7:

return 5

elif x >= 4 and x <= 6:

return 4

elif x >= 2 and x <= 3:

return 3

elif x >= 1 and x <= 1:

return 2

else:

return 1

reclass( !Blk\_Bridge! )

## Communication Lines

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

def reclass(x):

if x >= 78:

return 5

elif x >= 39 and x <= 77:

return 4

elif x >= 8 and x <= 38:

return 3

elif x >= 3 and x <= 7:

return 2

else:

return 1

reclass( !Dam\_Comm! )

## Damaged Houses

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

424622 - 1146217

187711 – 424621

89785 – 187710

7247 – 89784

2 – 7246

def reclass(x):

if x >= 424622 :

return 5

elif x >= 187711 and x <= 424621:

return 4

elif x >= 89785 and x <= 187710:

return 3

elif x >= 7247 and x <= 89784:

return 2

else:

return 1

reclass( !Dam\_House!)

## Damage to Agriculture

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

2529794285.000001 - 6182679072.060000

1785548135.420001 - 2529794285.000000

1284610039.000001 - 1785548135.420000

54784894.790001 - 1284610039.000000

0.000000 - 54784894.790000

def reclass(x):

if x >= 2529794285.000001 :

return 5

elif x >= 1785548135.420001 and x <= 2529794285.000000:

return 4

elif x >= 1284610039.000001 and x <= 1785548135.420000:

return 3

elif x >= 54784894.790001 and x <= 1284610039.000000:

return 2

else:

return 1

reclass( !Dam\_AgriP!)

## Damage to Infrastructure

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

4205649557.920001 - 22812640655.799999

1437594142.180001 - 4205649557.920000

688901000.000001 - 1437594142.180000

173400000.000001 - 688901000.000000

0.000000 - 173400000.000000

def reclass(x):

if x >= 4205649557.920001 :

return 5

elif x >= 1437594142.180001 and x <= 4205649557.920000:

return 4

elif x >= 688901000.000001 and x <= 1437594142.1800000:

return 3

elif x >= 173400000.000001 and x <= 688901000.000000:

return 2

else:

return 1

reclass( !Dam\_InfrP!)

# Calculating for Impact

IMPACT =

[IMP\_DEATH]+ [IMP\_INJURE]+ [IMP\_MISSIN]+ [IMP\_ROAD]+ [IMP\_BRIDGE]+ [IMP\_COMM]+ [IMP\_HOUSE]+ [IMP\_AGRI]+ [IMP\_INFRA]+ [IMP\_PATH]

Table

Description automatically generated

def reclass(x):

if x >= 38:

return 'H'

elif x >= 24 and x <= 37:

return 'M'

else:

return 'L'

reclass( !IMPACT!)

# Calculating for Adaptive Capacity

ADCAP = [ADC\_POLITI]+ [ADC\_GDP\_R]+ [ADC\_PROV]+ [ADC\_GREEN]+ [ADC\_TreeC]+ [ADC\_BioC]+ [ADC\_GHM]+ [ADC\_LCOVER]

Table

Description automatically generated

def reclass(x):

if x >= 30:

return 'H'

elif x >= 19 and x <= 29:

return 'M'

else:

return 'L'

reclass( !ADCAP! )

# Calculating Vulnerability

Table, calendar

Description automatically generated with medium confidence

def reclass(im, ac):

if im == 'L' and ac == 'L' :

return 'M'

elif im == 'L' and ac == 'M':

return 'L'

elif im == 'L' and ac == 'H':

return 'L'

elif im == 'M' and ac == 'L':

return 'H'

elif im == 'M' and ac == 'M':

return 'M'

elif im == 'M' and ac == 'H':

return 'L'

elif im == 'H' and ac == 'L':

return 'H'

elif im == 'H' and ac == 'M':

return 'H'

else:

return 'M'

reclass( !ADCAP\_LMH! , !IMPACT\_LMH! )