NISMOD

Hands-On 6 – Modelling infrastructure exposure and risk

This session uses the road network and flood dataset extracted in the previous session to assess road exposure, vulnerability and risk (as expected annual damages) under historical conditions and future climate scenarios.

Learning outcomes

- Assess direct damage and indirect disruptions to infrastructure assets
- Apply the risk calculation to understand how to generate loss-probability curves
- Show how different flood hazards introduce uncertainty in risk estimations.

Download data package

This exercise continues with the notebooks and data folder that were set up in the previous hands-on session (Hands-On 5).

Download the data package v0.1, unzip the contents and copy them into your working data folder.

The README contains full details of the datasets contained in this package: roads extracted from OpenStreetMap, river flooding extracted from Aqueduct Flood Hazard Maps, Ghana subnational administrative boundaries from the Ghana Statistical Service, country boundaries from Natural Earth Data, and a QGIS project and set of example results which you will be able to reproduce over the next couple of sessions.

Calculate exposure

Open the notebook “02-assess-damage-and-disruption.ipynb”. This contains code to implement each of the steps in this tutorial.

Run the first two cells to import Python libraries and set your data folder to the location of the data on your machine.

The geopandas library provides a lot of the methods that we will use for analysis: it lets us read spatial data files (with the "read_file" method) into Python as a type of variable called “GeoDataFrame”. GeoDataFrames are a tabular format, with columns for each of the data attributes and a column for geometry. They let us do lots of calculations on whole columns at a time, which makes for shorter, simpler code than would otherwise be possible.

Run the next cells to list all the hazard files in the flood layer folder, read in the roads data that we prepared in the previous hands-on session, then run intersections against all hazard scenarios.

The key line of code within the for-loop that actually runs the intersection operation uses the geopandas “overlay” method to find the parts of the line geometries in “roads” which fall within any of the polygon geometries in “flood”:

```
intersections = gpd.overlay(roads, flood, how='intersection')
```

The next cell uses the pyproj library. Pyproj provides methods for working with coordinate reference systems, projecting and reprojecting spatial data, and calculating distances, lengths and areas of spatial data directly from geographic coordinate systems, that is data where the coordinates are in latitude and longitude. A reasonably accurate way of calculating the distance from one point to another over the earth’s surface, remembering that the earth is not flat and not spherical, is to use a "geoid", and calculate the "geodesic length".

Calculate the geodesic length of the intersections, and assign the calculated lengths to the “flood_length_m” column of the dataframe:

```
geod = Geod(ellps='WGS84')
intersections['flood_length_m'] =
intersections.geometry.apply(geod.geometry_length)
```

The for-loop we just ran saved all of the intersections to separate files. Optionally, open some of these files in QGIS to check what was saved.

Next, read in and combine all the intersections into a single dataframe. This lets us then summarise the total length of roads exposed to depth 2m or greater flooding, under all return periods and climate scenarios.

```
summary = all_intersections \
    [all_intersections.depth_m >= 2.0] \
    .groupby(['hazard', 'rcp', 'gcm', 'epoch', 'rp']) \
    .sum() \
    .drop(columns=['depth_m'])
```

Finally plot exposure against return period, with separate plot areas for each Representative Concentration Pathway (RCP), and different colours for the different Global Climate Models (GCM).

```
sns.lmplot('rp', 'flood_length_m', data=summary.reset_index(),
hue='gcm', col='rcp', fit_reg=False)
```

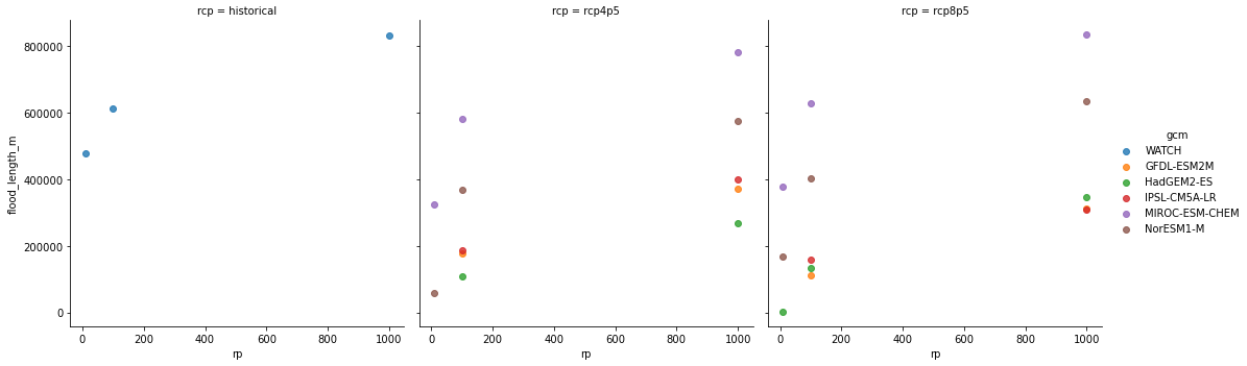

Figure 6.1: Exposure of roads to floods of different return periods, under historical, RCP 4.5 and RCP 8.5 climate scenarios, with different global climate models.

Assess vulnerability to direct damage.

First set up fragility curve assumptions, where probability of damage (pfail) depends on whether a road is paved and the depth of flood it is exposed to.

These assumptions are derived from @Koks2019 Figure S3, extrapolated to 2m and 3m depths.

The analysis is likely to be highly sensitive to these assumptions, and this approach is strongly limited by the availability and quality of fragility data, as well as the assumption that fragility can be related to flood depth alone - flood water velocity would be an important factor in a more detailed vulnerability assessment.

Next set up cost assumptions. These are taken from @Koks2019 again, Table S8, construction costs to be assumed as an estimate of full rehabilitation after flood damage. Again, the analysis is likely to be highly sensitive to these assumptions, which should be replaced by better estimates if available.

Set up assumptions about which roads are paved or unpaved, and number of lanes. Assume all tertiary roads are unpaved, all others are paved. Assume trunk roads and motorways have four lanes, primary and secondary roads have two lanes, and all other roads have a single lane. If complete and consistent data were available, we could skip these assumptions – or if partial data were available, we could validate and refine them.

Discard all information on flood depths greater than 3m to use the fragility curve to estimate the probability of failure for each exposed section.

```
all_intersections_coarse_depth = all_intersections.copy()
all_intersections_coarse_depth.depth_m =
all_intersections_coarse_depth.depth_m.apply(lambda d: str(d) if
d < 3 else ">=3")
```

Finally estimate cost of rehabilitation for each exposed section.

```
all_intersections_coarse_depth['damage_usd'] = \
    all_intersections_coarse_depth.flood_length_m * \
```

```
all_intersections_coarse_depth.cost_usd_per_km \
/ 1000
```

Save the estimates to files and summarise in a table something like the table below (numbers here are given purely as an example):

RCP	GCM	Return period (years)	Exposed length (km)	Direct damage (million USD)
historical	WATCH	10	610	580
		100	1630	1380
		1000	1790	1500
rcp4p5	GFDL-ESM2M	10	80	110
		100	740	660
		1000	1070	1000
	HadGEM2-ES	10	90	100
		100	380	340
		1000	880	800
	IPSL-CM5A-LR	10	80	110
		100	550	450
		1000	1120	1040
	MIROC-ESM-CHEM	10	480	490
		100	910	810
		1000	1400	1240
	NorESM1-M	10	160	170
		100	640	580
		1000	1100	1030

Calculate risk as expected annual damages

Calculate expected annual damages for each road under historical hazard.

Start by selecting only historical intersections, and keeping only the road ID, return period, probability of damage, and cost of rehabilitation if damaged.

```
historical = all_intersections_coarse_depth \
    [all_intersections_coarse_depth.rcp == 'historical'] \
    [['id', 'rp', 'pfail', 'damage_usd']]
```

Calculate the expected damage for each length exposed (under a given return period), then sum up the expected damage for each road, per return period.

```
historical['expected_damage_usd'] = historical.pfail *
historical.damage_usd
historical = historical.groupby(['id', 'rp']) \
    .sum() \
    .drop(columns=['pfail', 'damage_usd']) \
    .reset_index()
```

Pivot the table to create columns for each return period (replace any missing values with zero). Now there is one row per road.

```
historical = historical \
    .pivot(index='id', columns='rp') \
    .replace(float('NaN'), 0)
historical.columns = [f"rp{rp}" for _, rp in historical.columns]
```

Calculate expected annual damages, using the “np.trapz” function, which uses the trapezoidal method to integrate under the expected damage curve over return periods.

```
def expected_annual_damages(row):
    return np.trapz([row.rp1000, row.rp100, row.rp10], x=[0.001,
0.01, 0.1])
historical['ead_usd'] =
historical.apply(expected_annual_damages, axis=1)
```

Assess future risk under climate uncertainty

Calculate expected annual damages under each future scenario (for each global climate model and representative concentration pathway).

This follows the same method as for historical flooding above, with the added variables of climate model (GCM) and representative concentration pathway (RCP).

Calculate the expected damage for each length exposed (under a given return period, GCM and RCP), then sum up the expected damage for each road, per return period.

```
future['expected_damage_usd'] = future.pfail * future.damage_usd
future = future.groupby(['id', 'rp', 'rcp', 'gcm']) \
    .sum() \
    .drop(columns=['pfail', 'damage_usd']) \
    .reset_index()
```

Pivot the table to create columns for each return period - now there is one row per road, per return period, GCM and RCP.

```
future = future \
.pivot(index=['id', 'rcp', 'gcm'], columns='rp') \
.replace(float('NaN'), 0)
future.columns = [f"rp{rp}" for _, rp in future.columns]
```

Calculate expected annual damages, integrating under the expected damage curve over return periods.

```
future['ead_usd'] = future.apply(expected_annual_damages,
axis=1)
```

Summarise total expected annual (direct) damages, showing variation between climate models and representative concentration pathways.

```
summary = future.reset_index() \
.drop(columns=['id', 'rp10', 'rp100', 'rp1000']) \
.groupby(['rcp', 'gcm']) \
.sum()
```

Use seaborn again to plot the total expected annual damages over historical and future climate scenarios:

```
sns.lmplot('rcp', 'ead_usd', data=summary.reset_index(),
hue='gcm', fit_reg=False)
```

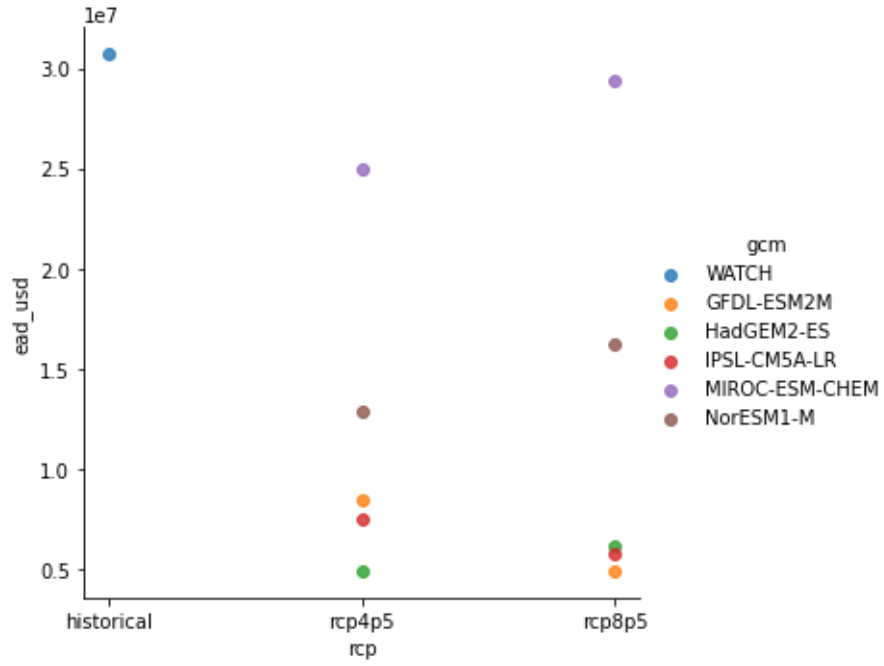

Figure 6.2: Expected Annual Damages (EAD) for roads exposed to river flooding under historical, RCP 4.5 and RCP 8.5 climate scenarios, with different global climate models.

Summary

In the exercises above we used Python in a Jupyter notebook, particularly functionality from the pandas and geopandas libraries, to assess exposure of road network to flooding. We used the geopandas "overlay" method to find the spatial intersection of infrastructure and hazard datasets. We used pandas "concat" method to combine data from multiple files, and the "groupby" method to summarise reasonably large datasets.

Once we had hazard exposure, we made some coarse initial assumptions about the fragility of roads and the costs of rehabilitation to estimate the impact of flooding on exposed roads.

Finally, we estimated the risk from floods of different return periods, integrating under the expected damage curve over return periods. This gave expected annual damages, under historical climate conditions, and under future climate scenarios (RCP 4.5 and RCP 8.5) as modelled by five global climate models (in combination with the Aqueduct hydrological model).