# Task 1: Flood Risk Analysis (Technical Notes)

## Dataset Overview:

These two datasets represent a flood risk analysis performed by Environment Canterbury, Waimakariri District Council, and Selwyn District Council. They show the depth and velocity of a hypothetical flood of the Canterbury plains. This modelling was performed in November 2019. I could not find any key parameters or assumptions for this specific dataset, but flood modelling generally relies on several assumptions. These include uncertainties in flow dynamics, and potential errors in digital elevation models.

## Implementation Efficiency & Optimisation

The key to a fast and efficient implementation is to ensure everything is vectorized. *Rasterio* provides the raster data in a 2D *numpy* array which means that all calculations can be performed with vectorization. The hazard score is the depth multiplied by the velocity. I originally calculated the hazard score (depth \* velocity) separately to the classification, but after struggling with the speed of the classification, I realized that a big slow-down came from the large memory footprint of having three enormous arrays being processed. As such, I calculated the hazard score inside the classification function, which sped up the process significantly. Additionally, I classified the vulnerability using integers (1-6) instead of strings (“H1” – “H6”) which provided a further ~25% time reduction.

## Methodology

The steps I followed to complete this task are as follows:

1. Firstly. I researched Python modules for loading, analyzing, and manipulating raster files.
2. I found both *Rasterio* and *GDAL*. I chose *Rasterio* because of its high-level functionality, efficient implementation, and shallow learning curve.
3. I created a Python environment and script, where I opened and explored both raster files. With help from the *Rasterio* documentation (API reference), I looked at the sizes, units, transforms, and datasets of each file.
4. Once I was familiar with the files, I read section 5.5 of Smith et al (2014).
5. Now that I understood the units and formula of the flooding hazard score, I realized that the files already had the correct units. So, I read the *Rasterio* documentation for resampling the data to 1m resolution, as instructed. I then implemented the resampling in code.
6. After that, I began implementing the flood hazard classification. I went through a few iterations before I settled on a function that I felt was sufficiently fast.
7. From there, I went back to the documentation to figure out how to write new raster files. With that knowledge, I used the flooding hazard score data to write a new raster file to disk.

The only real challenge I came across when completing this task is the optimisation of the vulnerability classification. I had to do some experimentation and research to allow *numpy* to run as quickly as possible. This challenge wasn’t overwhelming in any way, it just required some grunt work.

## Flood Modelling and its Implications

I am using a flood model provided by Environment Canterbury as the basis of my analysis. This means that any inaccuracies in the underlying flood model will trickle down into my analysis. Due to nature of flood modelling, there will almost certainly be some small errors in the data. Unfortunately, the only step I can take to mitigate this is to consider the possibility that the data is inaccurate when performing my analysis. That way, my conclusions include this possibility, and stakeholders will understand that my results do not claim to a perfect forecast of a potential flood.

# Task 2: National Schools Dataset (Technical Notes)

## Datasets

The first dataset I got is from [Education Counts](https://www.educationcounts.govt.nz/directories/list-of-nz-schools). At first, I thought it was missing the decile rating system, but I discovered that it had an Equity Index (EQI) variable which replaced the decile school system in 2023. The main issue with this dataset is that it only provides the SA2 codes, instead of the smaller and more detailed SA1 codes. This is a problem as I need to join 2023 census data (using SA1 codes) with the school’s dataset to extract useful information. Another problem is that a few schools don’t have any SA1 codes or coordinates. I removed these schools from the data as a result.

I have found a good shapefile dataset for the 2023 census. It has totals by topic for individuals, categorized by SA1. I tried for a while to find a way to connect the school’s data (SA2) with the census data (SA1). I eventually realized I could use the 2023 census dataset, projected with EPSG:4326 (latitude and longitude), and then assign SA1 areas to schools based on the coordinates of the school and the polygon of the SA1.

I have chosen to remove any schools without coordinates as I can’t spatially join them. I’ve also chosen to remove three schools with a negative longitude. These are all schools on Chatham Island. I decided to remove them as they are incredibly small, not relevant, and they ruin my data plots. The other main change I had to make to the datasets is handling the -999 values in the census data. I believe this signifies that the sample is small enough that it doesn’t provide any anonymity. I simply replaced these with NaNs.

## Combined Dataset

I spatially joined the two datasets by assigning an SA1 area to a school if the point of the school falls within the polygon of the SA1 area. This resulted in a clean join. The resulting dataset contains 2501 schools, fundamental information about those schools, and relevant statistics for the SA1 areas that each school is in. There are many variables provided by the census data, many of which are not relevant. I chose to include variables pertaining to overall population, age, ethnicity (including a Māori descent indicator), gender, sex at birth, and sexual identity. This broad range of variables allows for many different areas of analysis.

## Database Design

My choice for the database design comes down to disk vs. SQL database. The factors influencing this choice for me are the size of the database, how often I’m querying it, and its permanency. After merging the two datasets, the combined data is only 568.8 KB, which is small enough to simply save it to disk and to load it into memory for querying. Additionally, I don’t have any indication that this data will continue to be queried and analyzed moving forward, I don’t see the need to save it in a database. Therefore, I’ve chosen to keep both raw datasets, and the combined dataset on disk.

## Potential Future Developments

If I were to progress this project further, my main goal would be to improve user experience. Currently, the code itself needs to be changed to see different statistics and different schools. If this was an in-house tool only, I would change the program so statistics can be requested from the command line. If this was a client-facing tool, an entire front-end would need to be developed with UI and UX considerations.