**Introduction**

Our project uses the [Aqueduct Alliance Global Maps 3.0 data](https://www.wri.org/resources/data-sets/aqueduct-global-maps-30-data) as the primordial data source. This dataset aggregates diverse indicators of Water Risk for every water basin in the world. The information it conveys informs us about issues such as water depletion, water quality, and extreme conditions – like drought and flood. The richness of this data piked our curiosity, leading us to wonder what other factors are associated with these characteristics. Is low water quality more prevalent in developing countries? Is it correlated to the human development index of its surroundings? We embraced this challenge as a means to explore these relationships.

We have chosen to use the [global human settlement](https://ghsl.jrc.ec.europa.eu/data.php) dataset as a measure of populational density, the [global subnational infant mortality rates dataset](https://sedac.ciesin.columbia.edu/data/sets/browse?facets=theme:poverty) from the Socioeconomic Data and Applications Center (SEDAC) as a proxy for the Human Development Index (IDH), and the [Nighttime Lights Annual Composites](https://data.noaa.gov/metaview/page?xml=NOAA/NESDIS/NGDC/STP/DMSP/iso/xml/4393.xml&view=getDataView&header=none) from the National Oceanic and Atmospheric Administration (NOAA) as a proxy for economic development – as [recent research](https://www.aeaweb.org/articles?id=10.1257/aer.102.2.994) indicates that night light can be used to measure income growth.

Our main dataset is comprised of vector data. Each basin is a multipolygon, that has diverse features. Each feature is presented in its raw value, a 0 to 5 score, a label (categorical string), and a categorical integer. We chose to work solely with raw values. The full description of the categories can be found [here](https://github.com/wri/aqueduct30_data_download/blob/master/metadata.md). All of our complimentary datasets are raster images.

**Methods – Data Wrangling and Modeling**

The first step we had to take was to combine these 4 datasets into one. We used our water dataset as the baseline. Since it is a vector data (and all other datasets are rasters), we aggregated the mean values of the pixels that fall within the boundaries of each basin using the zonal statistics tool of the rasterstats library. This allowed us to have a single dataset, in which each row represents a basin, having all the original raw water risk values, plus the mean values of child mortality, night light, and population density.

Unfortunately, several lines of this dataset have NA values. We have tried to use the [multivariate feature imputation tool](https://scikit-learn.org/stable/modules/impute.html) from the Scikit-learn library to fill those values, but the results did not look good. Since it is advertised by the developer that this tool is experimental, we chose to remove the NA values from our dataset instead of imputing values to them, thus reducing the number of rows from 68,506 to 42,823.

The combined dataset has 11 water risk features and 3 aggregated features (mean child mortality, mean populational density, and mean night light). Before we performed the cluster analysis, we have subset the water risk features into 3 groups:

* Physical Risk variables: this group comprises variables related to the quality of the water and water systems infrastructure – Untreated connected wastewater (ucw\_raw), Unimproved/no drinking water (udw\_raw), and Unimproved/no sanitation (usa\_raw).
* Water depletion variables: This group holds variables regarding the scarcity or future scarcity of water – Baseline water stress (bws\_raw), Baseline water depletion (bwd\_raw), Interannual variability (iav\_raw), Seasonal variability (sev\_raw), and Groundwater table decline (gtd\_raw).
* Extreme conditions variables: This group has variables related to flood and drought – Riverine flood risk (rfr\_raw), Coastal flood risk (cfr\_raw), and Drought risk (drr\_raw).

We have used the standard scaler in a copy of our dataset to run the Kmeans clustering algorithm. To determine the optimal number of clusters in our Kmeans analysis, we used a for loop that ran the algorithm in a range from 1 to 20 clusters, appending the inertia (the sum of the squared distances to clusters) values for each number of clusters. We plotted then the “elbow function” for each of our variable groups, to determine the optimal number of clusters (the optimal value should significantly reduce the inertia in relation to its previous value, whereas the subsequent value improves it only marginally). We have used then 10 clusters for every group of variables. This allowed us to specify to which cluster each individual observation in the dataset belongs.

We have learned that Kmeans cluster analysis is not an optimal tool to handle this kind of data. However, we hypothesized that the clusters generated by the Kmeans analysis could be used as an input in a DBScan cluster analysis.

To perform the DBScan cluster analysis, the first step we took was to convert the multipolygons into points. We did that by converting the geometry of the observations in the dataset into the centroids of these shapes. We have also converted the dataset into the 6933 EPSG, to get units in meters. Since we wanted the labels from the Kmeans analysis (which were added to the untransformed dataset), we used the standardscaler tool again, transforming the raw variables, the x and y coordinates, and the Kmeans labels. We used this dataset as an input in the DBScan cluster analysis.

**The visualization web app**

**Conclusions and final remarks**