Project 3: Aviation Accidents

Our hand-in contains 2 python-scripts and 1 csv-file:

* *webscrap\_parsing.py*
* *aviation\_accidents.csv*
* *dash\_board.py*

# Main features of webscrap\_parsing.py

The relevant approximately 130 yearly Wikipedia overviews and 1.600-1.700 underlying pages covering specific accidents are not downloaded locally but parsed directly on Wikipedia, storing locally only the obviously relevant data-extract. A full parsing in this manner takes approximately 30 min.

Most Wikipedia pages on or close to the subject have a specific type of info box. We have chosen only to proceed with data from such pages, which have an info box. Furthermore we have chosen to only include info boxes with a “Date” category to ensure we only scrape relevant info boxes.

Accidents where latitude and longitude is not readily available are geolocated using ”Nominatim” geocoder.

Categories like ‘Survivors’, ‘Fatalities’ and ‘Injuries’ often vary on Wikipedia, which sometimes resulted in death and injury rates over 100% or division by zero. This we had to take into account and make some additional coding to extract a meaningful death and injury rate.

# Main features of dash\_board.py

* The requested wordcloud is basically static for each run of the script
* Hovering dots are colored according to number of fatalities (but color legend was deliberately suppressed)
* Pie-chart is only showing the “fate” of the people onboard, generally showing % who died, were injured or who came out unhurt. Variations in data-heading can in a number of cases lead to misleading results. We consider that a pie chart is only relevant where numerical data shows “parts of a whole” and therefore only visualized this data in a pie.
* When opening the dashboard a sample pie with 3 equal slices is shown. The pie-chart will update when clicking a row in the table next to it. However we have not implemented an pie-update from clicking a dot on the map.
* We have some unresolved issues when either the table of accidents is split in several shorter sections and/or when filtering is applied. The pie will keep referring to the non-filtered version of the table which leads to a mismatch.
* Using filtering on alphanumerical columns can be done with a partial string, for instance ‘Lufthansa’ as operator, however, numerical columns like year will need a mathematical operator, for instance ‘=2000’, so ‘2000’ will not work.
* The Wikipedia pages contain a ‘Date’ and we are asked specifically in the assignment to provide a date-column. However, it seems that only a limited part of the dates are analytically understood and interpreted by pandas, so to get a homogenous structure we have chosen only to include ‘Year’ in our final table.

Further possible improvements and enhancements will be addressed in our oral exam slideshow.