Universität Konstanz

Introduction to Computation for the Social Sciences

Assignment 6

Prof. Dr. Karsten Donnay, Marius Giebenhain, Stefan Scholz
Winter Term 2019 / 2020

Please solve the exercises below and commit your solutions to our GitHub Classroom until Dec, 10th midnight. Submit all your code in one executable file (*py* / *ipynb*) and your text in one text file (*txt* / *md* / *pdf*). You can score up to 10 points in this assignment. You will get individual feedback in your repository.

Exercise 1: Data Import (2 Points)

In practical applications you will often be required to import tabular data into Python that was provided to you as .csv files. Together with this assignment we have included the file *ACLED_South-Sudan_2017.csv*. The file codes 1200 conflict events in South-Sudan during the year 2017 coded by the *ACLED*[1] project an initiative that codes conflict events around the world.

Write a simple function `import_csv()` that parses the data into a nested list where the first index corresponds to each row, i.e., each data entry, and the second index to columns, i.e., different variables coded for each event.

Note: You may keep the first line with the column labels in the parsed data or separate it out into an index.

Exercise 2: Text Pre-Processing (3 Points)

The event data imported in *Exercise 1* mostly provides already fully coded categories. In addition, there is a free-text field called *notes* giving a detailed description of the event that was coded. In this exercise, we will prepare this text field for automated analysis performing standard pre-processing steps, such as separating the text into words, removing so called stop words, i.e., words that contain little to no semantic meaning, and normalizing the texts, e.g., by removing punctuation and capitalization.

a) Write a function called `normalize_text()` that converts all text to lower case and removes all punctuations within each field *notes*.
b) Write a function called `remove_stopwords()` that eliminates all stop words that are part of the list of English stop words (see *eng_stop_words.txt*)
c) Write a function `tokenize_text()` that tokenizes, i.e., splits the cleaned text into words.

Exercise 3: Simple Text-Based Analysis (5 Points)

The text in the field *notes* we pre-processed in *Exercise 2* contains additional information that was not routinely coded into the event categories that ACLED provides. The events coded in the database we are using rely on a combination of media-based reporting and local reports from the ground. The text in the field *notes* provides subtle indications, for example, for the degree of certainty that coders had in compiling each record or additional details about how an event occurred. We will here use regular expressions to extract specific records based on this information

a) Run the functions developed in the previous exercise to prepare the field *notes* for automated analysis. Note that it may be favorable here not to clean the original data entries directly but instead add a new column used only for processed text.

b) We are first interested in all events that appear to be coded based on information that apparently came from third party reporting. Write a function `extract_uncertain()` using a regular expression for filtering that extracts all records containing the word *reportedly*. Use the module *re* to write the necessary regular expression and extract records.

c) The character of each event is coded through the event categories that ACLED codes but additional information on the exact nature of the event could still be contained in the "notes" field. Write a function `extract_ambush()` using regular expressions to extract all records containing any mention of *ambush*. Use the module *re* to write the necessary regular expression and extract records.
Note: We suggest to use an "open-ended" regular expression to capture not only the noun but also any usage of the verb "ambush" regardless of conjugation.

d) Within the two sets of extracted records we would like to now examine the field *notes* more closely. For each set of records you extracted, find the 10 most frequent words used in the event description across all entries. Do you see any regularity among the entries based on this analysis?

[1] https://www.acleddata.com/