Course:	Advanced Methods for Geospatial Analysis
Semester:	Summer Semester 2019
Supervisors:	Prof. Dr.-Ing. Martin Kada
	Valentina Schimdt
Entry Deadline:	16.04.2019, 12:00h

09 – 4 – 2019

Topic : Introduction to Pandas

Pandas is a Python library for data analysis. It offers a number of data exploration, cleaning and transformation operations that are critical in working with data in Python. Pandas build upon **numpy** and **scipy** providing easy-to-use data structures and data manipulation functions with integrated indexing.

The main data structures *pandas* provides are 'Series' and 'DataFrames'.

The goal of this exercise is to explore some of the basic features of *'pandas'* library, using a given dataset.

Additional Recommended Resources:

- *pandas* Documentation: https://pandas.pydata.org/pandas-docs/stable/

Dataset :

You will use for this exercise the Sunspots dataset (posted on ISIS, which can be downloaded also from http://www.sidc.be/silso/infossntotdaily):

The data is stored in a .csv file, without column headers. The six columns contain the following information:

- Columns 0-2: Gregorian date ('year', 'month', 'day')
- Column 3: Date as fraction as year ('dec_date')
- Column 4: Daily total sunspot number ('sunspots')
    - Missing values in column 4: indicated by -1
- Column 5: Definitive/provisional indicator (1 or 0) ('definite')
-

Preliminaries: Please, download the file '*ISSN_D_tot.csv*' in a directory created for this exercise and open a new jupyter notebook at the same location.

Tasks :

1. Import *pand*as library
2. Read the *(.csv)* file in a *pandas data*frame using the pandas function *df.read_cvs(),* providing appropriate values for some relevant arguments, as follows:
    - the location of the .csv file
    - correct value for the *header* argument
    - a list with the names of columns (provided in the introductory part) as strings for the *col_names* argument
    - '-1' for *na_values* argument (use *na_values={'sunspots':[' -1']}*

- Also, provide for the keyword argument *parse_dates* a list with the columns containing date information (*parse_dates=[[0, 1, 2]]*)

3. Inspect the data using the methods *head()* and *tail().*
4. Get an overview of the table by calling the method *info()*
5. Get a statistics summary of the data with the method *describe().* In which form is the output?
6. Set the new column ‘year_mont_day’ as index of the dataframe and set the name of the index column ‘name’. Hint: df.index, df.index.name
7. Use again the info() method on your dateframe to get an updates overview of the table
8. Define a list cols = ['sunspots', 'definite'] and use it to subset your dataframe in a new one with the same name.
9. Get a view of the rows between 10 and 20, and all columns (Hint: *df.iloc())*
10. Obtain minimum values for the whole table (Hint: *df.min()*) and only for a column of your choice.
11. Add a new variable of your choice to the *dataframe* and set a value for all the records. E.g., a new variable ‘ones’ containing all ones.
12. Modify the value of ‘ones’ variable for a particular record,
13. Delete the column ‘ones’ (Hint: you may use *del(), drop(), pop()*). Which is the difference in the output?
14. Extract those dates/ records where the number of sunspots is larger than 25.
15. Find out the dates where number of sunspots is less than the average.
16. Write the new dataframe to a *.csv* file. Hint: *out_csv = 'sunspots.csv'*, *df.to_cvs (out_cvs)*