DAT565/DIT407
Introduction to Data Science and AI

Assignment 2
Deadline: 2024-11-18 12:00

In this assignment, we practice *data cleaning* by extracting the interesting data from a source that contains the data, not quite in the format we want. In the second part, we then produce some plots about the data.

Problem 1: Scraping house prices

The file `kungalv_slutpriser.tar.gz` contains the closing prices of houses, sold in Kungälv municipality between 2013 and 2023. The data has been scraped from Hemnet[1] in October 2023. Your task will be to convert the data into a form that can be analyzed.

The tarball contains a number of HTML files. Use Beautiful Soup[2] to parse the HTML and extract the relevant pieces of information for each advertisement of a closing price.

Consider the example of Figure 1. The advertisement contains the following information:

- Date of sale *Såld 28 oktober 2023*,
- Address (or sometimes the name of the plot of land) *Strömgatan 4*,
- Location of the estate *Ytterby, Kungälvs kommun*,
- Area in the form of *boarea+biarea 105+10* m^2,
- The number of rooms *5 rum*,
- Area of the plot *972* m^2 *tomt*, and
- Closing price *5 750 000 kr*.

When discussing the area of houses in Sweden, the number most people are interested in is *boarea*, the area of the house that is classified as habitable. The other part of the area, the *biarea*, relates to areas that are not considered habitable, such as warehouses, garages, and such. In total, they make *totalarea*. As some of the advertisements fail to include all of the pieces of information, assume that we always have information about *boarea*, and compute *totalarea* only if you have both *boarea* and *biarea* in the advertisement. Otherwise leave the field blank.

Use Beautiful Soup to extract all of the pieces of information described above, when applicable. Take into account that there may be discrepancies with how the information is presented, and some information will be missing. Do not impute missing values, just leave them missing.

As output, produce a CSV file that contains the information described above for each entry in the dataset.

[1] `https://www.hemnet.se/salda/bostader?location_ids%5B%5D=17973&item_types%5B%5D=villa`

[2] `https://www.crummy.com/software/BeautifulSoup/`

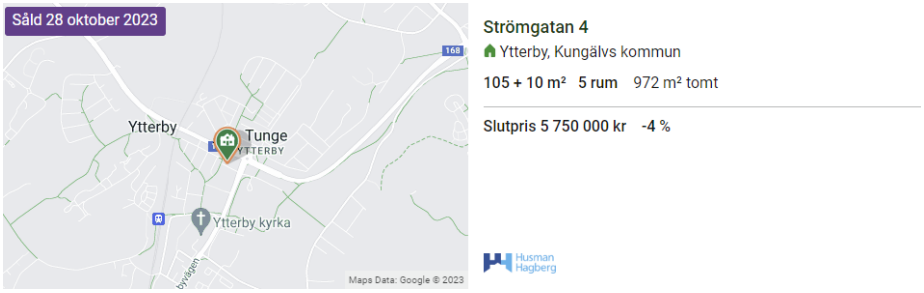

Figure 1: Example announcement of a closing price.

Problem 2: Analyzing 2022 house sales

We will now analyze the data, and produce some plots. Select all houses that were sold in 2022, and answer all points below. Include the plots you create in your report and the answers to any other questions.

- Compute the five-number summary[3] of the closing prices (minimum, maximum, median, first and third quartile).
- Construct a histogram of the closing prices. In your report, describe how you selected the number of bins (or equivalently the bin width).
- Construct a scatter plot that shows the relationship of the closing price with the *boarea* of the house.
- Finally, repeat the scatter plot, but instead of showing all observations in the same color, colorize the observations by the number of rooms in the house.
- Discuss shortly what the figures tell about the prices and the relationship between the different variables.

Hints

- Please get acquainted with the documentation of Beautiful soup: `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`.
- There are also useful tutorials on the Web that show how to do the most common operations in Beautiful Soup; specifically, you will need to select elements by suitable attributes.
- Regular expressions are a very powerful tool for matching strings that follow a certain pattern: `https://docs.python.org/3/library/re.html`.
- Dates can be manipulated using NumPy's `datetime64` objects (see NumPy documentation and the documentation of `pd.to_datetime`).

[3] `https://en.wikipedia.org/wiki/Five-number_summary`

- It may be useful to view side-by-side how the web page is rendered in a web browser and try to identify the HTML elements in the HTML source code; this tells you what to look for in Beautiful Soup.
- The method `findAll(text=True)` can be used to find the textual content of elements.
- If you need to find an element by some specific textual content, you can combine `find` and `findAll` with regular expressions, such as `find(text = re.compile(`*regex*`))`.
- HTML documents form a tree of elements, you can navigate between the parents and children using the attribute `parent` and the method `findChildren` (among others).
- Sometimes it is easier to locate a child of an element and then navigate to its parent (if the child is uniquely identifiable but the parent is not).
- The three preceding points can be particularly useful for determining the *location*.
- The *non-breaking spaces* or ` ` characters can cause trouble; they correspond to the unicode codepoint U+00A0, and as such, they can be replaced with ordinary spaces with `.replace('\u00a0',' ')`.
- It might be worth it to strip some unnecessary whitespace from the start or end or within string; see how the string functions `strip`, `lstrip`, and `rstrip` work, and you can also use regex to replace multiple whitespace with just one regular space (e.g., replace `\s+` with a space).
- The data may be inconsistently formatted and sometimes some values are simply missing (e.g., *boarea* may be present but the number of rooms not).
- Remember to scale the figures appropriately.
- *Always* label axes, including units.
- Since there is a natural order in the number of rooms, you should choose appropriate sequential color scheme.
- Store figures in vector format (PDF files) and embed them in your report in a `figure` environment. Refer to the figures (preferably with `\ref`) in your text when discussing them.

Returning your report

Write a report, typeset in LaTeX, that answers *all* questions above. Include all your Python code in your report as an appendix, preferably using the `listings` package. Your report should be legible even without having a look at your code.

If you refer to outside sources, remember to add an appropriate literature reference (including websites) in references by `\cite`ing the references. It is recommended that you use the package `biblatex` to manage citations.

Place your figures in numbered `figure` environments, with descriptive captions and `\ref` to the figures in your discussion. Likewise, place your tables in

numbered `table` environments with desciprive captions and \`ref` to the tables in your discussion.

After grading, you will be given another attempt to revise your report according to TA comments if it is not considered acceptable.

The deadline is *hard*. Late submissions will not be read at all and are considered failed. This means you will not get any feedback for the first round and the submission is considered a revision; there will be no third attempt, so if a late submission is failed, you will need to participate in a later iteration of the course for a re-attempt.