**Graphical representation of scraped data**:

*Pre-requisites*: Following python libraries are required: **json, numpy, matplotlib.pyplot, seaborn** and **mplcursors**

1. The json data that is generated by data crawling is first opened into a *TextIOWrapper* which is then fed to ***json.load()*** to convert it into a list called *json\_data*.

**Task 1**: *A histogram of argument lengths*

1. The length of arguments is calculated by the **number of tokens**.
2. A function ***Argument\_Length()*** is called with the above list to calculate the length of all arguments as the number of tokens irrespective of topics or categories. This function iterates over all the arguments one by one.
3. Next, another method called ***tokenization()*** is called with each sentenceto calculate the number of tokens by a white space and returns the number of tokens of that sentence back to ***Argument\_Length()***
4. Finally, a list *token\_size\_lst* is returned which contains the length of each sentence present in json file after performing above steps iteratively.

**Task 2**: *Bar graph representing pro/con arguments per category*

1. A grouped bar chart is generated to distinguish between number of pro arguments and con arguments per category.
2. A function ***Category\_Title()*** is called with the list *json\_data* to count the number of pro and con arguments.
3. Before adding a category to a new list, a condition to check for identical category is employed. Therefore, only distinct categories that are present in *json\_data* are added to this list.
4. Pro and con arguments are added to the separate lists by iterating over distinct categories. Later, the number of pro and con arguments of the same category that is already present in the list are added accordingly in their respective lists.
5. Finally, three different lists that contain categories, pro and con arguments separated based on categories are returned.

**Task 3**: *Bar graph representing pro/con arguments per title*

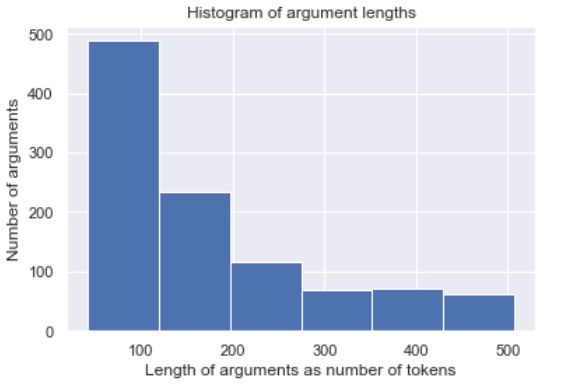
1. A grouped bar chart is generated to distinguish between number of pro arguments and con arguments per title.
2. A function ***Category\_Title()*** is called with the list *json\_data* to count the number of pro and con arguments.
3. The title of each topic, pro and con arguments of the same topic are added to three separate lists iteratively by performing dictionary operations over *json\_data.*
4. Finally three different lists that contain titles, pro and con arguments separated based on titles are returned.

**Evaluation**:

1. A histogram of argument lengths is plotted using *matplot.pyplot* using ***hist()*** function.

*Representation of axes*:

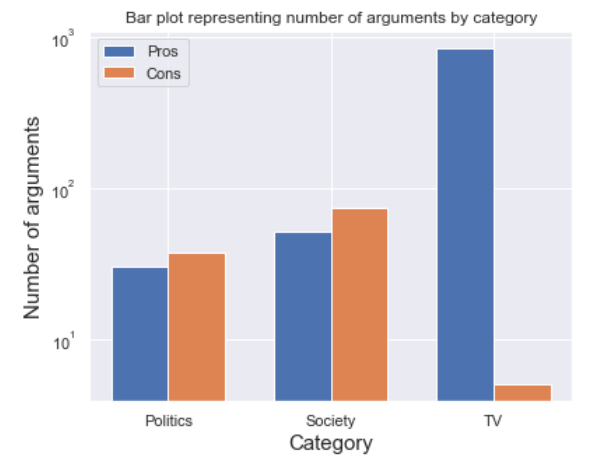
* **x-axis**: Length of arguments as number of tokens
* **y-axis**: Number of arguments



1. Grouped bar graphs that represent number of arguments per category and title are plotted using *matplotlib.pyplot* using ***bar()*** function. As one of the titles has a large number of pro arguments, the y-axis scale has been made *logarithmic* for better view and understanding. A legend for each of the graphs is created to represent pro arguments (blue) and con arguments (orange).

*Representation of axes (per category)*:

* **x-axis**: Category
* **y-axis**: Number of arguments



*Representation of axes (per title)*:

* **x-axis**: Title
* **y-axis**: Number of arguments

