00 Configuration Manual - README

August 15, 2022

1 Configuration Manual - README

Together but not mixed: Dynamic and social networks structures in the Constitutional Convention in Chile 2021-2022

Juan Pablo Palma B. (x20225776)

2 0. Previous considerations

This project had developed in four steps. First, it was thought to be run in any platform compatible with JupyterNotebook formats. I chose this format to make the code easier to follow, combined with a story-telling form to explain the step-by-step project.

1. Data Collection
2. Data Wrangling
3. Exploratory and Descriptive Analysis
4. Network Analysis

Each stage has its own `.ipynb` document and addresses the requirements in the next step of this document. All the scripts should run with no issues.

3 1. Getting Started

To replicate the analysis process for this project properly, I have included a `requirements.txt` to assure that it is using the same version of libraries, avoiding potential errors and also allowing to install of the requirements even in virtual environments. To use this just run the next line:

[ ]:

```
pip install -r requirements.txt
```

4 2. Libraries used and brief description

4.1 Beautiful Soup

Beautiful Soup is a Python library for pulling data out of HTML and XML files.

Documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

4.2 Requests

Requests is a simple, yet elegant, HTTP library. Allows you to send HTTP/1.1 requests extremely easily. There's no need to manually add query strings to your URLs, or to form-encode your PUT & POST data — but nowadays, just use the json method!

Documentation: https://pypi.org/project/requests/

4.3 Pandas

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Documentation: https://pandas.pydata.org/docs/pandas.pdf

4.4 Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Documentation: https://numpy.org/doc/stable/numpy-user.pdf

4.5 Fuzzywuzzy

Fuzzy string matching like a boss. It uses Levenshtein Distance to calculate the differences between sequences in a simple-to-use package.

Documentation: https://pypi.org/project/fuzzywuzzy/

4.6 Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Documentation/Tutorial: https://seaborn.pydata.org/tutorial.html

4.7 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Documentation: https://matplotlib.org/stable/index.html

4.8 Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Documentation/User Guide: https://scikit-learn.org/stable/user_guide.html

4.9 NetworkX

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

Documentation: https://networkx.org/documentation/stable/_downloads/networkx_reference.pdf

Additionally, during the uses of `NetworkX` is possible to get an error which is solved using the version of decorator.

```
decorator==5.0.9
```

4.10 Decorator

A decorator is a design pattern in Python that allows a user to add new functionality to an existing object without modifying its structure. Decorators are usually called before the definition of a function you want to decorate.

Documentation: https://python101.pythonlibrary.org/chapter25_decorators.html

(c) Juan Pablo Palma B. (2022)