‘CleanlockHolmes’, Unraveling Data Mysteries with Python

**Rebecca Bronfeld, Sarah Gore, Kieran Black**

<https://github.com/kierblk/NU-DS5010-Project>

**Summary:**

**Project Purpose:**

In the field of Data Science, data quality is paramount. A clean and well-prepared dataset is essential for accurate and reliable analysis, modeling, and decision-making. Issues can arrive when data is incorrect, has missing values or is of an incorrect format or data-type. CleanlockHolmes detects and removes missing, erroneous, or inconsistent values, which can compromise the reliability and accuracy of any data-driven project.

**Related Work**

Identify similar libraries such as pandas – using pandas to help with underlying data frame not utilizing their own built in cleaning functionalities

**REBECCA TO DO**

**Structure Overview**

CleanlockHolmes is a single class developed in Python that allows users to define and customize valid or invalid data with a single raw data set and then correct any problematic data in one of three prespecified ways. CleanlockHolmes does this by creating an instance of the data wrapped in the object and allows the user to interact with the package in a number of methods that identify what is considered “invalid” or “valid” data.

**Design:** A technical description of the modules, classes, and functions implemented in the package. This does not need to be exhaustive, and should only cover the most important aspects of the package for a user to understand. The description should be clear and concise so that a user can understand the overall design and organization of the package.

CleanlockHolmes utilizes a one to one mapping between a dataset and an object. This means that the user can only clean a single data set at a time and would need to create additional instances of CleanlockHolmes if they have multiple data sets that need cleaning. This is conducted through the \_\_init\_\_ function which instantiates the CleanlockHolmes object in Python. A CleanlockHolmes object consists of a dataframe that is imported using the Pandas package. Pandas allows large data sets to be imported efficiently.

*Design Decision 1*: Which data formats will the package accept?

CleanlockHolmes accepts csv and json data types as these are both innately supported by the Pandas package and can be simply imported using CleanlockHolmes’ **read\_data** function.

*Design Decision 2 & 3*: What constitutes an invalid or valid observation? Is it predefined or user defined?

Invalid or valid data is dependent on the data set and can vary for each individual variable in the data set. Therefore, CleanlockHolmes provides flexibility for the user to make these distinctions at the variable level. For example, numeric variables will be treated differently than categorical variables as the user may want to handle identification in a different manner. CleanlockHolmes has three methods of defining invalid values at the variable level. The first is defining a valid range in the *data\_ranges* dictionary for numerical variables (int and float types) by the user defining lower and upper bounds. This can be done with user prompts through the **interactive\_specify\_viable\_range** function or as inputs in the **specify\_viable\_range** function. In addition to specifying a valid range, CleanlockHolmes will also test whether an observation is not a number and mark it as invalid if true.

The second and third methods are used for categorical variables (str type). The CleanlockHolmes object includes two dictionaries that the user will populate – one for valid values and one for invalid values (*valid\_dictionary* and *invalid\_dictionary*, respectively). When instantiated these dictionaries are empty, but they are populated by the **specify\_valid\_entries** and **specify\_invalid\_entries** functions.

For any particular variable ne of the two modes will be more efficient

Eg :

1. decide what a blank value looks like - is this something that the user gets to tell us
2. \*\* NA , null, blank, empty strings, dashes (--)
3. How is problematic data corrected

Note design choices that were made and justify why

Explain that backbone of class is cleanloclholmes object. One to one mapping between a dataset and an object . represents only a single cleaning project.

Break down what the methods are within the class by organizing them by which stage ion the workflow they belong in .

Eg creating the object specifying a notion of valid values, running a search for invalid values, running a method to remediate these values and then running a method to produce the data as an output file.

Specify which methods are public and private ( public – user calls direct, private – internal to the class)

What are class attributes and how is it doing the job / justifying why I stored what I did as a class attribute

**Usage:** Description and examples of how the package should be used, how it fits into a workflow. This section may include code snippets, but it should primarily be a written description.

Insert snippets of user workflows (cleaned up and concise)

Write out workflow 1

5 step process :

Create object

Specific valid?invalid

Clean using method of choice

Use and output

1 interactive

2. non interactive

- why? Give the user the option to customize . some users will like that interface and involves writing less code but other uses prefer to write the specification themselves incase they want to reproduce it .

**Discussion:** Discuss how the package compares to related libraries, and how it fits into the overall ecosystem. Why should people use this package? How could the package be improved?

**REBECCA TO DO**

**Statement of contributions:** List the full names of the authors and how each member contributed to the completion of the project.

TO BE FINALIZED

**References:** Cite any external libraries used by the project, and any sources that were used as a reference. Use a consistent format and numbering scheme

**REBECCA TO DO**