‘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 datatype. 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 several methods that identify what is considered “invalid” or “valid” data.

**Design:**

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 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 variable one of the two modes will be more efficient depending on whether the variable has fewer valid or invalid potential values.

*Design Decision 4*: Determine how to rectify invalid observations when detected?

Invalid values can be rectified in one three ways. The user can determine which method is best suited to their data set for analysis. This is done through the **clean\_data** function which has the selection as an input. The first method (**\_clean\_data\_row\_drop**) drops all rows that contain at least one invalid value. The second method (**\_clean\_data\_replace\_value**) replaces all invalid values with user determined replacement value for each variable. The third method (**\_clean\_data\_replace\_average**) replaces invalid values with either the median if the variable is numeric or a mode for categorical variables.

Once the user has cleaned the data, CleanlockHolmes exports the cleaned data into a new csv file using the **write\_data** method.

**Text

Description automatically generatedUsage:**

Workflow 1 exhibits a user utilizing the CleanlockHolmes package to clean a data set. The user imports the package and instantiates the object. The user is then able to specify valid and invalid values for 5 different variables in this case. The user then cleans the data based on the valid and invalid values that they have already specified using method 3 which replaces invalid entries with the median or mode. Finally the user exports the cleaned data set to a new csv file.

Graphical user interface, text, application, email

Description automatically generated

Workflow 2 exhibits a user utilizing the CleanlockHolmes package to clean a data set. In this workflow the user interacts with the package to identify a viable range for the numeric values. This gives the user the option to customize. Some users prefer that interface in which they can specify ranges for themselves to reproduce the code. The user is also able to use a predetermined version of these interactive functions should they prefer.

**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**