Module 4: Cleaning Messy Data

DAT-375

Brandon Hobbs

March 19, 2023

Before data can be analyzed it must be cleaned so that it fits into the defined schema or data model. Extra white spaces are among the most common issue in data but also the most insidious as they are not easily noticed by a human. Another issue commonly seen is typing, i.e., storing integers as ASCII characters or strings – which can be an infuriating issue as normally 1+7 = 8 but in ASCII 1+7 = ‘h’!

Different tools have different methods for cleaning and restoring data to useable standards. In this short paper I will describe and compare methods in Excel and Python (particularly the library pandas).

|  |  |  |
| --- | --- | --- |
| Function | Excel | Python |
| Clearing extra spaces | Excel has the inbuilt function TRIM(): “Removes all spaces from text except for single spaces between words” | Python’s control over the whitespace is more granular. Strip() removes leading and trailing, Replace() replaces strings with other strings, e.g., 2 spaces for a single space. |
| Converting numbers stored as text into numerals | Excel indicates numbers stored as text in 2 ways: error carrot and aligning the value to the left side of the cell.  To fix this, the function VALUE() can be used, or more simply, multiply the number by 1 and Excel will convert automatically. | Python demands more granularity again. Python will ask that the numeric string is cast to a type, e.g., integer or float. It is safer to choose float since a random string can contain an integer or float, but an error is thrown casting a string float to integer. |
| Removing irrelevant or duplicate data | Excel has a few ways to remove duplicates. The easiest is to use the Remove Duplicates tool under Data:    Another method is to use an Advanced filter:      Irrelevant data is best determined using filters or Pivot tables. | Working with large blobs of data in Python generally calls for the package pandas. This package allows the use of data frames which makes data blobs portable and easy to work with.  From GeeksForGeeks (2022) using a panda DataFrame to identify any duplicates is easy:    The duplicates can then be dropped from the frame using the panada method, drop\_duplicates():    Irrelevant data is still best done through filters. panada also supports filters: |
| Fixing structural errors and altering formatting as needed | Excel has quite a few functions to alter or change a format. Text() converts a number into text with a specific format:  Dollar converts a numeric into a currency type.  Upper can force all characters to uppercase, lower does the opposite (these functions are important for comparisons)  Value will take a text number and convert to a numeric value.  Datevalue() is useful for converting string representing a date into an actual date type so that it may be used in more advanced ways, e.g., using month(), year(), weekday().  Then of course there are all of Excel’s string operators: len() for length, mid() for substrings, left() and right() for string slicing.  Excel’s GUI can also be used by right click and choosing FORMAT CELLS. | Python doesn’t employ static typing but rather dynamic, therefore, a variable can change type as needed through operation or casting.  However, sometimes it is important to control the type in Python. To convert a numeric to integer or float the value needs to be cast using int() or float().  ord() converts a character to an integer, hex() an integer to its equivalent hexadecimal string, and str() reverses that by changing an integer into a string.  When working with dates or datetime Strftime() will convert one of those formats into a string. This very useful for determing the day of the week a date is referencing, e.g., *print("The Current week is: ",current\_datetime.strftime("%W")).*  Python also has many string operators just as does Excel: len(), lower(), upper(), etc. |
| Filtering unnecessary outliers | To start filtering *outliers* needs to be defined. The center value is found using AVERAGE(), MEDIAN(), MODE().  The spread is STDEV.P() or STDEV.S().  Outliers are often found through quartiles and the IQR. Excel can calculate the quartile with QUARTILE. This could also be plotted with the same concept (below shows two outliers). | Python and panda have all the same functions Excel does.  Assuming the data was loaded into a DataFrame named df:  average is df[column].mean()  std dev is df[col].stf()  median is df[col].median()  mode is df[col].mode()  Quartiles are found using the percentile (GeeksForGeeks, 2022)    Then once the upper and lower whiskers are calculated any value outside this range could be removed in the DataFrame using drop: |
| Handling missing data | The first task is to determine how missing data is to be handled. Larose (2015) supplies four potential techniques:   1. Replace with some constant 2. Replace with the field mean (for numeric variables) or the mode (for categorical variables) 3. Replace with a value generated at random 4. Replace with imputed values based on the other characteristics of the record   In Excel there are no bulk update functions unless one defines one through a VBA function. This means that each blank needs to be identified manually.  Conditional formatting is helpful here to make them stand out. The user then just needs to compute the different vales as mentioned before and then replace the value in the red box.    Filtering the data in place and then selecting blanks is also very effective. | To detect missing values in a panda DataFrame isnull(), notnull(), dropna(), fillna(), replace(), or interpolate() could be used.  Again, GeeksForGeeks shows how to use isnull() to build a new series with any values equal to null:    Or, if replacing these values is more important fillna() could be used.    replace() may also be used with any of the methods mentioned in Larose (2015): |

**Citations**:

David. (2019, December 10). *How to clean messy data in R*. <https://rfortherestofus.com/2019/12/how-to-clean-messy-data-in-r>

GeeksforGeeks. (2022, February 16). *Find duplicate rows in a Dataframe based on all or selected columns.* <https://www.geeksforgeeks.org/find-duplicate-rows-in-a-dataframe-based-on-all-or-selected-columns>

Larose, D. T. (2015). Data mining and predictive analytics (2nd ed.). Wiley Global Research (STMS). <https://mbsdirect.vitalsource.com/books/9781118991121>

pandas.DataFrame.filter (2023, January 18). pandas 1.5.3 documentation. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.filter.html