Danny Teng

LHS 712

Assignment 1 Report

Regular Expression

* pat1 = r'(\d{1,2})[/-](\d{1,2})[/-](\d{4}|\d{2})'
* pat2 = r'(\d{1,2})[/-](\d{4}|\d{2})'
* pat3 = r'(\d{1,2}) (%s[a-z]\*),? (\d{4}|\d{2})' % MONTHS
* pat4 = r'(%s[a-z]\*) (\d{1,2})?,? (\d{4}|\d{2})' % MONTHS
* pat5 = r'(%s[a-z]\*) (\d{4}|\d{2})' % MONTHS
* pat6 = r'(\d{4})'

Cleaning Steps

1. Read the text file in using pandas read\_csv(), using ‘\t’ as delimiters.
2. Reassigned the index to the ‘Row’ column.
3. Created one function w/ helper function to process regular expressions row by row.
4. Applied functions to pandas data.frame, saving output to a new column in the data.frame.
5. Wrote the ‘Row’ and ‘Date’ columns to a new text file, using ‘\t’ as delimiter.

Issues

1. Minor issues with line 86 and 464 of the dates.txt file; the delimiter was somehow different, causing issues with the next 30 lines as well.
2. Originally, year was extracted using (\d{2,4}). This led to issues with pulling 3-digit values for year, leading to odd years being outputted (19163 for example).
   1. Year was then extracted using (\d{2}|\d{4}). Unfortunately, it defaulted to pulling the first two digits of years, causing some years (2002) to be outputted as 1920 instead.
   2. Issue was final fixed by searching for (\d{4}|\d{2}) instead; if no 4-digit year could be found, then moved onto 2-digit year.
3. As well, I originally only had 4 different regex patterns, but added 2 more to account for dd MONTH yy(yy) and MONTH yy(yy).

Overall, all lines have date conversions; individual accuracy of the lines is unknown.