We've recently made an accessibility improvement to the community and therefore posts without any content are no longer allowed. Please use the spoiler feature or add a short message in the message body in order to submit your weekly challenge. ×

2022-05-26 Updates: Email: If you're not seeing emails be delivered from the Community, please check your spam and mark the Community emails as not junk. Thank you for your patience. ×

SIGN IN

Free Trial

Weekly Challenge

Solve the challenge, share your solution and summit the ranks of our Community!

Also available in | Français | Português | Español | ???

IDEAS WANTED

We're actively looking for ideas on how to improve Weekly Challenges and would love to hear what you think!

SUBMIT FEEDBACK ▸

< Weekly Challenge

Challenge #56: Parsing and Counting Hashtags

JoeM
Alteryx Alumni (Retired)

Last week's solution can be found HERE.

This week's challenge, by its title, sounds like a simple task. However, a quick look at the data will show that it's not going to be just a summarize tool. If you have been meaning to finally learn the RegEx tool, this exercise will be a great first foray with a relatively simple expression. In this exercise, we will take a set of very ugly data, parse it to find hashtags within a text field, find out how many times the hashtag was written, and what users used the hashtag.

If you would like to learn more about our RegEx Tool - check out our help on the tool. If you would like a simpler RegEx overview, check this interactive learning out.

Good luck!

challenge_56_start_file.yxmd

challenge_56_solution.yxzp

Data Analysis   Intermediate   Parse   Preparation   Transform

Share   3 LIKES   Reply

ACE **mceleavey**
17 - Castor

I'm not sure the results provided are correct as they show a #followme result against id_str_13, but that text does not contain a hashtag, unless I'm missing something?

That aside, my solution is as follows:

▷ Spoiler

bu:lien

challenge_56_solution.yxmd

Share   3 LIKES   Reply

ACE **MarqueeCrew**
19 - Altair

Of course I wanted to solve this without the use of RegEx (just being difficult). I don't agree that id_str_13 should have any counts.

▷ Spoiler

Alteryx ACE & Top Community Contributor

Chaos reigns within. Repent, reflect and restart. Order shall return.
Please Subscribe to my youTube channel.

challenge_56_start_file (My answer).yxmd

Share 2 LIKES Reply

JoeM
Alteryx Alumni (Retired)

Good catch @mceleavey and @MarqueeCrew, I have updated the start file to have the correct expected results.

Share 2 LIKES Reply

Max06270
7 - Meteor

Nice challenge, have a great week guys!

▷ Spoiler

challenge_56_MK.yxmd

Share 1 LIKE Reply

Harbinger
9 - Comet

▷ Spoiler

I have attached my work for a deep dive-- fun stuff!

challenge_56_start_file_JHCompleted.yxzp

Share 2 LIKES Reply

jury_maggay
5 - Atom

Hello,

I would like to know how the expression #.*?\> works. I was also trying to solve the challenge using RegEx tool but can't seem to figure out the correct expression to use. Appreciate your feedback on this :-)

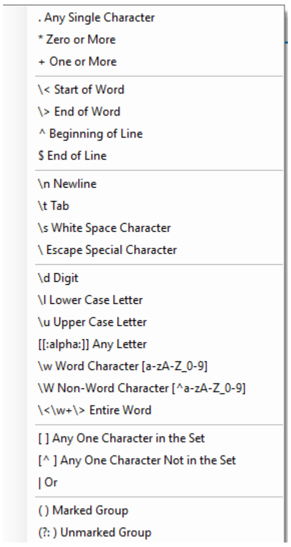
Share 1 LIKE Reply

ACE mceleavey
17 - Castor

In regex the .*? part simply means "take everything in between". From the dropdown you'll see a list of Regex tools that allow you to then construct things like that seen in the example.
The # here, followed by the .*? simply means take everything from the # symbol to...
I then used the \> which means "the end of the word".
So, in speech terms, it literally translates to "take everything from the # symbol to the end of the word".

Regex really isn't as scary as it first appears. I myself had no Regex experience before using Alteryx but it's really easy once you get your head around it.

Try experimenting with the dropdown list:



. Any Single Character
* Zero or More
+ One or More

\< Start of Word
\> End of Word
^ Beginning of Line
$ End of Line

\n Newline
\t Tab
\s White Space Character
\ Escape Special Character

\d Digit
\l Lower Case Letter
\u Upper Case Letter
[[:alpha:]] Any Letter
\w Word Character [a-zA-Z_0-9]
\W Non-Word Character [^a-zA-Z_0-9]
\<\w+\> Entire Word

[ ] Any One Character in the Set
[^] Any One Character Not in the Set
| Or

( ) Marked Group
(?:) Unmarked Group

Feel free to message me if you have any questions or if you have a specific issue with which you're struggling.

bulien

Share 5 LIKES Reply

jury_maggay
5 - Atom

Thank you so much for the explanation. I will surely reach out to you guys here in the forum should I need help on anything about Alteryx. Thanks

Share 1 LIKE Reply

Harbinger
9 - Comet

Also, a great reference for experimentation is regexr.com. You can paste your own test text in the text box and get real-time feedback on how your regular expression evaluates on that text.

Share 4 LIKES Reply

‹ 1 2 3 ... 28 ›

All forum topics ‹ ›

La

COMMUNITY INDUSTRY AWARDS WINNER 2021