

# Regular Expressions

As you read, regular expressions (or regex) provide a language for describing a search pattern that will match patterns in a string. They can get very complex, but usually we only need relatively simple regular expressions for data science applications.

<code>\A</code>	start of string	
<code>\Z</code>	end of string	[note: you will need to write two slashes in R]
<code>\w</code>	match word character (note: lowercase w)	
<code>\W</code>	match any non-word character (note: uppercase W)	
<code>.</code>	any character (other than new line)	
<code>[ ]</code>	match any (single) character inside bracket	
<code>[ ]*</code>	match 0 or more characters inside bracket	
<code>[ ]+</code>	match 1 or more characters inside bracket	
<code>[^ ]</code>	match any number of character NOT inside bracket	
<code>0-9</code>	numeric digits	
<code>a-z</code>	lowercase letters	[note: these must be used inside brackets]
<code>A-Z</code>	uppercase letters	

# Example #1

How would you describe the following regex query?

**`[\w]+ing`**

## Example #2

How would you describe the following regex query?

`"[^"]+"`

# stringi

The stringi package is probably my favorite R package—at least, my favorite R package that I did not create! It provides an amazing amount of fast functionality that almost always just works as I expect it to. All functions start with the prefix “stri\_” and most accept either a regex OR fixed string search pattern (the latter turns out to be very helpful in a lot of cases).

As an example, here is an example of a regular expression that counts the number of words (from the latin alphabet) with capital letters:

```
stri_count(input, regex = "[A-Z][\\w-']+")
```

# Using Regular Expressions

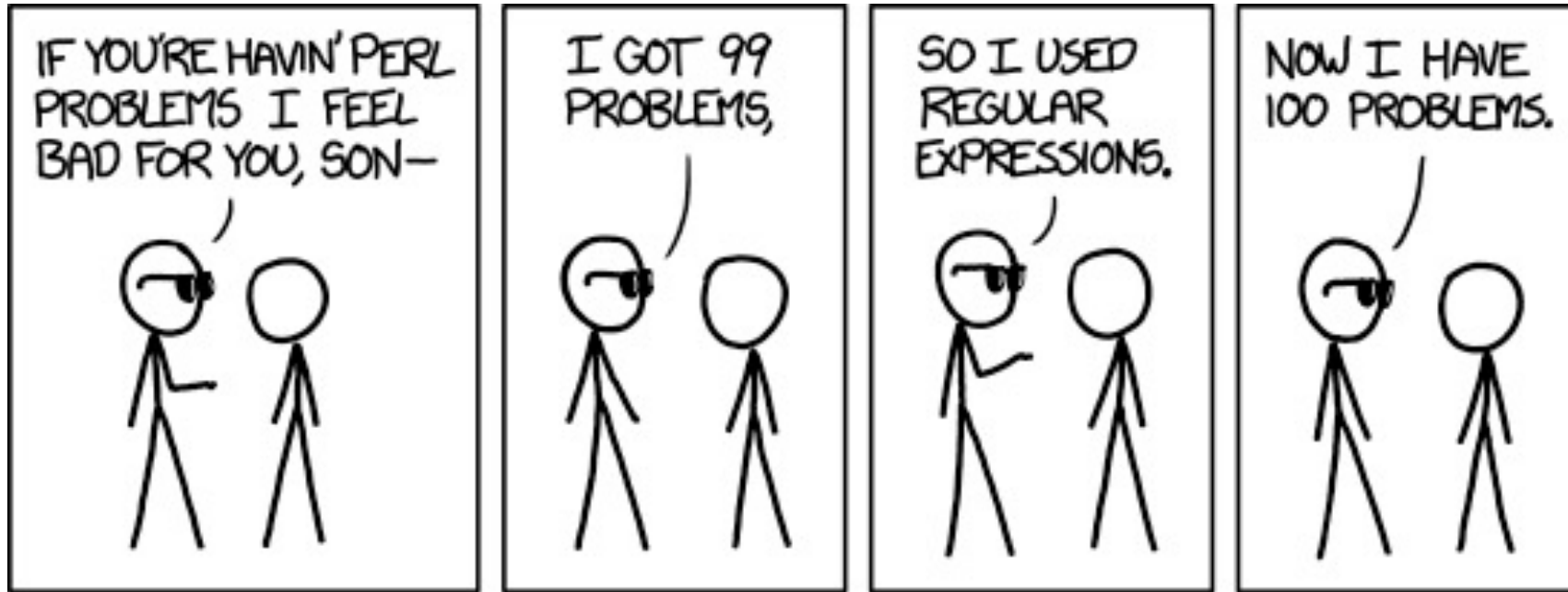
There are several common things we can do once we are able to match a pattern in a string, all supported by the R stringi package:

<code>stri_count</code>	count the number of time the pattern occurs
<code>stri_detect</code>	does pattern occur at least once?
<code>stri_extract_all</code>	extract all occurrences of the pattern
<code>stri_extract_first</code>	extract the first occurrences of the pattern
<code>stri_replace_all</code>	replace all occurrences of the pattern with a fixed string
<code>stri_split</code>	split string into substrings, using patterns as the split

There are also some related stringi functions that do not take a regular expression as an input:

<code>stri_length</code>	how many characters are in the string
<code>stri_sub</code>	take a substring by position
<code>stri_trans_toupper</code>	convert to uppercase
<code>stri_trans_tolower</code>	convert to lowercase
<code>stri_trans_totitle</code>	convert to title case

# Regular Expressions



<https://xkcd.com/1171/>

WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.

OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!



BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!


IT'S HOPELESS!



EVERYBODY STAND BACK.




I KNOW REGULAR EXPRESSIONS.



PERL!

TAP TAP



<https://xkcd.com/208/>