Regular Expression Reference (not comprehensive)

BIOS 669

Characters

Use to look for specific characters

Z would find all upper-case Z’s

r would find all lower-case r’s

3 would find all 3’s

= would find all equals signs

qu would find all sequences of a lower-case q followed by a lower-case u

Special characters

/ / a pair of forward slashes surrounds regular expressions in any context (at least any context that we will use)

| used for a logical OR; parentheses ( ) can be used to group items being OR'ed

\ sometimes used as part of a metacharacter (as in \d) and sometimes as an "escape character" (that is, to say to treat the next character as itself and not as the special meaning it normally has in regular expressions; for example, . by itself is a metacharacter meaning to match any single character, but \. means to match an actual dot)

Metacharacters

.	A wildcard matching any single character
\d	Match any single digit between 0 and 9
\D	Match any single non-digit
\w	Match any single digit, letter, or underscore, but not hyphen, dash, space, etc.
\W	Match any non-word character (that is, anything that \w does not match)
\s	Match a single whitespace character, the most important of which is a space or blank
\n	Match a newline character

Repetition modifiers

*	Match the preceding character 0 or more times – “greedy”
+	Match the preceding character 1 or more times – “greedy”
?	Match the preceding character 0 or 1 times
{n}	Match the preceding character exactly n times
{n,}	Match the preceding character n or more times
{n,m}	Match the preceding character between n and m times

“Greedy” means that characters continue to be grabbed until they no longer match. To make a metacharacter non-greedy (grab characters as few times as necessary to match), use *? or +? .

Groupers

[ ]	Match any one of the characters or meta-characters included between the brackets
[-]	Match any one of the characters in the specified range
( )	Can be used to group items being OR'ed with |. Also used to mark "capture groups", which are not covered in this introduction to regular expressions.

Anchors

^	Look for the succeeding pattern only at the beginning of a line
$	Look for the preceding pattern only at the end of a line
\b	Look for the preceding pattern only as a complete word
\B	Look for the preceding pattern only as part of a word

(Sometimes ^ also means “not” – it’s context-specific.)

Not covered

There are lots of metacharacters and modifiers not covered in this reference, but you can find information online, of course.

For example, all sample matches that we will do use only upper-case letters, but there are case modifiers that you can use to request that all of a search ignores case, that only lower-case characters be considered a match, and so forth.

References

Cassell, David, *The Basics of the PRX Functions*
http://www2.sas.com/proceedings/forum2007/223-2007.pdf

Pless, Richard, *An Introduction to Regular Expressions with Examples from Clinical Data*
http://www2.sas.com/proceedings/sugi29/043-29.pdf

Windham, K. Matthew. 2014. Introduction to Regular Expressions in SAS®. Cary, NC: SAS Institute Inc.