Just super bare bones:

<http://www.petefreitag.com/cheatsheets/regex/>

Well organized, easy to find things, no examples

<http://www.cheatography.com/davechild/cheat-sheets/regular-expressions/>

Extensive, and with examples, but little harder to find things:

<http://www.rexegg.com/regex-quickstart.html>

R-SYNTAX FOR REGEX

Here's how w/o filtering

isUnited <- grepl("^United", dtGDP$Long.Name)

summary(isUnited)

## Mode FALSE TRUE NA's

## logical 211 3 0

match BOTH patterns regardless of order

(pattern1.\*pattern2)|(pattern2.\*pattern1)

res <- grepl("(june).\*(fiscal year end)|(fiscal year end).\*(june)", tolower(ijGE$Special.Notes))

summary(res)

**Match all possible cases with []**

**Character Classes with []**

We can list a set of characters we will accept at a given point in the match

[Bb][Uu][Ss][Hh]

**^[Ii] am**

Will match ‘i am’ or ‘I am’ , only at the beginning of a line

Similarly, you can specify a range of letters [a-z] or [a-zA-Z]; notice that the order doesn’t matter

**^[0-9][a-zA-Z]**

will match the lines

7th inning stretch

2nd half soon to begin. OSU did just win something

3am - cant sleep - too hot still.. :(

5ft 7 sent from heaven

1st sign of starvagtion

\*Just to fuck with you, When used at the beginning of a character class, the carrot is also a metacharacter and indicates matching characters NOT in the indicated class

[^?.]$

# return any line that does NOT(^) end($) with a ‘?’ or ‘.’

^[Gg]ood|[Bb]ad

#Will match G/good at the start of the line OR B/bad ANYWHERE in the line

^([Gg]ood|[Bb]ad)

#Will match G/good or B/bad at the start of the line

The question mark indicates that the indicated expression is optional

[Gg]eorge( [Ww]\.)? [Bb]ush

# the period is a metacharacter and must be escaped.

#a ‘w’ cap or lower, with or w/o a period OR nothing in middle at all (george bush) will be matched

**More metacharacters: \* and +**

The \* and + signs are metacharacters used to indicate repetition; \* means “any number, including none, of the item” and + means “at least one of the item”

**(.\*)**

#will match the lines

anyone wanna chat? (24, m, germany)

hello, 20.m here... ( east area + drives + webcam )

(he means older men)

()

{ and } are referred to as interval quantifiers; the let us specify the minimum and maximum number of matches of an expression

[Bb]ush( +[^ ]+ +){1,5} debate #(space+[not space]+space+)

#See the pattern ‘space word space’ between 1 & 5 times between Bush at start and ‘debate’

{m,n} means at least m but not more than n matches

{m} means exactly m matches

{m,} means at least m matches

In most implementations of regular expressions, the parentheses not only limit the scope of alternatives divided by a “|”, but also can be used to “remember” text matched by the subexpression enclosed

We refer to the matched text with \1, \2, etc.

**+([a-zA-Z]+) +\1 +**

#space,pattern of characters, space, SAME pattern of characters, space

#will match the lines

time for bed, night night twitter!

blah blah blah blah

my tattoo is so so itchy today

i was standing all all alone against the world outside...

hi anybody anybody at home

estudiando css css css css.... que desastritooooo

**The \* is “greedy” so it always matches the longest possible string that satisfies the regular expression.**

**^s(.\*)s**

#matches

sitting at starbucks

setting up mysql and rails

studying stuff for the exams

spaghetti with marshmallows

stop fighting with crackers

sore shoulders, stupid ergonomics

**Return TRUE only if 2 patterns occur (in any order) in a string**

(pattern1.\*pattern2)|(pattern2.\*pattern1)

#Here’s how to do it in R

res <- grepl("(june).\*(fiscal year end)|(fiscal year end).\*(june)", tolower(ijGE$Special.Notes))