In a broad term, pattern matching, as the name implies, is to match patterns between two entities. In terms of data wrangling and specifically when it comes to textual data, pattern matching is done not in a one-to-one matching, but based on a given pattern. The classical “beginswith”,” contains”, and “endswith” patterns we nice, but we need more capability than that when we want to do things like sentimental analysis with textual data. Simple example is to remove special characters from a string (i.e. '</?.*?>'), to removing stop words such as ‘I’ ,’ You”, etc. Regular expressions have evolved to where can now use pattern matching to decipher and dissect text. The tutorial in “Mastering quantifiers” offers a good description of quantifiers which can be greedy or lazy. The article argues that quantifiers are more than just greedy and “lazy” and that we must choose quantifiers carefully, and having in mind the following:

The greedy trap – The quantifiers may return more matches that expected

Pattern matching - https://en.wikipedia.org/wiki/Pattern_matching

Mastering quantifiers - https://www.rexegg.com/regex-quantifiers.html

Pattern matching - https://docs.scala-lang.org/tour/pattern-matching.html