

WEEK 6: REGULAR EXPRESSIONS

THOMAS ELLIOTT

Regular expressions are advanced ways of searching and manipulating characters. For an introduction to how regular expressions work more generally, search your favorite search engine for regular expressions. This website (<http://www.regular-expressions.info/>) has a lot of information about how regular expressions work.

R comes with a number of functions for using regular expressions. They share the same help page, so you can see what is available by typing `?grep`. By default, these functions use the POSIX implementation of regular expressions. You can tell R to use perl-style regular expressions by supplying the `perl=TRUE` argument to any of the following functions.

`grep()` — this function takes a character vector and returns a vector of indices that match the regular expression. For example:

```
> test<-c("apple","banana","orange","pear")
> grep("ap",test)
[1] 1
```

`grepl()` — this function takes a character vector and returns a logical vector indicating whether the corresponding element in the character vector matches the regular expression.

```
> grepl("ap",test)
[1] TRUE FALSE FALSE FALSE
```

`sub()` — this function takes a character vector and, in each element, replaces the first match with the regular expression with a replacement character.

```
> sub("[aeiou]","4",test)
[1] "4pple" "b4nana" "4range" "p4ar"
```

`gsub()` — this function takes a character vector and, in each element, replaces the all matches with the regular expression with a replacement character.

```
> gsub("[aeiou]","4",test)
[1] "4ppl4" "b4n4n4" "4r4ng4" "p44r"
```

There are three more regular expression functions in the help file, but the returned objects are more complicated and you will not likely use them when beginning with R. You can read about what they return (and play around with what they return) on your own.