**R commands Useful For Data Cleaning**

The R Statistical Software that is used in this monograph has many useful editing tools. The functions **grep**, **sub** and **regexpr** are particularly useful. The function **grep** searches for matches to a specified pattern (its first argument) within the studied character vector x (its second argument). The output of the function **grep** is a vector that gives either the indices of the elements of x that yield a match or, if the function argument value is set TRUE, the matched elements of x.

The variable **text** in the following toy example contains a single sentence of seven words (a string of characters, containing letters, numbers and punctuation symbols, with words separated by a single blank space). This sentence is then split (parsed) into a vector of seven elements (tokens), **textm**. Here are a few simple examples that illustrate the great versatility of the function **grep**. Running the subsequent code (that is also given on our webpage) illustrates what is being done.

**text="The gentle.man ca7nnot have any-thing to explain" ## text example**

**text**

**text=tolower(text) ## makes everything lower case**

**text**

**textw=strsplit(text," ")[[1]] ## splits the text into its elements (tokens)**

**textw**

**grep("e",textw,ignore.case=TRUE,value=T)**

**## list of all elements that include the search item "e"**

**grep("e",textw) ## index of elements that include the search item "e"**

**grep("e",textw,value=T) ## list of elements**

**grep("gent",textw,ignore.case=TRUE,value=T)**

**grep("ent",textw,ignore.case=TRUE,value=T)**

**grep("get",textw,ignore.case=TRUE,value=T)**

**grep("n{2}",textw,value=T) ## searches for the presence of "nn"**

**grep("nn",textw,value=T)**

**grep("e....i",textw,value=T)**

**## searches for presence of string with "e", followed by any four characters and "i"**

**grep("the|gen",textw,value=T)**

**## searches for presence of either one expression or the other**

**grep("[[:alnum:]]{3}",textw,value=T)**

**## searches for the presence of three alphanumeric characters in row**

**grep("[[:alnum:]]{7}",textw,value=T)**

**grep("[[:alnum:]]{8}",textw,value=T)**

**grep("[[:blank:]]",textw,value=T) ## searches for presence of blank**

**grep("[[:punct:]]",textw,value=T) ## searches for presence of punctuation**

**grep("[[:punct:]]{2}",textw,value=T)**

**grep("[[:digit:]]",textw,value=T) ## searches for presence of digit**

**grep("[[:space:]]",textw,value=T) ## searches for presence of space**

**grep("[.]",textw,value=T) ## searches for period; need [] as . a special symbol**

**grep("[\*]",textw,value=T) ## searches for \*; need [] as \* a special symbol**

**grep("[$]",textw,value=T) ## searches for $; need [] as $ special symbol**

**grep("-",textw,value=T) ## searches for presence of dash**

**grep("7",textw,value=T) ## searches for presence of number "7"**

**grep("^t",textw,value=T,ignore.case=TRUE)**

**## searches for presence of letter t at beginning**

**grep("^.",textw,value=T,ignore.case=TRUE)**

**## searches for presence of anything at beginning**

**grep("^[[:digit:]]",textw,value=T,ignore.case=TRUE)**

**## searches for presence of digit at beginning**

**grep("^[[:alnum:]]",textw,value=T,ignore.case=TRUE)**

**## searches for presence of alphanumeric symbols at beginning**

**grep("[[:alnum:]]{6}",textw,value=T,ignore.case=TRUE) ## six consecutive**

**grep("[[:alnum:]]{6}[.]",textw,value=T,ignore.case=TRUE)**

**## searches for string of 6 consecutive alums followed by .**

**grep("[[:alnum:]]{6}[7]",textw,value=T,ignore.case=TRUE)**

**## searches for string of 6 consecutive alums followed by "7"**

**grep("t$",textw,value=T,ignore.case=TRUE)**

**## searches for "t" at the end of element**

**grep("t$|n$",textw,value=T,ignore.case=TRUE)**

**## searches for "t" or "n" at the end of element**

**grep(".$",textw,value=T,ignore.case=TRUE)**

**## searches for anything at the end of element**

The function **sub** (and **gsub**, when carrying out the operation multiple times) replaces one pattern (its first argument) with another pattern (its second argument), all within the studied character vector x (its third argument). Try the following examples

**t=gsub("[[:punct:]]","",textw) ## removes any punctuation**

**t**

**t=gsub("[.]","",textw) ## removes the period**

**t**

**t=gsub("^.","O",textw) ## replaces any beginning character with O**

**t**

**t=gsub(".$","O",textw) ## replaces any ending character with O**

**t**

**grep("[[:alnum:]]{2}[.]",textw,value=T,ignore.case=TRUE)**

**## searches for 2 consecutive alnum characters followed by "."**

**t=gsub("[[:alnum:]]{2}[.]","88.2",textw) ## replaces this with "88.2"**

**t**

**textw[1]=" the"**

**grep("^ ",textw,value=T,ignore.case=TRUE) ## searches for a starting blank**

**t=gsub("^ ","",textw) ## removes a single blank**

**t**

**grep("^[[:space:]]{1,}",textw,value=T,ignore.case=TRUE)**

**## searches for one or more empty spaces before the first character**

**t=gsub("^[[:space:]]{1,}","",textw)**

**## removes all empty spaces before the first character**

**t**

**textw[1]="the"**

The function **regexpr** uses the specified pattern (its first argument) to create an integer vector of the same length as the studied character vector (the second argument; in our example, the input textw is a vector of length 7) that contains the starting position of the first match, or -1 if there is none. It also creates the attribute "match.length" that gives the length of the matched text (or -1 if there is no match). The function **gregexpr** creates a list with the starting positions of all matches (not just the first). These functions are similar to **grep**, but return the information in more detail, and also in a different format. Try the following examples to appreciate the flexibility of these functions.

**regexpr("e",textw,ignore.case=TRUE)**

**## information on vector elements with match and position at which match occurs**

**## if no match for certain element, value -1 is assigned.**

**## match with "e" in four elements of textw: elements 1, 2, 4 and 7)**

**## "e" occurred (first) at positions 3, 2, 4 and 1, respectively.**

**gregexpr("e",textw,ignore.case=TRUE)**

**## looks for all multiple occurrences of specified pattern, not just the first**

**freq=as.vector(unlist(lapply(gregexpr("e",textw,ignore.case=TRUE),length)))**

**## obtains the vector of number of occurrences**

**freq**

**regexpr("gent",textw,ignore.case=TRUE)**

**regexpr("ent",textw,ignore.case=TRUE)**

**regexpr("get",textw,ignore.case=TRUE)**