Northeastern University

CS6020: Collecting, Storing, and Retrieving Information

Basic Data Shaping

Basic Data Shaping

TEXT PROCESSING

Lesson Objectives

- After completing this lesson, you are able to:
  - process character and text values
  - install the ***stringr*** package for text processing
  - manipulate text using ***stringr***

Basic Text Processing Functions

Base R Function	*stringr* Function
`paste()` – concatenates a vector of characters with spaces in between	`str_c()` – similar to paste() but does not insert extra spaces and removes empty strings
`nchar()` – returns the length of a string. For `NA` it returns 2.	`str_length()` – same as `nchar()` but handles `NA` correctly.
`substr()` – extracts or replaces a substring sequence	`str_sub()` – similar to `substr()` but also accepts negative values for offsets.
	`str_dup()` – duplicates a string
	`str_trim()` – removes leading and trailing spaces
	`str_pad()` – pad string with extra whitespace on left or right

Basic Data Shaping

REGULAR EXPRESSIONS

What is a Regular Expression

- A special text string for describing a search pattern.
- Similar to wild cards, Searching all files in directory: *.txt, Regular expression: ".*\.txt$"
- Also known as Reg Exp.

Meta characters in Reg Exp

- . → Normally matches any character except a newline. Within square brackets the dot is literal.
- () → Groups a series of pattern elements to a single element.
- + → Matches preceding operator one or more times.
- ? → Matches preceding operator zero or one time

Usage examples

- [hc]at: matches “hat” or “cat”
- .at: matches any 3 characters ending with “at”
- [^b]at: works like .at but do not matches “bat”
- [^hc]at: matches all string like “.at” but skips “hat” and “cat”
- ^[hc]at: matches "hat" and "cat", but only at the beginning of the string or line.
- Many more

Usage examples

- [hc]at: matches “hat” or “cat”
- .at: matches any 3 characters ending with “at”
- [^b]at: works like .at but do not matches “bat”
- [^hc]at: matches all string like “.at” but skips “hat” and “cat”
- ^[hc]at: matches "hat" and "cat", but only at the beginning of the string or line.
- Many more

Regexp in R

- Two types of regular expression in R:
  - Extended regular expression: They use an implementation of the POSIX 1003.2 standard: that allows some scope for interpretation and the interpretations here are those currently used by R.
  - Perl-like regular expression: pattern matching using the same syntax and semantics as Perl 5.10
- Available in base package

Basic Data Shaping

THE GREP FUNCTION

The **grep** Function

- Search for matches to argument pattern within each element of a character vector.
- If you pass value=FALSE, returns a new vector with the indexes of the elements in the input vector that could be matched by the regular expression.
- If you pass value=TRUE, returns a vector with copies of the actual elements in the input vector that could be matched.

Grep function usage

- Takes 2 inputs, first: Regular Expression and Second: Character vector.
- Example with value=FALSE:

```
> grep("a+", c("abc", "def", "cba a", "aa"), perl=TRUE, value=FALSE)
[1] 1 3 4
```

- Example with value= TRUE:

```
> grep("a+", c("abc", "def", "cba a", "aa"), perl=TRUE, value=TRUE)
[1] "abc"    "cba a" "aa"
```

Summary

- In this lesson, you learned that:
  - strings are an important data value in R
  - R has built-in functions to deal with strings and characters but that the ***stringr*** package is available to ease text processing
  - the **grep** function allows parsing of strings based on regular expressions

Summary, Review, & Questions…