Handling strings

Andrew Ba Tran

Contents

String length ................................................................. 2
Combine strings .............................................................. 2
subset strings ................................................................. 3
detect matches ............................................................... 4
count matches ............................................................... 4
extract matches .............................................................. 4
split strings ................................................................. 4
replace a pattern ........................................................... 5
change case ................................................................. 5
trim strings ................................................................. 5
Your turn ................................................................. 5

This is from the third chapter of learn.r-journalism.com.

We’re going to use the stringr package to manipulate text.

```r
#install.packages("stringr")
library(stringr)
library(dplyr)
```

Each function starts with `str_`

Let’s load this data in:

```r
messy <- data.frame(name=c("Bill Smith", "jane doe", "John Forest-William"),
                    email=c("bsmith@themail.com", "jdoe@themail.com",
                            "jfwilliams$geemail.com"),
                    income=c("$90,000", "$140,000", "E8500"),
                    phone=c("(203) 847-334", "207-999-1122", "2128487345"),
                    activites=c("fishing, sailing, planting flowers", "reading,
                                raising flowers, biking", "hiking, fishing"))
```

```r
messy
```

<table>
<thead>
<tr>
<th></th>
<th>name</th>
<th>email</th>
<th>income</th>
<th>phone</th>
<th>activites</th>
</tr>
</thead>
<tbody>
<tr>
<td>bill</td>
<td>Bill Smith</td>
<td><a href="mailto:bsmith@themail.com">bsmith@themail.com</a></td>
<td>$90,000</td>
<td>(203) 847-334</td>
<td>fishing, sailing, planting flowers</td>
</tr>
<tr>
<td>jane</td>
<td>jane doe</td>
<td><a href="mailto:jdoe@themail.com">jdoe@themail.com</a></td>
<td>$140,000</td>
<td>207-999-1122</td>
<td>reading, raising flowers, biking</td>
</tr>
<tr>
<td>john</td>
<td>John Forest-William</td>
<td>jfwilliams$geemail.com</td>
<td>E8500</td>
<td>2128487345</td>
<td>hiking, fishing</td>
</tr>
</tbody>
</table>

What problems do you see?

Tasks

1. Split name into First name and Last name
2. Convert names to title case
3. Create a new variable identifying bad email addresses
4. Convert income to a new number in US dollars
5. Create a new variable containing area code
6. Creating a new variable counting how many activities each person is engaged in
7. Break activities into a set of useful dummy codes

<table>
<thead>
<tr>
<th>Order of elements in date-time</th>
<th>Parse function</th>
</tr>
</thead>
<tbody>
<tr>
<td>str_length()</td>
<td>figure out length of string</td>
</tr>
<tr>
<td>str_c()</td>
<td>combine strings</td>
</tr>
<tr>
<td>str_sub()</td>
<td>substitute string</td>
</tr>
<tr>
<td>str_detect()</td>
<td>detect string in string</td>
</tr>
<tr>
<td>str_match()</td>
<td>does string match</td>
</tr>
<tr>
<td>str_count()</td>
<td>count strings</td>
</tr>
<tr>
<td>str_split()</td>
<td>split strings</td>
</tr>
<tr>
<td>str_to_upper()</td>
<td>convert string to upper case</td>
</tr>
<tr>
<td>str_to_lower()</td>
<td>convert string to lower case</td>
</tr>
<tr>
<td>str_to_title()</td>
<td>convert the first letter of each word to upper case</td>
</tr>
<tr>
<td>str_trim()</td>
<td>eliminate trailing white space</td>
</tr>
</tbody>
</table>

String length

str_length(string) counts the number of characters in each element of a string or character vector.

```r
x <- c("Bill", "Bob", "William")
str_length(x)
```

```
## [1] 4 3 7
```

Combine strings

str_c(strings, sep="")

It’s like the equivalent of =concatenate in Excel.

But there are a couple of quirks

```r
data <- data.frame(place=c("HQ", "HQ", "HQ"),
                   id=c("A", "B", "C"),
                   number=c("001", "002", "003"))
data
```

```
# place id number
## 1 HQ A 001
## 2 HQ B 002
## 3 HQ C 003
```

We can add a string to each value in the number column this way:

```r
data <- data %>%
  mutate(combined=str_c("Num: ", number))
data
```

```
# place id number combined
## 1 HQ A 001 Num: 001
```
A couple options that would've done the same thing:

```r
data$combined <- str_c("Num: ", data$number)
```

# or

```r
data <- data %>%
mutate(combined=str_c("Num", number, sep": "))
```

You can also pass the variable `collapse` to `str_c()` if you're turning an array of strings into one.

```r
data <- tibble(place="HQ", "HQ", "HQ"),
id="A", "B", "C"),
number="001", "002", "003")
```

```r
data
```

```r
## place id number
## 1 HQ A 001
## 2 HQ B 002
## 3 HQ C 003
```

```r
data %>%
group_by(place) %>%
summarize(ids_combined=str_c(number, collapse="-"))
```

```r
## # A tibble: 1 x 2
## place ids_combined
## <fct> <chr>
## 1 HQ 001-002-003
```

subset strings

```r
str_sub(strings, start, end) extracts and replaces substrings
```

```r
x <- "Dr. James"

str_sub(x, 1, 3)
```

```r
## [1] "Dr."
```

```r
str_sub(x, 1, 3) <- "Mr."
```

```r
x
```

```r
## [1] "Mr. James"
```

Negative numbers count from the right.

```r
x <- "baby"

str_sub(x, -3, -1)
```

```r
## [1] "aby"
```

```r
str_sub(x, -1, -1) <- "ies"
```
detect matches

\texttt{str\_detect(strings, pattern)} returns T/F

\begin{verbatim}
x <- c("Bill", "Bob", "David.Williams")
x
## [1] "Bill" "Bob" "David.Williams"
str\_detect(x, "il")
## [1] TRUE FALSE TRUE
\end{verbatim}

count matches

\texttt{str\_count(strings, pattern)} count number of matches in a string

\begin{verbatim}
x <- c("Assault/Robbery/Kidnapping")
x
## [1] "Assault/Robbery/Kidnapping"
str\_count(x, "/")
## [1] 2
# How many offenses
str\_count(x, "/") + 1
## [1] 3
\end{verbatim}

extract matches

\begin{verbatim}
x <- c("bsmith@microsoft.com", "jdoe@google.com", "jfwilliams@google.com")
str\_extract(x, @.+.\com$)
## [1] @microsoft.com @google.com @google.com
\end{verbatim}

split strings

\texttt{str\_split(string, pattern)} split a string into pieces

\begin{verbatim}
x <- c("john smith", "mary todd", "bill holis")
str\_split(x, ",", simplify=TRUE)
## [,1] [,2]
## [1,"john" "smith"
## [2,"mary" "todd"
## [3,"bill" "holis"

first <- str\_split(x, ",", simplify=TRUE)[,1]
last <- str\_split(x, ",", simplify=TRUE)[,2]
\end{verbatim}
replace a pattern

```r
str_replace(strings, pattern, replacement) replace a pattern in a string with another string
```

```r
x <- c("john smith", "mary todd", "bill holis")
str_replace(x, "[aeiou]", "-")
```

```r
## [1] "j-hn smith" "m-ry todd" "b-l holis"
```

```r
str_replace_all(x, "[aeiou]", "-")
```

```r
## [1] "j-hn sm-th" "m-ry t-dd" "b-l h-l-s"
```

describe case

```r
str_to_upper(strings) is upper case str_to_lower(strings) is lower case str_to_title(strings) is title case
```

```r
x <- c("john smith", "Mary Todd", "BILL HOLLIS")
str_to_upper(x)
```

```r
## [1] "JOHN SMITH" "MARY TODD" "BILL HOLLIS"
```

```r
str_to_lower(x)
```

```r
## [1] "john smith" "mary todd" "bill hollis"
```

```r
str_to_title(x)
```

```r
## [1] "John Smith" "Mary Todd" "Bill Hollis"
```

describe strings

```r
str_trim(strings) remove white space at the beginning and end of string
```

```r
x <- c(" Assault", "Burglary ", " Kidnapping ")
str_trim(x)
```

```r
## [1] "Assault" "Burglary" "Kidnapping"
```

Your turn

Challenge yourself with these exercises so you’ll retain the knowledge of this section.

Instructions on how to run the exercise app are on the intro page to this section.