Working with String Variables using base R and stringr

Dr. Austin Brown

Kennesaw State University

12/6/2021

Introduction

- A special type of variable we commonly encounter is a character string, containing text
  - E.g., A customer's home address
- On occasion, we may have to manipulate these text strings in various ways to create new columns and/or extract specific pieces of information from said string
  - E.g., maybe we want to isolate the city someone's home address is in
- To do this, we need to work with special functions specifically designed to achieve these aims
  - We have functions in both base R as well as in the `stringr` package to help us out!

String Variable Basics

- At its core, string variables are comprised of "regular expressions." For example:

```
funny_shows <- c("Arrested Development",
           "The Office",
           "Curb Your Enthusiasm")

knitr::kable(funny_shows,
             col.names="Dr Brown's Favorite Shows")
```

Dr Brown's Favorite Shows
Arrested Development
The Office
Curb Your Enthusiasm

String Variable Basics

- Okay, so suppose we want to know how many words there are in each element of this character vector. We can use a combination of `sapply` and `stringr::str_split` to help us out!

```
sapply(stringr::str_split(funny_shows," "),
       length)
```

```
## [1] 2 2 3
```

String Variable Basics

- What if we wanted to know how many characters are in each string of text?

```
stringr::str_length(funny_shows)
```

```
## [1] 20 10 20
```

String Variable Basics

- Let's look at a slightly more applied example. Suppose we have a dataset with MLB player names which are stored in a column with the ordering of: lastname, firstname.

```
baseball <- readxl::read_xlsx('baseball.xlsx')
```

String Variable Basics

- We can see that the first word of the string is the player's lastname followed by a comma and a space and then the firstname.

```
## Examine the Structure of name ##
knitr::kable(baseball$Name[1:5],
             col.names="Name")
```

Name
Allanson, Andy
Ashby, Alan
Davis, Alan
Dawson, Andre
Galarraga, Andres

String Variable Basics

- So our "delimiter" in this case is: ",". We can use the `stringr::str_split` function to help us out!

- Let's see what happens when we change our delimiter from a space to a comma and then a space:

```
stringr::str_split(baseball$Name,", ")[1:2]
```

```
## [[1]]
## [1] "Allanson" "Andy"
##
## [[2]]
## [1] "Ashby" "Alan"
```

String Variable Basics

- Okay, cool! So what we get is a list where each element is a character vector where all of the characters to the left of the delimiter is considered the last name and all of the characters to the right of the delimiter is considered the first name.

- Having this structure in place (and understanding it) allows us to arrive at our final goal of separating first name and last name into individual columns.
  - Let's take in the code to see how to accomplish this!

String Variable Basics

- Well what if we wanted to do the opposite? What if we wanted to take these two individual columns and then concatenate them into one column?

- Here, we can use a couple of different functions to help us out, including our old friend `paste`! Let's take a look!

String Variable Basics

- What happens when we have a different scenario? Let's take a look at the agents.xlsx dataset:

```
agents <- readxl::read_xlsx("agents.xlsx")
knitr::kable(agents %>% dplyr::select(-Agency,-ID))
```

LastName	FirstName	MiddleName
CICHOCK	ELIZABETH	MARIE
BENINCASA	HANNAH	LEE
SHERE	BRIAN	THOMAS
HODNOFF	RICHARD	LEE

String Variable Basics

- Let's concatenate the names using the following form: lastname, firstname middlename

```
agents$agent_name <- str_c(agents$LastName,", ",
                           agents$FirstName," ",
                           agents$MiddleName)
knitr::kable(agents %>% dplyr::select(agent_name))
```

agent_name
CICHOCK, ELIZABETH MARIE
BENINCASA, HANNAH LEE
SHERE, BRIAN THOMAS
HODNOFF, RICHARD LEE

String Variable Basics

- Well, we know that it is also common to include just the middle initial rather than the full middle name. So how do we extract just an element of a text string?

- We can use two different functions: `str_sub` and `substr`
  - Let's go to the code to see what to do!

String Variable Basics

- Now, suppose that we just want to know whether a particular substring of text is within a full string. We don't need to substitute or do anything like that.

- For example, consider the following string:

```
x <- c("apple", "banana", "pear")
knitr::kable(x,
             col.names="Fruits")
```

Fruits
apple
banana
pear

String Variable Basics

- Now suppose we want to know if any of these words contain the letter "e"

- To do this we can use the function, `str_detect`

```
x <- c("apple", "banana", "pear")
stringr::str_detect(x,"e")
```

```
## [1]  TRUE FALSE  TRUE
```