How can I format a string containing a date into R “Date” object?

Authored by

stats writer

June 30, 2024

The process of formatting a string containing a date into an R "Date" object involves converting the string into a specific format that can be recognized and manipulated by the R programming language. This can be achieved by using the built-in functions and methods in R, such as strptime() and as.Date(). These functions allow you to specify the format of the date string and convert it into a standardized "Date" object that can be used for various calculations and operations in R. By following the proper syntax and specifying the correct format, you can easily convert a date string into an R "Date" object.

How can I format a string containing a date into R "Date" object? | R FAQ

Converting dates entered as strings into numeric dates in R is simple for a single string, similarly simple for a vector of strings if the date information is represented consistently, and a little trickier if the date information is not represented consistently. Below, we define strings and vectors of strings for each of these scenarios.

Single date string:

sdate1 <- "6aug2005"
sdate2 <- "jan151999"
sdate3 <- "12-15-2001"

Vector with consistent date formats:

```
dates1 <- c("06sep2001","21jun2004","04jul2006","6aug2005")
```

Vector with inconsistent date formats:

```
dates2 <- c("20jan07","06sep01","6august05","6august2005","20june1999")
```

In these examples, we can see that date strings can vary in

just to name a few. Once R formats a string as a "Date" object, it prints the date as "(4-digit year)-(2-digit month)-(2-digit day)". We will see this below.

Case 1: Formatting a single date

When looking at a single date string, one can simply identify how month, day, and year are represented in the one instance and use those in the R function

as.Date (probably after looking at the long list of date representations in
the strptime help file).
The arguments given to as.Date are the date string and then, in
quotations, the format in which the date appears within the string. Spaces,
slashes, and hyphens should be placed between the month, day, and year formats
as they appear in the string.

ndate1 <- as.Date(sdate1, "%d%B%Y"); ndate1
"2005-08-06"

class(ndate1)
"Date"

ndate2 <- as.Date(sdate2, "%B%d%Y"); ndate2
"1999-01-15"

ndate3 <- as.Date(sdate3, "%m-%d-%Y"); ndate3
"2001-12-15"

Case 2: Formatting a vector with consistent date formats

If you can determine that the date formats in a vector

are consistent, you
can simply identify how month, day, and year are represented and use the
appropriate arguments in as.Date.

ndates1 <- as.Date(dates1, "%d%B%Y"); ndates1
"2001-09-06" "2004-06-21" "2006-07-04"
"2005-08-06"class(ndates1) "Date"

Case 3: Formatting a vector with inconsistent date formats

If you can see that there are multiple date formats present in a vector, you
can indicate some date formats in R that are a bit flexible and then use some
data checks to see which dates have been read in properly.

In our vector dates2, we can start by specifying a date format that appears
appropriate for the first few dates including %d, which allows for days
within a month to optionally have a leading zero when less than 10, and %b,
which can match either an entire or an abbreviate

month name. Unfortunately, we
have some years that appear as 2 digits and some that appear as 4. As a
result, we can see in our results that the first three dates are correct and the
last two are not.

```
ndates2 <- as.Date(dates2, "%d%b%y"); ndates2
"2007-01-20" "2001-09-06" "2005-08-06" "2020-08-06" "2019-06-20"
```

We can see in the last two dates of our vector, R was expecting a two digit
date and so it only read the first two digits of the date portion of these
strings ("19" and "20") and assumed we referred to 2019 and 2020. In the
loop below, we go through our vector of numeric dates and see if any appear
later than 2018. For these, we assume the date is presented with 4 digits and
reread the string with the appropriate format.

```
for (i in 1:length(ndates2)){
```

```
if ((ndates2) > as.Date("2018-01-01")){
ndates2 <- as.Date(dates2, "%d%b%Y")
}
}

ndates2
"2007-01-20" "2001-09-06" "2005-08-06" "2005-08-06"
"1999-06-20"
```

The code used here cannot be applied to many situations, but the steps we used can be: