

# Dealing with dates

Andrew Ba Tran

## Contents

Reading dates . . . . .	2
Accessing date parts . . . . .	2
Date arithmetic . . . . .	3
Your turn . . . . .	3

This is from the third chapter of [learn.r-journalism.com](http://learn.r-journalism.com).

Dates come in as characters, most of the time.

You'll need to convert them into a date variable

We'll be using the **lubridate** package.

Here's an example of a **character variable** that might be in a data frame.

```
some_date <- "12-31-1999"
```

Convert that date into a **date variable** with the function **mdy()**

```
# If you don't have lubridate installed yet uncomment the line below and run it  
#install.packages("lubridate")
```

```
# NOTE: IF YOU GET AN ERROR ABOUTZ NOT HAVING A PACKAGE CALLED stringi  
# UNCOMMENT AND RUN THE LINES BELOW IF YOU HAVE A WINDOWS MACHINE
```

```
#install.packages("glue", type="win.binary")  
#install.packages("stringi", type="win.binary")  
#install.packages("stringr", type="win.binary")  
#install.packages("lubridate", type="win.binary")
```

```
# UNCOMMENT AND RUN THE LINES BELOW IF YOU HAVE A MAC MACHINE
```

```
#install.packages("glue", type="mac.binary")  
#install.packages("stringi", type="mac.binary")  
#install.packages("stringr", type="mac.binary")  
#install.packages("lubridate", type="mac.binary")
```

```
library(lubridate)
```

```
mdy(some_date)
```

```
## [1] "1999-12-31"
```

The **mdy()** function is very versatile. It stand for month-date-year.

And it'll be able to parse any version of that (with slashes or commas, or dashes) as long as that's the order of the information.

Check it out:

```
data <- data.frame(First=c("Charlie", "Lucy", "Peppermint"),  
                  Last=c("Brown", "van Pelt", "Patty"),
```

```

        birthday=c("10-31-06", "2/4/2007", "June 1, 2005"))

data$DOB <- mdy(data$birthday)

```

```

data
##      First      Last      birthday      DOB
## 1   Charlie    Brown    10-31-06 2006-10-31
## 2     Lucy van Pelt    2/4/2007 2007-02-04
## 3 Peppermint    Patty June 1, 2005 2005-06-01

```

## Reading dates

Order of elements in date-time	Parse function
year, month, day	ymd()
year, day, month	ydm()
month, day, year	mdy()
day, month, year	dmy()
hour, minute	hm()
hour, minute, second	hms()
year, month, day, hour, minute, second	ymd_hms()

## Accessing date parts

Date component	Function
Year	year()
Month	month()
Week	week()
Day of year	yday()
Day of month	mday()
Day of week	wday()
Hour	hour()
Minute	minute()
Second	ymd_hms()
Time zone	ymd_hms()

Now that we have the date in the right format, we can extract data from it with the functions above.

```

data$year <- year(data$DOB)
data$month <- month(data$DOB, label=TRUE)
data$day <- day(data$DOB)
data$weekday <- wday(data$DOB, label=TRUE, abbr=FALSE)

```

```

data
##      First      Last      birthday      DOB year month day  weekday
## 1   Charlie    Brown    10-31-06 2006-10-31 2006   Oct  31   Tuesday
## 2     Lucy van Pelt    2/4/2007 2007-02-04 2007   Feb   4    Sunday
## 3 Peppermint    Patty June 1, 2005 2005-06-01 2005   Jun   1   Wednesday

```

## Date arithmetic

The function `difftime()` extracts the number of days between two dates that are passed to it

```
# We're going to use the now() function which brings in the date for today
```

```
today <- now()
data$age <- difftime(today, data$DOB)
```

```
data
```

```
##      First      Last      birthday      DOB year month day  weekday
## 1   Charlie    Brown    10-31-06 2006-10-31 2006   Oct  31   Tuesday
## 2     Lucy van Pelt    2/4/2007 2007-02-04 2007   Feb   4     Sunday
## 3 Peppermint    Patty June 1, 2005 2005-06-01 2005   Jun   1 Wednesday
##           age
## 1 4283.185 days
## 2 4187.185 days
## 3 4800.185 days
```

And how does that translate into years?

With some math. We'll have to turn the column into a number, first.

```
data$age_years <- as.numeric(data$age) / 365.25 #.25 because of leap years
```

```
data
```

```
##      First      Last      birthday      DOB year month day  weekday
## 1   Charlie    Brown    10-31-06 2006-10-31 2006   Oct  31   Tuesday
## 2     Lucy van Pelt    2/4/2007 2007-02-04 2007   Feb   4     Sunday
## 3 Peppermint    Patty June 1, 2005 2005-06-01 2005   Jun   1 Wednesday
##           age age_years
## 1 4283.185 days  11.72672
## 2 4187.185 days  11.46389
## 3 4800.185 days  13.14219
```

That's a pretty good start for now. To see more functions and examples, check out the vignette for **lubridate**.

---

## 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.