Post 1

Ashley Chien

October 16, 2017

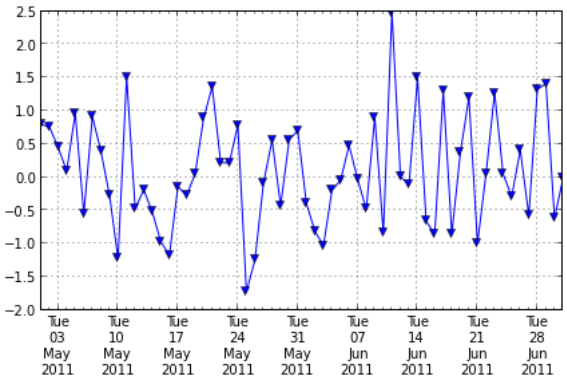
Introduction

Major Theme: Data Manipulation

Specific Topic: Lubridate (A way to work with datetimes in R)

About Lubridate and Motivation

For my first post, I am exploring the data manipulation package of Lubridate, which makes it easier to work with dates and times. The reason I chose to look at Lubridate is because analyzing time series data seems to be important, and timestamps are often messy, and non-uniform, so I thought working with dates could be a useful skill to have.



Example of a graph with time involved

Basics of Lubridate

First, make sure you download Lubridate. I've already downloaded it, so let's load it it now.

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.4.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

The way you can use lubridate is super intuitive. The function names correspond to the order of the year, month, and day. For example, if your date is in the order Month, Day, Year, you would take the letters of Month, Day, Year, or (MDY), and use that function.

```
# A date in a string that we wish to parse
random_date = "10172017"
parsed = mdy(random_date)
parsed
```

```
## [1] "2017-10-17"
```

If there is also information about the time, you can just include to the function (h/m/s). Here is an example:

```
# This string has both time and date
date_and_time = "10-17-2017 7:34:02"
converted = mdy_hms(date_and_time)
converted
```

```
## [1] "2017-10-17 07:34:02 UTC"
```

In order to extract information from date-times, you use the second, minute, hour, etc. function. You can also change the content of the dates in this manner.

```
# How to change the value of something
sample = "11-07-2017"
wrong_year = mdy(sample)
print (year(wrong_year))
```

```
## [1] 2017
```

```
year(wrong_year) <- 1998
wrong_year
```

```
## [1] "1998-11-07"
```

However, sometimes, we do not want to change the actual object. Lubridate provides the update method for this purpose, so we can save the modified datetime into some other variable.

```
# Example of using update
first = "01-01-2001"
first_parsed = mdy(first)

edited = update(first_parsed, year=2000)

print(edited)
```

```
## [1] "2000-01-01"
```

```
print(first_parsed)
```

```
## [1] "2001-01-01"
```

Versatility of Lubridate

Lubridate can be used with a variety of common date-time and time series objects, such as charcter strings, chron, timeDate,.., etc. Let's see an example with the base R way of manipulating dates.

```
# Example using base R Date object
d8 <- as.Date('1915-6-16')
year(d8)
```

```
## [1] 1915
```

In this example, we used the Lubridate method on a DateTime object we made in base R! Amazing!

Arithmetic with Lubridate

Arithmetic is actually pretty complicated with DateTimes, so there are four different classes in lubridate to handle all of the different situations. First, let's start off with an instant.

Instant

An instant is a specific moment in time. Every time we parse a date, we create an insant.

```
# Example of an instant
instant = mdy("10-17-2017")
```

Interval

An interval is a span of time between two instants. We can make an interval by either subtracting two instants or using the function new_interval().

```
independence = ymd_hms("1776-07-04 12:00:00")
birthday = ymd_hms("1826-07-04 12:00:00")

# interval = now - independence # Getting an interval is easy!
```

Duration

A duration is a generic length of time. It is an exact length. We create durations using new_duration or dminutes, dyears(), etc.

```
# Example of duration
ten_minutes = dminutes(10)
ten_seconds = dseconds(10)
```

Periods

Periods no longer have consistent lengths. They are different from durations. An example of no longer having consistent lengths is that the length of 2 months depends on when the 2 months starts. Here's how we make periods.

```
# Example of periods
ten_minutes = minutes(10)
ten_seconds = seconds(10)
```

Example of Usefulness

Let's say we have a flight from LAX to LGA. The flight takes off at 8 AM (PST) in Los Angeles, the flight is 5 hours long, and we want to know at what time (EST), we will arrive in New York. We can do that easily with lubridate.

```
# Example of adding hours
thing = ymd_hms("2017-10-17 8:00:00")
thing = thing + hours(5)
arrival = as.POSIXct(thing, tz="America/Los_Angeles")

with_tz(arrival, "EST")
```

```
## [1] "2017-10-17 08:00:00 EST"
```

Message

Lubridate is a super useful library when you are working with date-times. You should learn to use the library to use it to the maximum usefulness. These are just some examples of the uses of lubridate! There are many more to explore!

References:

[1] https://cran.r-project.org/web/packages/lubridate/vignettes/lubridate.html

[2] https://stackoverflow.com/questions/8765621/length-of-lubridate-interval

[3] Grolemund, G., & Wickham, H. (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1 - 25. doi:http://dx.doi.org/10.18637/jss.v040.i03

[4] https://rpubs.com/davoodastaraky/lubridate

[5] http://data.library.virginia.edu/working-with-dates-and-time-in-r-using-the-lubridate-package/

[6] https://stackoverflow.com/questions/34189367/find-day-of-year-with-the-lubridate-package-in-r

[7] https://stackoverflow.com/questions/16932132/r-sequence-of-dates-with-lubridate