Dates and Times in R

Biostatistics 140.776

Dates and Times in R

R has developed a special representation of dates and times

- Dates are represented by the `Date` class
- Times are represented by the `POSIXct` or the `POSIXlt` class
- Dates are stored internally as the number of days since 1970-01-01
- Times are stored internally as the number of seconds since 1970-01-01

The `lubridate` package

- The `lubridate` package is a very useful package for dealing with all the little annoying aspects of dates/times
- Largely replaces the default date/time functions in base R
- Methods for date/time arithmetic
- Handles time zones, leap year, leap seconds, etc.

Dates in R

Dates are represented by the `Date` class and can be coerced from a character string using the `ymd()` function.

```
library(lubridate)
x <- ymd("1970-01-01")
x
```

```
[1] "1970-01-01"
```

`Date` objects have their own special print methods that will always format as "YYYY-MM-DD".

Alternate Formulations

Different locales have different ways formatting dates

```
ymd("2016-09-13")  ## International standard
```

```
[1] "2016-09-13"
```

```
mdy("09-13-2016")  ## Mostly U.S.
```

```
[1] "2016-09-13"
```

```
dmy("13-09-2016")  ## Europe
```

```
[1] "2016-09-13"
```

All of the above are valid and lead to the exact same object.

Times in R

Times are represented using the `POSIXct` or the `POSIXlt` class

- `POSIXct` is just a very large integer under the hood; it is a useful class when you want to store times in something like a data frame
- `POSIXlt` is a list underneath and it stores a bunch of other useful information like the day of the week, day of the year, month, day of the month

There are a number of generic functions that work on dates and times

- `weekdays`: give the day of the week
- `month`: give the month name (possibly abbreviated)
- `quarter`: give the quarter number (1, 2, 3, 4)

Times in R

Times can be coerced from a character string with `ymd_hms()`

```
ymd_hms("2016-09-13 14:00:00")
```

```
[1] "2016-09-13 14:00:00 UTC"
```

```
ymd_hms("2016-09-13 14:00:00", tz = "America/New_York")
```

```
[1] "2016-09-13 14:00:00 EDT"
```

```
ymd_hms("2016-09-13 14:00:00", tz = "")
```

```
[1] "2016-09-13 14:00:00 EDT"
```

Time Zones!

Time zones were created to make your data analyses more difficult.

- `ymd_hms()` function will by default use UTC as the time zone
- Specifying `tz = ""` will use the local time zone
- Better to specify time zone when possible to avoid ambiguity

You can go to Wikipedia to find the list of time zones
http://goo.gl/xJ8K6q

Specifying Times in R

Finally, there is the `strptime()` function in case your dates are written in a different format

```
datestring <- c("January 10, 2012 10:40",
                "December 9, 2011 9:10")
x <- strptime(datestring, "%B %d, %Y %H:%M",
              tz = "America/Los_Angeles")
x
```

```
[1] "2012-01-10 10:40:00 PST"
[2] "2011-12-09 09:10:00 PST"
```

- Check `?strptime` for details of formatting strings
- When reading in data with `read_csv()`, you may need to read in as character first and then convert to date/time

Operations on Dates and Times

You can add and subtract dates and times. You can do comparisons too (i.e. ==, <=)

```
x <- ymd("2012-01-01", tz = "")  ## Midnight
y <- dmy_hms("9 Jan 2011 11:34:21", tz = "")
x - y
```

```
Time difference of 356.5178 days
```

```
x + y  ## Nope!
```

```
Error in `+.POSIXt`(x, y): binary '+' is not defined for "P
```

Operations on Dates and Times

```
y + 1  ## Add a number to a time
```

```
[1] "2011-01-09 11:34:22 EST"
```

```
y <- date(y)  ## Just keep the date portion
y
```

```
[1] "2011-01-09"
```

```
y + 1  ## Add a number to a date
```

```
[1] "2011-01-10"
```

Operations on Dates and Times

Even keeps track of leap years, leap seconds, daylight savings, and time zones.

```
x <- ymd("2012-03-01")
y <- ymd("2012-02-28")
x - y

Time difference of 2 days

x <- ymd_hms("2012-10-25 01:00:00", tz = "")
y <- ymd_hms("2012-10-25 06:00:00", tz = "GMT")
y - x

Time difference of 1 hours
```

Extracting Elements of Dates/Times

There are a set of helper functions in `lubridate` that can extract sub-elements of dates/times

```
x <- ymd_hms(c("2012-10-25 01:13:46",
               "2015-04-23 15:11:23"), tz = "")
year(x)
```

```
[1] 2012 2015
```

```
month(x)
```

```
[1] 10  4
```

```
day(x)
```

```
[1] 25 23
```

```
weekdays(x)
```

```
[1] "Thursday" "Thursday"
```

Extracting Elements of Dates/Times

```
minute(x)
```

```
[1] 13 11
```

```
second(x)
```

```
[1] 46 23
```

```
hour(x)
```

```
[1]  1 15
```

```
week(x)
```

```
[1] 43 17
```

Summary

- Dates and times have special classes in R that allow for numerical and statistical calculations
- Dates use the `Date` class
- Times use the `POSIXct` and `POSIXlt` class
- Character strings can be coerced to Date/Time classes using the `ymd()` and `ymd_hms()` functions. In strange cases, you can use the `strptime()`or the `as.Date()` functions.
- The `lubridate` package is essential for manipulating date/time data