Northeastern University

CS6020: Collecting, Storing, and Retrieving Information

Basic Data Shaping

Basic Data Shaping

DATE AND TIME PROCESSING

Lesson Objectives

- After completing this lesson, you are able to:
  - process date and time values
  - locate and install R packages
  - install the ***lubridate*** package for date and time processing
  - process dates using ***lubridate***

Data Manipulation

DATE PROCESSING

Dates

- Dates are an important data format and require special processing.
- R has built-in functions to deal with date values encoded in different formats.
- The built-in `as.date()` function can only handle dates but not time values.
- Better support for date processing is provided by the ***lubridate*** package.

R Date Object

- The base object `date` stores dates internally as number of days elapsed since January 1, 1970.

```
> as.Date("1970-01-01")
[1] "1970-01-01"
> as.numeric(as.Date("1970-01-02"))
[1] 1
> as.numeric(as.Date("1970-01-01"))
[1] 0
```

Specifying Date Format

- By default, the date object requires a string is the format “YYYY-MM-DD”, but other format specification are possible:

```
> d<-as.Date("Dec-27-2014",format="%b-%d-%Y")
> as.numeric(d)
[1] 16431
```

- Use `?strptime` to get a full description of date formatting strings.

Date Parsing Specification

String	Date Part	Example
%m	two digit month	10
%d	two digit date	31
%Y	four digit date	2014
%y	two digit year (assume 20 prefix)	14
%B	full month	December
%b	abbreviated month	Dec

Additional Date Parsing Examples

```
> dt1 <- as.Date("12/29/2014", format="%m/%d/%Y")
> dt1
[1] "2014-12-29"

> dt2 <- as.Date("Oct-11-14", format="%b-%d-%y")
> dt2
[1] "2014-10-11"

> dt3 <- as.Date("17.12.14", format="%d.%m.%y")
> dt3
[1] "2014-12-17"

> dt4 <- as.Date("December 17, 2014", format="%B %d, %Y")
> dt4
[1] "2014-12-17"
```

Calculation with Dates

- R supports calculations on dates, including differences between dates, adding and subtracting days and dates.

```
> dt1
[1] "2014-12-29"
> dt2
[1] "2014-10-11"
> dt1-dt2
Time difference of 79 days
> dt1 + 10
[1] "2015-01-08"
```

Date Vectors

- R supports vectors (collections) of dates and can calculate the interval between them.

```
> dts <- as.Date(c("2014-06-01","2014-07-
08","2014-10-14"))
> dts
[1] "2014-06-01" "2014-07-08" "2014-10-14"
> diff(dts)
Time differences in days
[1] 37 98
```

Date Sequences

- The `seq()` function can also be applied to the construction of date sequences.

```
> dt1
[1] "2014-12-29"
> every.other.week <- seq(dt1,length=6,by=14)
> every.other.week
[1] "2014-12-29" "2015-01-12" "2015-01-26" "2015-02-09"
"2015-02-23"
[6] "2015-03-09"
```

Data Manipulation

TIME PROCESSING

The POSIXct Class for Times

- In addition to dates, times are an important data value in data science.
- Time values are intrinsically processed using the POSIXct class.
- The POSIXct class represents combined date and time value.

```
> tm1 <- as.POSIXct("2014-12-28 09:59:43")
> tm1
[1] "2014-12-28 09:59:43 EST"
```

Time Parsing Specification

String	Time Part	Example
%H	hour of the day	08
%M	minutes past the hour	44
%S	seconds past the minute	12

Time Parsing Examples

```
> tm2 <- as.POSIXct("2014-12-19 11:38:42", tz="GMT")
> tm2
[1] "2014-12-19 11:38:42 GMT"

> tm3 <- as.POSIXct("25072013 08:32:07", format = "%d%m%Y
%H:%M:%S")
> tm3
[1] "2013-07-25 08:32:07 EDT"
```

Time Calculations

- Time values can be compared, adding, and subtracted.

```
> tm1 <- as.POSIXct(tm1 <- as.POSIXct("2013-07-24 23:55:26")>
> tm1 <- as.POSIXct("2013-07-24 23:55:26")
> tm1
[1] "2013-07-24 23:55:26 EDT"
> tm2 <- as.POSIXct("25072013 08:32:07", format = "%d%m%Y
%H:%M:%S")
> tm2
[1] "2013-07-25 08:32:07 EDT"
> tm2 > tm1
[1] TRUE
> tm2 - tm1
Time difference of 8.611389 hours
> tm1 + 25
[1] "2013-07-24 23:55:51 EDT"
```

Current Time and Date

- To get the current time and date as a POSIXct object , use the `now()` and `Sys.time()` functions.

```
> now()
[1] "2014-12-29 10:08:47 EST"
> Sys.time()
[1] "2014-12-29 10:09:21 EST"
```

- To get the current date as a `Date` object, use `today()`.

```
> today()
[1] "2014-12-29"
> class(today())
[1] "Date"
```

The `POSIXlt` Class for Times

- The `POSIXlt` class enables easy extraction of specific components of time.

```
> tm1.lt <- as.POSIXlt("2014-12-28 10:12")
> tm1.lt
[1] "2014-12-28 10:12:00 EST"
> tm1.lt$min
[1] 12
```

Add-On Date and Time Packages

- While the built-in time and date processing of R is often sufficient, there are several packages that simplify certain date and time processing tasks.
- Two of the most popular packages are:
  - ***lubridate***
  - ***chron***
- These are add-on packages and must be downloaded and installed before they can be used.

Data Manipulation

THE *LUBRIDATE* PACKAGE

The ***lubridate*** Package

- The ***lubridate*** package is one of several date processing add-on packages available for R.
- It provides more intuitive handling of date and time values.
- The library is a wrapper around the `POSIXct` class with more intuitive syntax.
- Using ***lubridate*** requires installation of the package first.

Installing the *lubridate* Package

- Steps to install a package in R:
  1. Select Packages/Install package(s)…
  2. Select the closest download site
  3. Choose the package “lubridate” from the list
  4. Select Packages/Install package(s) from local zip files…
  5. Select the directory into which the package was downloaded
  6. Select the package zip file
- The package is now installed and needs to be loaded using the `library(lubridate)` function

Creating Date and Time Objects

- The ***lubridate*** package has numerous functions for creating date and time objects that do not require a parse format string.

```
> tm1.lub <- ymd_hms("2014-12-28 10:26:23")
> tm1.lub
[1] "2014-12-28 10:26:23 UTC"
> tm2.lub <- mdy_hm("12/28/14 11:44")
> tm2.lub
[1] "2014-12-28 11:44:00 UTC"
> tm3.lub <- dmy_hm("28.12.14 9:30AM")
> tm3.lub
[1] "2014-12-28 09:30:00 UTC"
```

Component Extraction

- The ***lubridate*** package simplifies extraction of date and time components.

```
> tm3.lub <- dmy_hm("28.12.14 9:30AM")
> tm3.lub
[1] "2014-12-28 09:30:00 UTC"
> year(tm3.lub)
[1] 2014
> wday(tm3.lub, label=TRUE)
[1] Sun
Levels: Sun < Mon < Tues < Wed < Thurs < Fri < Sat
> hour(tm3.lub)
[1] 9
> tz(tm3.lub)
[1] "UTC"
```

Updating Date and Time

- The ***lubridate*** package also simplifies updating date and time components of a `POSIXct` object.

```
> tm3.lub <- dmy_hm("28.12.14 9:30AM")
> tm3.lub
[1] "2014-12-28 09:30:00 UTC"
> year(tm3.lub) <- 2013
> tm3.lub
[1] "2013-12-28 09:30:00 UTC"
```

Conversion to Decimal Time

- To facilitate time calculations, convert a time to a decimal value.

```
> tm3.lub
[1] "2013-12-28 09:30:00 UTC"
> tm3.dechr <- hour(tm3.lub) + minute(tm3.lub)/60 +
second(tm3.lub)/3600
> tm3.dechr
[1] 9.5
```

Summary

- In this lesson, you learned that:
  - dates and times are an important data value in R
  - R has built-in functions to deal with dates and times but that several packages are available to ease date and time processing
  - external packages must be downloaded and installed before they can be loaded

Summary, Review, & Questions…