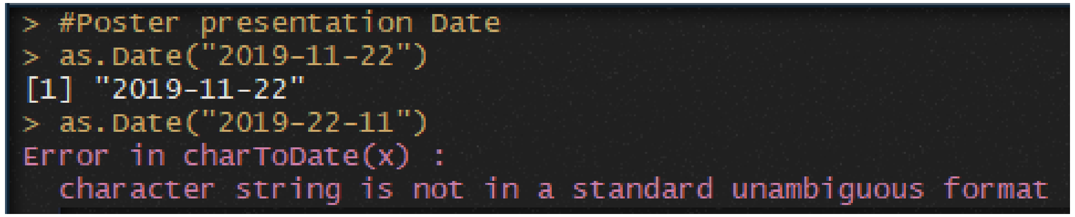
Lubridate R Package

Satyen Amonkar

IST 719

Importance of Lubridate Package



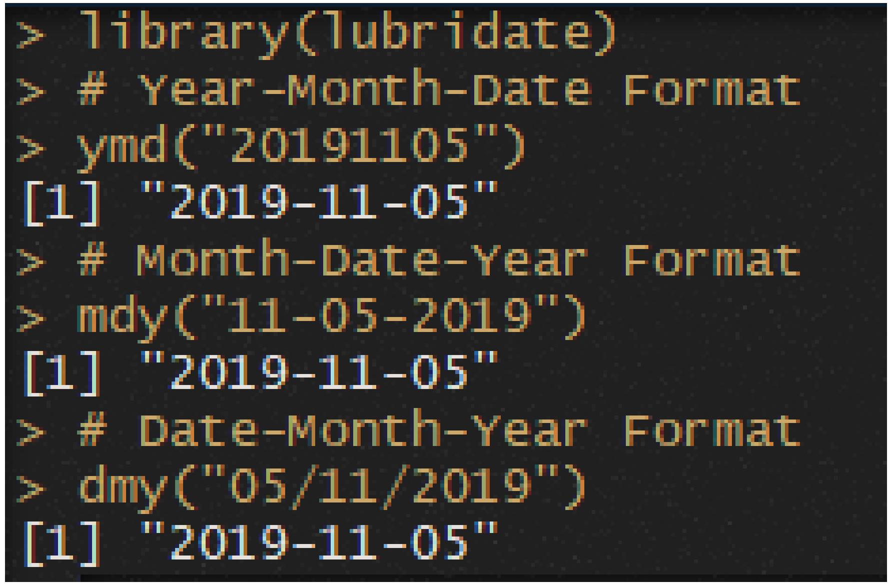
```
> #Poster presentation Date
> as.Date("2019-11-22")
[1] "2019-11-22"
> as.Date("2019-22-11")
Error in charToDate(x) :
  character string is not in a standard unambiguous format
```

- Converting string to date is a tedious task in R especially when the standard format of R is not known
  - as.Date() function can be used to convert string to date in R
  - However, the input function has to be **Year-Month-Date format** (2019-01-20), else it throws error
- Lubridate is an R package that makes it easier to work with dates and times

Common Applications of Lubridate Package

- Parsing dates and times
- Setting and Extracting information
- Instants
- Time Zones
- Timespans
- Arithmetic with date times

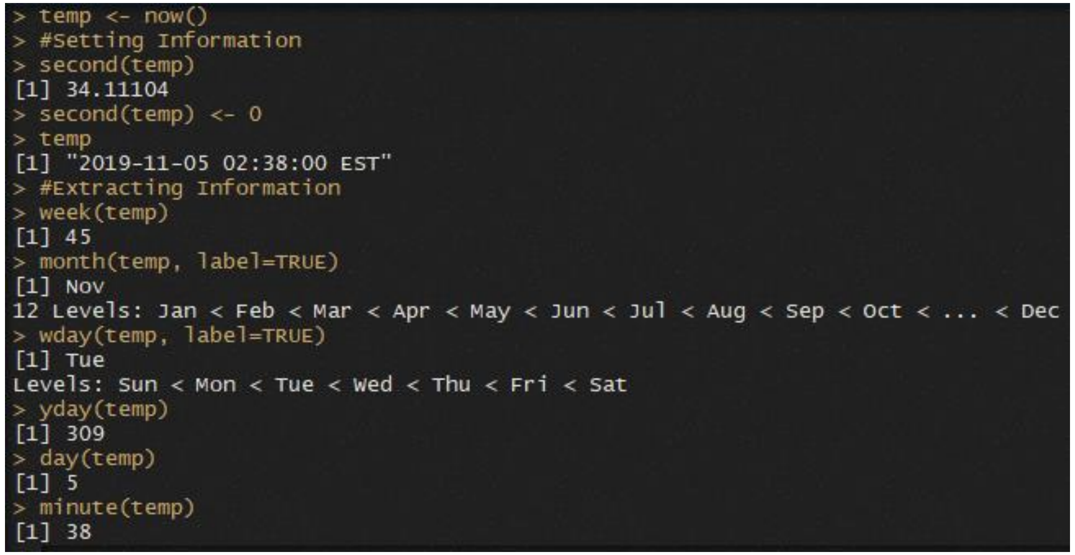
Fast Parsing of data and times



```
> library(lubridate)
> # Year-Month-Date Format
> ymd("20191105")
[1] "2019-11-05"
> # Month-Date-Year Format
> mdy("11-05-2019")
[1] "2019-11-05"
> # Date-Month-Year Format
> dmy("05/11/2019")
[1] "2019-11-05"
```

- Lubridate simplifies the function name of date formats:
  - Identify order of year, month and day that appears in the dates
  - Arrange “y”, “m”, and “d” in the same order
  - This is the name of the function in Lubridate that will parse your dates

Setting and Extracting information



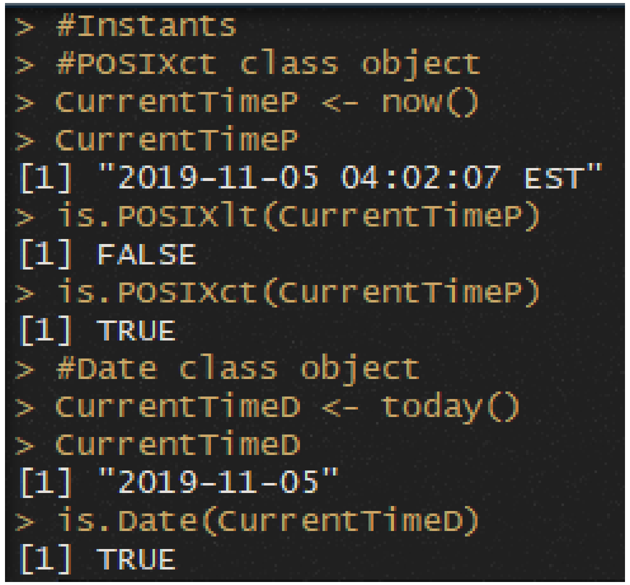
```
> temp <- now()
> #Setting Information
> second(temp)
[1] 34.11104
> second(temp) <- 0
> temp
[1] "2019-11-05 02:38:00 EST"
> #Extracting Information
> week(temp)
[1] 45
> month(temp, label=TRUE)
[1] Nov
12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < Oct < ... < Dec
> wday(temp, label=TRUE)
[1] Tue
Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
> yday(temp)
[1] 309
> day(temp)
[1] 5
> minute(temp)
[1] 38
```

- Information can be extracted from date time functions:

  second, minute, hour, day, week, month, year, wday, yday and tz

- month and wday functions have an optional argument called label, which replaces the numeric output to weekday or month

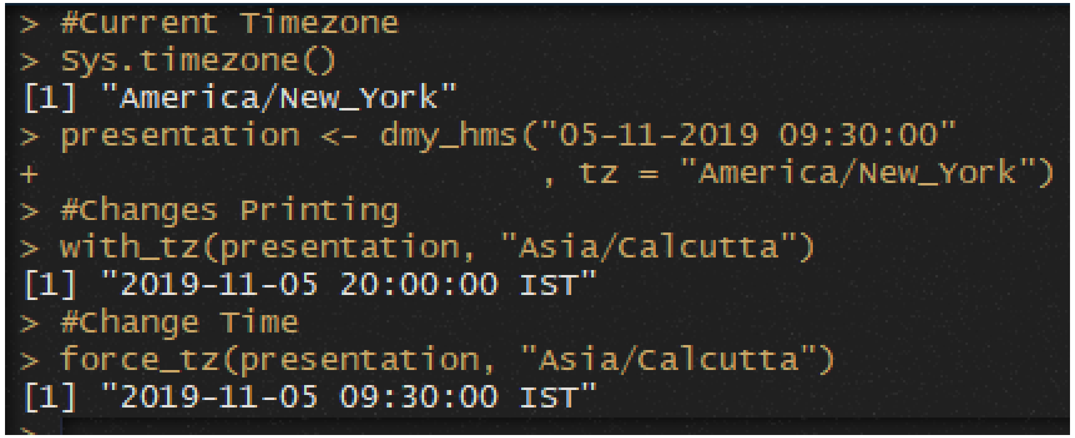
Instants



```
> #Instants
> #POSIXct class object
> CurrentTimeP <- now()
> CurrentTimeP
[1] "2019-11-05 04:02:07 EST"
> is.POSIXlt(CurrentTimeP)
[1] FALSE
> is.POSIXct(CurrentTimeP)
[1] TRUE
> #Date class object
> CurrentTimeD <- today()
> CurrentTimeD
[1] "2019-11-05"
> is.Date(CurrentTimeD)
[1] TRUE
```

- Instants are **specific moments of time**.
- Date, POSIXct, and POSIXlt are its three object classes

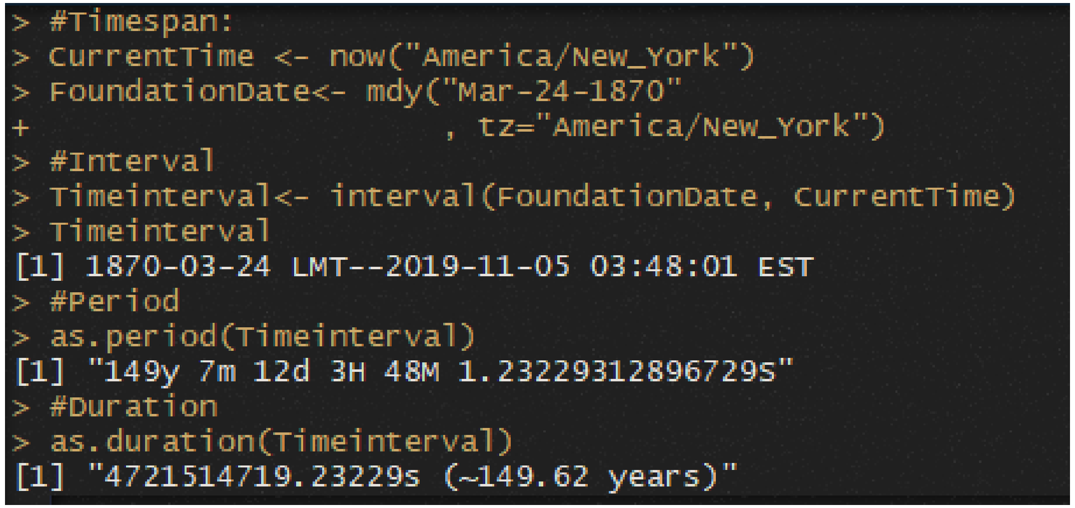
Time Zones



```
> #Current Timezone
> Sys.timezone()
[1] "America/New_York"
> presentation <- dmy_hms("05-11-2019 09:30:00"
+                         , tz = "America/New_York")
> #Changes Printing
> with_tz(presentation, "Asia/Calcutta")
[1] "2019-11-05 20:00:00 IST"
> #Change Time
> force_tz(presentation, "Asia/Calcutta")
[1] "2019-11-05 09:30:00 IST"
>
```

- Lubridate provides **two helper functions** for working with **time zones**.
    - with_tz: changes the time zone in which an instant is displayed. The clock time displayed for the instant changes, but the **moment of time** described **remains the same**
    - Force_tz: changes only the time zone element of an instant. The clock time displayed remains the same, but the **resulting instant describes a new moment** of time

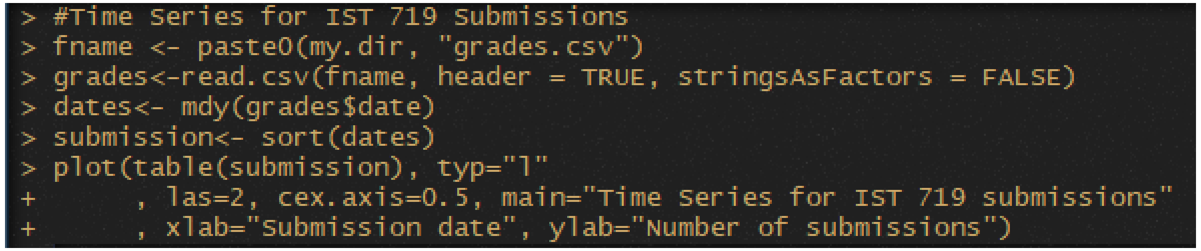
Timespans



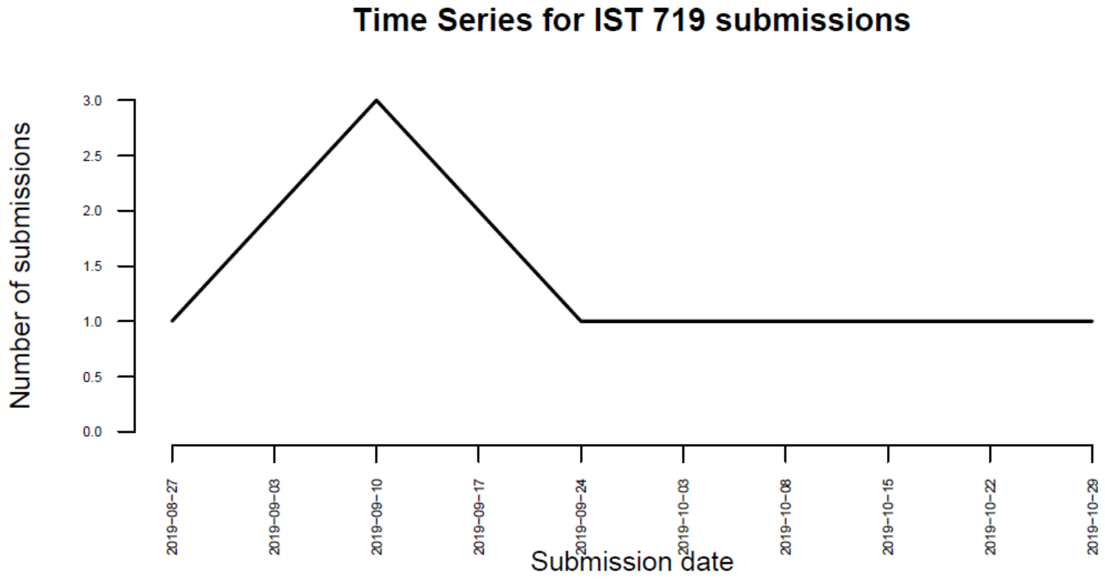
```
> #Timespan:
> CurrentTime <- now("America/New_York")
> FoundationDate<- mdy("Mar-24-1870"
+                      , tz="America/New_York")
> #Interval
> Timeinterval<- interval(FoundationDate, CurrentTime)
> Timeinterval
[1] 1870-03-24 LMT--2019-11-05 03:48:01 EST
> #Period
> as.period(Timeinterval)
[1] "149y 7m 12d 3H 48M 1.23229312896729S"
> #Duration
> as.duration(Timeinterval)
[1] "4721514719.23229s (~149.62 years)"
```

- Lubridate provides three additional classes to better **navigate nuances of time**:
  - **Interval**: summary of the time information between two points
  - **Period**:   accurately track clock times despite leap years, leap seconds, and day light savings time
  - **Duration**: measure the exact amount of time between two points

Time Series for IST 719 Graded Submissions



```
> #Time Series for IST 719 Submissions
> fname <- paste0(my.dir, "grades.csv")
> grades<-read.csv(fname, header = TRUE, stringsAsFactors = FALSE)
> dates<- mdy(grades$date)
> submission<- sort(dates)
> plot(table(submission), typ="l"
+      , las=2, cex.axis=0.5, main="Time Series for IST 719 submissions"
+      , xlab="Submission date", ylab="Number of submissions")
```



Thank You!

Satyen Amonkar