

Task	lubridate	Date	POSIXct
now (system time zone)	now()		Sys.time()
now (GMT)	now("GMT")	Sys.Date()	
origin	origin	structure(0, class = "Date")	structure(0, class = c("POSIXt", "POSIXct"))
x days since origin	origin + days(x)	structure(floor(x), class = "Date")	structure(x*24*60*60, class=c("POSIXt", "POSIXct"))
next day	date + days(1)	date + 1	seq(date, length = 2, by = "day")[2]
previous day	date - days(1)	date - 1	seq(date, length = 2, by = "-1 day")[2]
<i>DST and time zones</i>			
x days since date (day exactly 24 hours)	date + ddays(x)		seq(date, length = 2, by = paste(x, "day"))[2]
(allowing for DST)	date + days(x)	date + floor(x)	seq(date, length = 2, by = paste(x, "DSTday"))[2]
display date in new time zone	with_tz(date, "TZ")		as.POSIXct(format(as.POSIXct(date), tz = "TZ"), tz = "TZ")
keep clock time, replace time zone	force_tz(date, tz = "TZ")		
<i>Exploring</i>			
sequence	date + c(0:9) * days(1)	seq(date, length = 10, by = "day")	seq(date, length = 10, by = "DSTday")
every 2nd week	date + c(0:2) * weeks(2)	seq(date, length = 3, by = "2 week")	seq(date, length = 3, by = "2 week")
first day of month	floor_date(date, "month")	as.Date(format(date, "%Y-%m-01"))	as.POSIXct(format(date, "%Y-%m-01"))
round to nearest first of month	round_date(date, "month")		
extract year value	year(date)	as.numeric(format(date, "%Y"))	as.numeric(format(date, "%Y"))
change year value	year(date) <- z	as.Date(format(date, "z-%m-%d"))	as.POSIXct(format(date, "z-%m-%d"))
day of week	wday(date) # Sun = 1	as.numeric(format(date, "%w")) # Sun = 0	as.numeric(format(date, "%w")) # Sun = 0
day of year	yday(date)	as.numeric(format(date, "%j"))	as.numeric(format(date, "%j"))
express as decimal of year	decimal_date(date)		
<i>Parsing dates</i>			
z = "1970-10-15"	ymd(z)	as.Date(z)	as.POSIXct(z)
z = "10/15/1970"	mdy(z)	as.Date(z, "%m/%d/%Y")	as.POSIXct(strptime(z, "%m/%d/%Y"))
z = 15101970	dmy(z)	as.Date(as.character(z), format = "%d%m%Y")	as.POSIXct(as.character(z), tz = "GMT", format = "%d%m%Y")
<i>Durations comparison</i>			
Duration	lubridate	Base R	
1 second	seconds(1)	as.difftime(1, unit = "secs")	
5 days, 3 hours and - 1 minute	new_duration(day = 5, hour = 3, minute = -1)	as.difftime(60 * 24 * 5 + 60 * 3 - 1, unit = "mins") # Time difference of 7379 mins	
1 month	months(1)		
1 year	years(1)		

Table 3: **lubridate** provides a simple alternative for many date and time related operations. Table adapted from [Grothendieck and Petzoldt \(2004\)](#).