Dates and Times and Date-times (oh my!)

Grinnell College

March 3, 2026
2026-03-03
2026 Mar 3
3/3/26

What is time?

There are (typically) three ways we will refer to time:

- With **dates**, indicating a date (2026-03-03)
- With **time**, indicating time (08:30:00)
- With **date-time**, uniquely identifying an instant in time (2026-03-03 08:30:00 UTC)

Difficulties

1. Non-standard format
2. Missing or malformed values
3. Time Zones
4. Daylight savings
5. Leap years

Strategy

Our goals are not to learn all the syntax (it never is)

Instead, we need to orient ourselves with ideas around:

- Parse
- Extract
- Store
- Manipulate

Parsing Dates

3/2/2026 or 2/3/2026?

We get around this by being explicit

```
> mdy("3/2/26")
[1] "2026-03-02"

> dmy("3/2/26")
[1] "2026-02-03"
```

This is dangerous, but lubridate will do it's best

```
> mdy("Today is January 1st, 2020")
[1] "2020-01-01"
```

Constructing and Extracting Dates

We can also directly construct dates using *constructor* functions

```
> make_datetime(year = 1970, month = 1, day = 1)
[1] "1970-01-01 UTC"
```

Likewise, extraction functions allow us to isolate individuals components of a date

```
> year("1970-01-01 UTC")
[1] 1970
```

This is of course more useful if you are constructing dates out of columns where the arguments for year, month, and day are columns in a date.frame

Representation

How we see a date ("1970-01-01" vs "01-01-1970") is separate than how a date or time object is stored in R. There are a few types in base R (no packages)

```
> Sys.time()
[1] "2026-03-03 07:37:09 CST"
> Sys.time() %>% class()
[1] "POSIXct" "POSIXt"
>
> today()
[1] "2026-03-03"
> today() %>% class()
[1] "Date"
```

Base R Dates cont.

Underneath all date or time objects in R is a numeric reference to time since Jan 1, 1970

```
## Days since 1/1/1970
> today() %>% as.numeric()
[1] 20515

## Seconds since 1/1/1970
> Sys.time() %>% as.numeric()
[1] 1772545179
```

Having a fixed point of reference makes it possible to do arithmetic

```
> today()
[1] "2026-03-03"
> today() - 30
[1] "2026-02-01"
```

More arithmetic

While subtracting an integer from a date will give us a date, we can also subtract dates from dates. This gives us a `difftime` object

```
> today() - (today() - 10)
Time difference of 10 days
>
> (today() - (today() - 10)) %>% class()
[1] "difftime"
```

Under the hood, though, this is still represented as a numeric

```
> (today() - (today() - 10)) %>% as.numeric()
[1] 10
```

Time Spans

There are three other ways mentioned in the text that time, and in particular time spans, can be represented with the help of lubridate

1. Durations, representing a span of time in seconds
2. Periods, representing a span of time in “human units” like weeks or months
3. Intervals, with start and end dates