ymd()

- 27th of February 2013
- `ymd()` - year, then month, then day

```
ymd("2013-02-27")
```

```
"2013-02-27"
```

```
ymd("2013.02.27")
```

```
"2013-02-27"
```

```
ymd("2013 Feb 27th")
```

```
"2013-02-27"
```

Friends of ymd()

`ymd()`, `ydm()`, `mdy()`, `myd()`, `dmy()`, `dym()`

```
dmy("27-02-2013")
```

```
"2013-02-27"
```

```
mdy("02-27-2013")
```

```
"2013-02-27"
```

```
dmy_hm("27-02-2013 12:12pm")
```

```
"2013-02-27 12:12:00 UTC"
```

parse_date_time(x = ___, order = ___)

```
parse_date_time("27-02-2013", order = "dmy")
```

```
"2013-02-27 UTC"
```

```
parse_date_time(c("27-02-2013", "2013 Feb 27th"),
+     order = c("dmy", "ymd"))
```

```
"2013-02-27 UTC" "2013-02-27 UTC"
```

Formatting characters

Character	Meaning
`d`	Numeric day of the month
`m`	Month of year
`y`	Year with century
`Y`	Year without century
`H`	Hours (24 hour)
`M`	Minutes
`S`	Seconds

Character	Meaning
`a`	Abbreviated weekday
`A`	Full weekday
`b`	Abbreviate month name
`B`	Full month name
`I`	Hours (12 hour)
`p`	AM/PM
`z`	Timezone, o set from UTC

make_date(year, month, day)

```
make_date(year = 2013, month = 2, day = 27)
```

```
"2013-02-27"
```

`make_datetime(year, month, day, hour, min, sec)` for datetimes

Extracting parts of a datetime

```
x <- ymd("2013-02-23")
```

```
year(x)
```

```
2013
```

```
month(x)
```

```
2
```

```
day(x)
```

```
23
```

Extracting parts of a datetime

Function	Extracts
`year()`	Year with century
`month()`	Month (1-12)
`day()`	Day of month (1-31)
`hour()`	Hour (0-23)
`min()`	Minute (0-59)
`second()`	Second (0-59)
`wday()`	Weekday (1-7)
`yday()`	Day of year a.k.a. Julian day (1-366)
`tz()`	Timezone

Setting parts of a datetime

```
x
```

```
"2013-02-23"
```

```
year(x) <- 2017x
```

```
"2017-02-23"
```

Other useful functions

Function	Extracts
`leap_year()`	In leap year (`TRUE` or `FALSE`)
`am()`	In morning (`TRUE` or `FALSE`)
`pm()`	In a ernoon (`TRUE` or `FALSE`)
`dst()`	During daylight savings (`TRUE` or `FALSE`)
`quarter()`	Quarter of year (1-4)
`semester()`	Half of year (1-2)

Rounding in lubridate

- `round_date()` - round to nearest
- `ceiling_date()` - round up
- `floor_date()` - round to down
- Possible values of `unit` :
  - `"second"` , `"minute"` , `"hour"` , `"day"` , `"week"` , `"month"` , `"bimonth"` , `"quarter"` , `"halfyear"` , or `"year"` .
  - Or multiples, e.g `"2 years"` , `"5 minutes"`

Arithmetic for datetimes

- `datetime_1 - datetime2` : Subtraction for time elapsed
- `datetime_1 + (2 * timespan)` : Addition and multiplication for generating new datetimes in the past or future
- `timespan1 / timespan2` : Division for change of units

difftime()

`units =` `"secs"`, `"mins"`, `"hours"`, `"days"`, or `"weeks"`

```
difftime(Sys.Date(), last_release$date, units = "secs")
```

```
Time difference of 8553600 secs
```

```
difftime(Sys.Date(), last_release$date, units = "weeks")
```

```
Time difference of 14.14286 weeks
```

now() and today()

```
today()
```

```
"2017-10-07"
```

```
str(today())
```

```
 Date[1:1], format: "2017-10-07"
```

```
now()
```

```
"2017-10-07 09:44:52 PDT"
```

```
str(now())
```

```
 POSIXct[1:1], format: "2017-10-07 09:44:59"
```

Time spans in lubridate

period

- Human concept of a time span
- datetime + period of one day = same time on the next date
- variable length

Duration

- Stopwatch concept of a time span
- datetime + duration of one day = datetime + 86400 seconds
- xed number of seconds

Functions to create time spans

Time span	Duration	Period
Seconds	`dseconds()`	`seconds()`
Minutes	`dminutes()`	`minutes()`
Hours	`dhours()`	`hours()`
Days	`ddays()`	`days()`
Weeks	`dweeks()`	`weeks()`
Months	-	`months()`
Years	`dyears()`	`years()`

Creating intervals

`datetime1 %--% datetime2`, or
`interval(datetime1, datetime2)`

```
dmy("5 January 1961") %--% dmy("30 January 1969")
```

```
1961-01-05 UTC--1969-01-30 UTC
```

```
interval(dmy("5 January 1961"), dmy("30 January 1969"))
```

```
1961-01-05 UTC--1969-01-30 UTC
```

Operating on an interval

```
beatles <- dmy("5 January 1961") %--% dmy("30 January 1969")
int_start(beatles)
```

```
"1961-01-05 UTC"
```

```
int_end(beatles)
```

```
"1969-01-30 UTC"
```

Operating on an interval

```
int_length(beatles)
```

```
254620800
```

```
as.period(beatles)
```

```
"8y 0m 25d 0H 0M 0S"
```

```
as.duration(beatles)
```

```
"254620800s (~8.07 years)"
```

Comparing intervals

```
hendrix_at_woodstock <- mdy("August 17 1969")
```

```
hendrix_at_woodstock %within% beatles
```

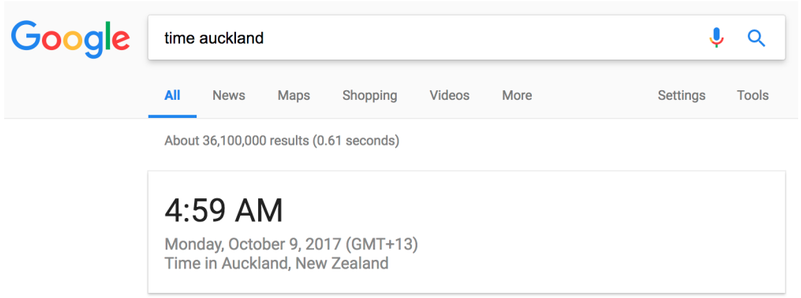
```
FALSE
```

```
hendrix <- dmy("01 October 1966") %--% dmy("16 September 1970")
```

```
int_overlaps(beatles, hendrix)
```

```
TRUE
```

Time zones




```
Sys.timezone()
```

```
"America/Los_Angeles"
```

IANA Timezones

```
OlsonNames()
```

```
"Africa/Abidjan"          "Africa/Accra"
"Africa/Addis_Ababa"      "Africa/Algiers"
"Africa/Asmara"           "Africa/Asmera"
"Africa/Bamako"           "Africa/Bangui"
...
```

```
length(OlsonNames())
```

```
594
```

Setting and extracting

```
mar_11 <- ymd_hms("2017-03-11 12:00:00",
+      tz = "America/Los_Angeles")
mar_11
```

```
"2017-03-11 12:00:00 PST"
```

```
tz(mar_11)
```

```
"America/Los_Angeles"
```

Manipulating timezones

`force_tz()` - change the timezone without changing the clock time

```
mar_11
```

```
"2017-03-11 12:00:00 PST"
```

```
force_tz(mar_11,
    tzone = "America/New_York")
```

```
"2017-03-11 12:00:00 EST"
```

`with_tz()` - view the same instant in a di erent timezone

```
mar_11
```

```
"2017-03-11 12:00:00 PST"
```

```
with_tz(mar_11,
    tzone = "America/New_York")
```

```
"2017-03-11 15:00:00 EST"
```

fast_strptime()

```
x <- "2001-02-27"
parse_date_time(x, order = "ymd")
```

```
"2001-02-27 UTC"
```

```
fast_strptime(x, format = "%Y-%m-%d")
```

```
"2001-02-27 UTC"
```

```
fast_strptime(x, format = "%y-%m-%d")
```

```
NA
```

See Details of `format` in **`strptime()`**

Formatting datetimes

```
my_stamp <- stamp("Tuesday October 10 2017")
```

```
Multiple formats matched: "%A %B %d %y%H"(1), "%A %B %y %d%H"(1),
"%A %B %d %Y"(1), "%A October %m %y%d"(1), "%A October %m %Y"(0),
"%A October %H %M%S"(1), "Tuesday %B %d %y%H"(1), "Tuesday %B %y %d%H"(1),
"Tuesday %B %d %Y"(1), "Tuesday October %m %y%d"(1),
"Tuesday October %m %Y"(1), "Tuesday October %H %M%S"(1)
Using: "%A %B %d %Y"
```

```
my_stamp(ymd("2003-02-27"))
```

```
"Thursday February 27 2003"
```

```
my_stamp
function(x)
format(x, format = "%A %B %d %Y")
```

```
<environment: 0x1086ed780>
```