WORKING WITH DATES AND TIMES IN R

Introduction to dates

Charlotte Wickham
Instructor

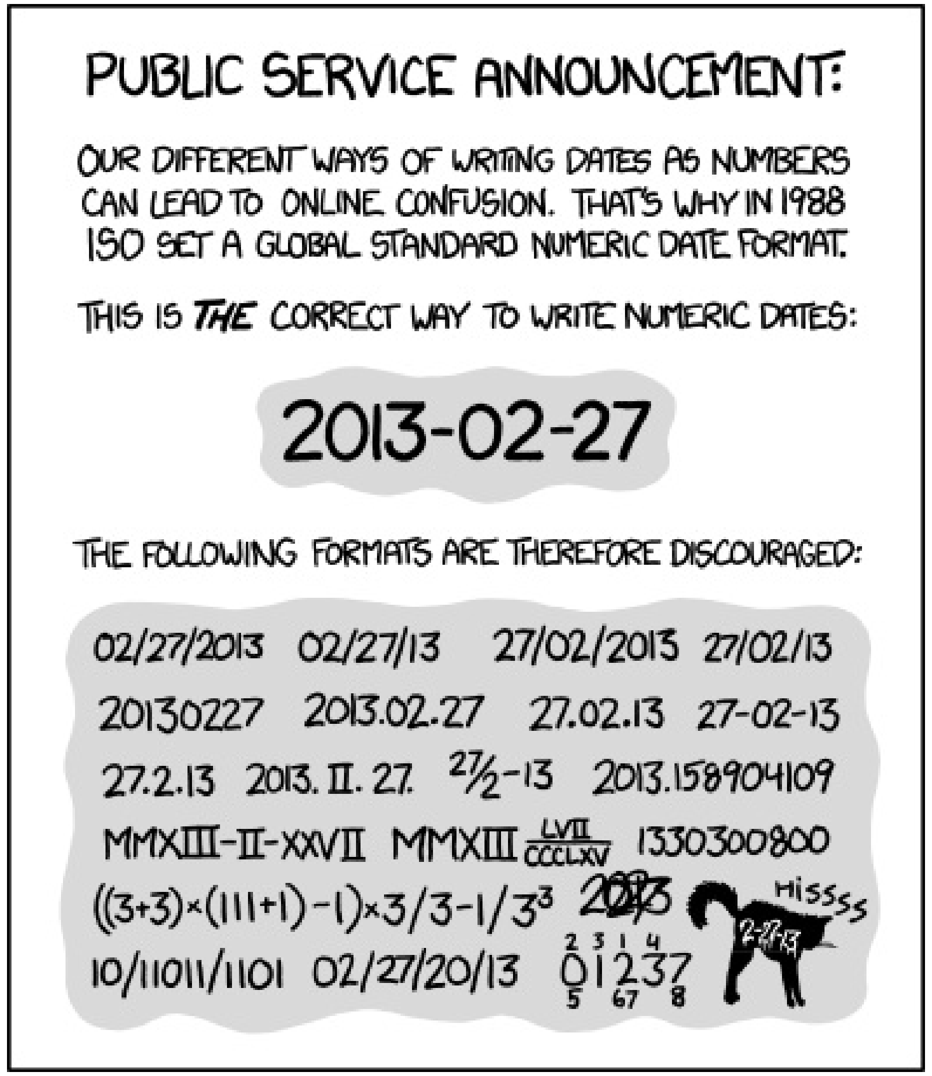
Dates

Different conventions in different places

27th Feb 2013

- NZ: 27/2/2013
- USA: 2/27/2013

The global standard numeric date format



PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS ***THE*** CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27

THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013. II. 27. $^{27}/_{2}$-13 2013.158904109
MMXIII-II-XXVII MMXIII $\frac{LVII}{CCCLXV}$ 1330300800
((3+3)×(111+1)-1)×3/3-1/3^3 2013
10/11011/1101 02/27/20/13 $\overset{2}{\underset{5}{0}}\overset{3}{1}\overset{1}{\underset{67}{2}}\overset{4}{3}\underset{8}{7}$

[1] https://xkcd.com/1179/

ISO 8601 YYYY-MM-DD

- Values ordered from the largest to smallest unit of time
- Each has a fixed number of digits, must be padded with leading zeros
- Either, no separators for computers, or - in dates
- 1st of January 2011 -> 2011-01-01

Dates in R

```
> 2003-02-27
[1] 1974

> "2003-02-27"
[1] "2003-02-27"

> str("2003-02-27")
 chr "2003-02-27"

> as.Date("2003-02-27")
[1] "2003-02-27"

> str(as.Date("2003-02-27"))
 Date[1:1], format: "2003-02-27"
```

- Packages that import dates: readr, anytime

WORKING WITH DATES AND TIMES IN R

Let's practice!

WORKING WITH DATES AND TIMES IN R

Why use dates?

Charlotte Wickham
Instructor

Dates act like numbers

Date objects are stored as days since 1970-01-01

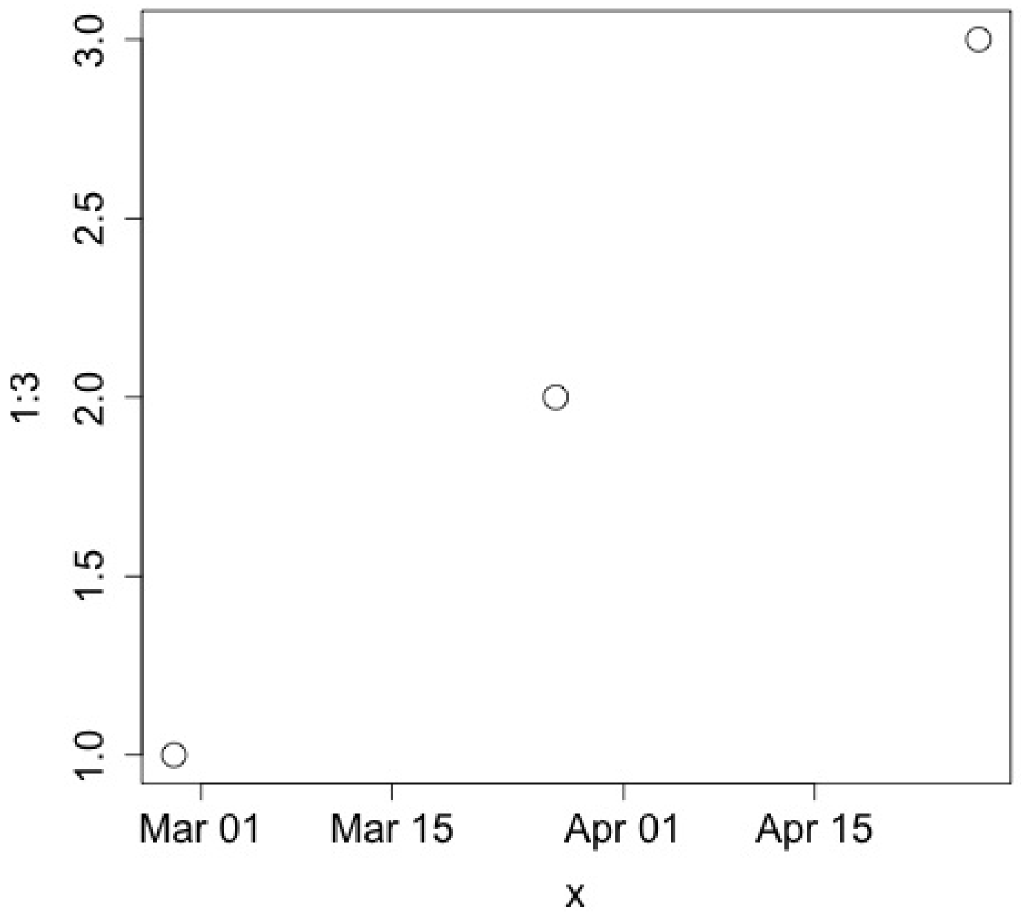
```
> as.Date("2003-02-27") > as.Date("2002-02-27")
[1] TRUE

> as.Date("2003-02-27") + 1
[1] "2003-02-28"

> as.Date("2003-02-27") - as.Date("2002-02-27")
Time difference of 365 days
```

Plotting with dates

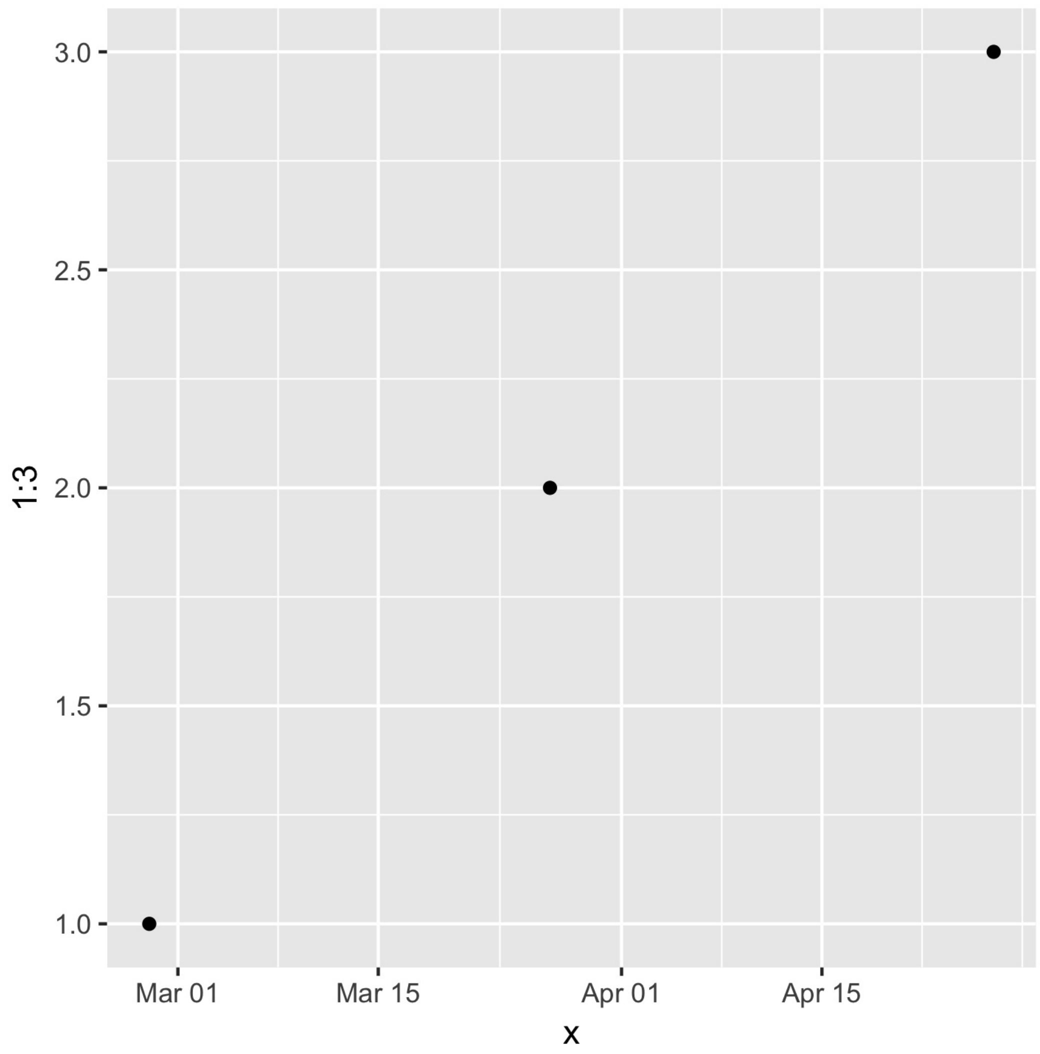
```
x <- c(as.Date("2003-02-27"),
       as.Date("2003-03-27"),
       as.Date("2003-04-27"))
plot(x, 1:3)
```



Plotting with dates

```
x <- c(as.Date("2003-02-27"),
       as.Date("2003-03-27"),
       as.Date("2003-04-27"))

library(ggplot2)
ggplot() +
  geom_point(aes(x = x, y = 1:3))
```



R releases

```
> releases
# A tibble: 105 x 7
   major minor patch       date            datetime     time  type
   <int> <int> <int>     <date>              <dttm>   <time> <chr>
 1     0    60    NA 1997-12-04 1997-12-04 08:47:58 08:47:58 patch
 2     0    61    NA 1997-12-21 1997-12-21 13:09:22 13:09:22 minor
 3     0    61     1 1998-01-10 1998-01-10 00:31:55 00:31:55 patch
 4     0    61     2 1998-03-14 1998-03-14 19:25:55 19:25:55 patch
 5     0    61     3 1998-05-02 1998-05-02 07:58:17 07:58:17 patch
 6     0    62    NA 1998-06-14 1998-06-14 12:56:20 12:56:20 minor
 7     0    62     1 1998-06-14 1998-06-14 22:13:25 22:13:25 patch
 8     0    62     2 1998-07-10 1998-07-10 11:13:45 11:13:45 patch
 9     0    62     3 1998-08-28 1998-08-28 09:02:19 09:02:19 patch
10     0    62     4 1998-10-23 1998-10-23 12:08:41 12:08:41 patch
# ... with 95 more rows
```

WORKING WITH DATES AND TIMES IN R

Let's practice!

WORKING WITH DATES AND TIMES IN R

What about times?

Charlotte Wickham
Instructor

ISO 8601

HH:MM:SS

- Largest unit to smallest
- Fixed digits
  - Hours: 00 -- 24
  - Minutes: 00 -- 59
  - Seconds: 00 -- 60 (60 only for leap seconds)
- No separator or :

Datetimes in R

- Two objects types:
  - POSIXlt - list with named components
  - POSIXct - seconds since 1970-01-01 00:00:00
- POSIXct will go in a data frame

- as.POSIXct() turns a string into a POSIXct object

```
x <- as.POSIXct("1970-01-01 00:01:00")
> str(x)
 POSIXct[1:1], format: "1970-01-01 00:01:00"
```

Timezones

- "2013-02-27T18:00:00" - 6pm local time
- "2013-02-27T18:00:00Z" - 6pm UTC
- "2013-02-27T18:00:00-08:00" - 6pm in Oregon

```
> as.POSIXct("2013-02-27T18:00:00Z")
[1] "2013-02-27 PST"

> as.POSIXct("2013-02-27T18:00:00Z", tz = "UTC")
[1] "2013-02-27 UTC"
```

Datetimes behave nicely too

Once a `POSIXct` object, datetimes can be:

- Compared
- Subtracted
- Plotted

WORKING WITH DATES AND TIMES IN R

Let's practice!

WORKING WITH DATES AND TIMES IN R

Why lubridate?

Charlotte Wickham

Instructor

lubridate

- Make working with dates and times in R easy!
- tidyverse package
  - Plays nicely with builtin datetime objects
  - Designed for humans not computers
- Plays nicely with other tidyverse packages
- Consistent behaviour regardless of underlying object

Parsing a wide range of formats

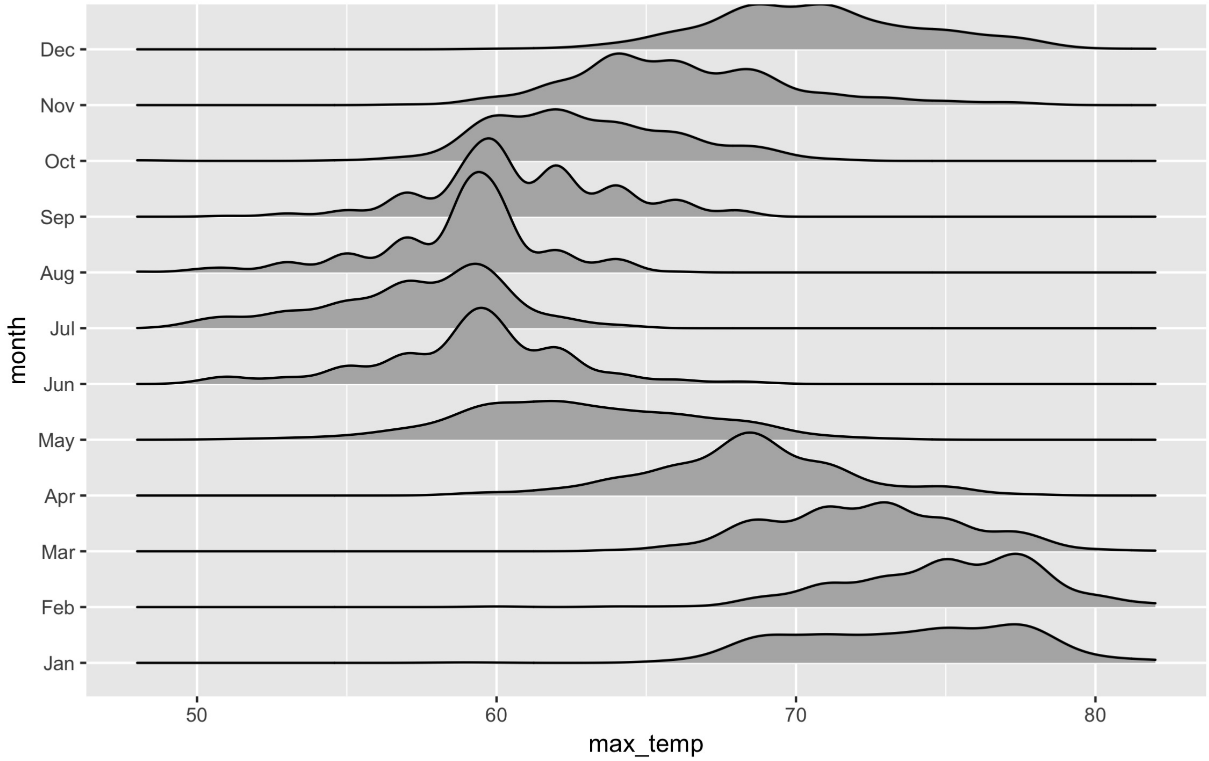
```
> ymd("2013-02-27")
[1] "2013-02-27"

> dmy("27/2/13")
[1] "2013-02-27"

> parse_date_time(c("Feb 27th, 2017", "27th Feb 2017"),
+   order = c("mdy", "dmy"))
[1] "2017-02-27 UTC" "2017-02-27 UTC"
```

Manipulating datetimes

```
# Extract components
akl_daily <- akl_daily %>%
  mutate(
    year = year(date),
    yday = yday(date),
    month = month(date, label = TRUE)
  )
```



Time spans

```
# A tibble: 131 x 3
           name              period
          <chr>        <S4: Period>
1  Elizabeth II 65y 7m 27d 0H 0M 0S
2      Victoria  63y 7m 2d 0H 0M 0S
3      George V 25y 8m 14d 0H 0M 0S
4    George III 19y 0m 28d 0H 0M 0S
5     George VI 15y 1m 26d 0H 0M 0S
```

Other lubridate features

- Handling timezones
- Fast parsing of standard formats
- Outputting datetimes

WORKING WITH DATES AND TIMES IN R

Let's practice!