WORKING WITH DATES AND TIMES IN R

Parsing dates with lubridate

Charlotte Wickham
Instructor

ymd()

- 27th of February 2013
- ymd() - **y**ear, then **m**onth, then **d**ay

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

> ymd("2013.02.27")
[1] "2013-02-27"

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

Friends of ymd()

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

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

> mdy("02-27-2013")
[1] "2013-02-27"

> dmy_hm("27-02-2013 12:12pm")
[1] "2013-02-27 12:12:00 UTC"
```

parse_date_time(x = ___, order = ___)

```
> parse_date_time("27-02-2013", order = "dmy")
[1] "2013-02-27 UTC"

> parse_date_time(c("27-02-2013", "2013 Feb 27th"),
+     order = c("dmy", "ymd"))
[1] "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, offset in hours and minutes from UTC

WORKING WITH DATES AND TIMES IN R

Let's practice!

WORKING WITH DATES AND TIMES IN R

Weather in Auckland

Charlotte Wickham
Instructor

akl_weather_daily.csv

```
date,max_temp,min_temp,mean_temp,mean_rh,events,cloud_cover
2007-9-1,60,51,56,75,NA,4
2007-9-2,60,53,56,82,Rain,4
2007-9-3,57,51,54,78,NA,6
2007-9-4,64,50,57,80,Rain,6
2007-9-5,53,48,50,90,Rain,7
```

akl_weather_hourly_2016.csv

```
year,month,mday,time,temperature,weather,conditions,events,humidity,date_utc
2016,1,1,00:00:00,68,Clear,Clear,NA,68,2015-12-31T11:00:00Z
2016,1,1,00:30:00,68,Clear,Clear,NA,68,2015-12-31T11:30:00Z
2016,1,1,01:00:00,68,Clear,Clear,NA,73,2015-12-31T12:00:00Z
2016,1,1,01:30:00,68,Clear,Clear,NA,68,2015-12-31T12:30:00Z
2016,1,1,02:00:00,68,Clear,Clear,NA,68,2015-12-31T13:00:00Z
```

make_date(year, month, day)

```
> make_date(year = 2013, month = 2, day = 27)
[1] "2013-02-27"
```

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

dplyr Review

- mutate() - add new columns (or overwrite old ones)
- filter() - subset rows
- select() - subset columns
- arrange() - order rows
- summarise() - summarise rows
- group_by() - useful in conjuction with summarise()

Pipe %>%

```
# Without the pipe: nested functions
summarise(group_by(filter(releases, major == 3), minor), n =  n())

# With pipe: more linear
releases %>%
  filter(major == 3) %>%
  group_by(minor) %>%
  summarise(n = n())
```

WORKING WITH DATES AND TIMES IN R

Let's practice!

WORKING WITH DATES AND TIMES IN R

Extracting parts of a datetime

Charlotte Wickham
Instructor

Extracting parts of a datetime

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

> year(x)
[1] 2013

> month(x)
[1] 2

> day(x)
[1] 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
[1] "2013-02-23"

> year(x) <- 2017
> x
[1] "2017-02-23"
```

Other useful functions

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

WORKING WITH DATES AND TIMES IN R

Let's practice!

WORKING WITH DATES AND TIMES IN R

Rounding datetimes

Charlotte Wickham
Instructor

Rounding versus extracting

```
> release_time <- releases$datetime
> head(release_time)
[1] "1997-12-04 08:47:58 UTC" "1997-12-21 13:09:22 UTC"
[3] "1998-01-10 00:31:55 UTC" "1998-03-14 19:25:55 UTC"
[5] "1998-05-02 07:58:17 UTC" "1998-06-14 12:56:20 UTC"

> head(release_time) %>% hour()
[1]  8 13  0 19  7 12

> head(release_time) %>% floor_date(unit = "hour")
[1] "1997-12-04 08:00:00 UTC" "1997-12-21 13:00:00 UTC"
[3] "1998-01-10 00:00:00 UTC" "1998-03-14 19:00:00 UTC"
[5] "1998-05-02 07:00:00 UTC" "1998-06-14 12:00:00 UTC"
```

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"

WORKING WITH DATES AND TIMES IN R

Let's practice!