MATH 118: Notes P

Code

Lubridate

Hide

```
#LOAD PACKAGES
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.2      ✔ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Hide

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Hide

```
#LOAD DATA
friendsvisits <- read.csv("data/friendsvisits.csv")
# In short, this made-up dataset contains information about the times that our host(s) had friends over and the
 times that they left.
```

Date Formats

Think of how many different formats you know of to list today's date:

Nov 15, 2022 November 15, 2022 11-15-2022 2022-11-15 11/15/2022

Yikes!

Date, Time, and Datetime

Date/time data are data that conveys information about, you guessed it, date and/or time! There are three relevant data types when we talk about date/time data:

1. Date - only has the date (e.g. 2020-05-15)
2. Time - only has the time (e.g. 20:45:00)
3. Datetime - has both the date and time (e.g. 2020-05-15 20:45:00)

Standard Date Format

The `ymd()` function transforms data in all kinds of different formats into a standardized date format displaying year, then month, then day.

Hide

```
ymd("06 02 04")
```

```
## [1] "2006-02-04"
```

Hide

```
ymd("06/02/04")
```

```
## [1] "2006-02-04"
```

Hide

```
ymd("20060204")  # works as well
```

```
## [1] "2006-02-04"
```

Hide

```
ymd("2006 2 4")
```

```
## [1] "2006-02-04"
```

Hide

```
ymd(060204)  # works with numbers
```

```
## [1] "2006-02-04"
```

`mdy()` (month day year) and `dmy()` (day month year) formats also exist.

Hide

```
ymd_hms("2020-04-01 10:30:13")
```

```
## [1] "2020-04-01 10:30:13 UTC"
```

Hide

```
ymd_hm("2020/04/01 10.30")
```

```
## [1] "2020-04-01 10:30:00 UTC"
```

Making Dates From Values

Hide

```
make_date(year = 2020, month = 7, day = 13)
```

```
## [1] "2020-07-13"
```

Hide

```
make_datetime(year = 2020, month = 7, day = 13, hour = 8, min=30)
```

```
## [1] "2020-07-13 08:30:00 UTC"
```

Unfortunately, because the information about datetime is divided up into different columns, R does not recognize it as date/time data. What we need to do is combine and convert all of these columns into datetime. To do this, we can use the function `make_datetime()`

Hide

```
friendsvisits <- friendsvisits %>%
    mutate(Visit_timedate = make_datetime(Year, Month, Day, Hour, Minute))
```

How can we turn values like 1951 into 19:51:00? Once again, `make_datetime()` is the answer!

Hide

```
#%% indicates x mod y and %/% indicates integer division.
# %% indicates 830 mod 100 = 8
# %/% indicates the remainder of 830 mod 100 = 30

friendsvisits <- friendsvisits %>%
    mutate(Left_timedate = make_datetime(Year, Month, ifelse(Left_time %/%100 > Hour,Day,Day+1), Left_time %/% 1
00, Left_time %% 100))
```

Working with Time Spans

Hide

```
friendsvisits <- friendsvisits %>%
  mutate(visit_length = difftime(Left_timedate, Visit_timedate))
```

Rounding Times

Hide

```
t <- ymd_hms("2022-11-15 00:08:30")
```

Hide

```
#Round
round_date(t, unit = "second") #no change
```

```
## [1] "2022-11-15 00:08:30 UTC"
```

Hide

```
round_date(t, unit = "minute")
```

```
## [1] "2022-11-15 00:09:00 UTC"
```

Hide

```
round_date(t, unit = "hour")
```

```
## [1] "2022-11-15 UTC"
```

Hide

```
round_date(t, unit = "day")
```

```
## [1] "2022-11-15 UTC"
```

Hide

```
#Force Round UP
ceiling_date(t, unit = "minute")
```

```
## [1] "2022-11-15 00:09:00 UTC"
```

Hide

```
ceiling_date(t, unit = "hour")
```

```
## [1] "2022-11-15 01:00:00 UTC"
```

Hide

```
#Force Round DOWN
floor_date(t, unit = "minute")
```

```
## [1] "2022-11-15 00:08:00 UTC"
```

Hide

```
floor_date(t, unit = "hour")
```

```
## [1] "2022-11-15 UTC"
```