Exploring data #1

Dates in R

`Date` class

A common task when changing or adding columns is to change the class of some of the columns. This is especially common for dates, which will often be read in as a character vector when reading data into R.

`lubridate` package

The `lubridate` package is helpful for working with vectors of dates or date-times.

You will see dates represented in many different ways. For example, October might be included in data as "October", "Oct", or "10". Further, the way the elements are separated can vary.

Computers are very literal, so this ambiguity can be confusing for them.

`lubridate` package

The `lubridate` package has a number of functions for converting character strings into dates (or date-times). To decide which one to use, you just need to know the order of the elements of the date in the character string.

For example, here are some commonly-used `lubridate` functions:

`lubridate` function	Order of date elements
`ymd`	year-month-day
`dmy`	day-month-year
`mdy_hm`	month-day-year-hour-minute
`ymd_hms`	year-month-day-hour-minute-second

(Remember, you can use `vignette("lubridate")` and `?lubridate` to get help with the `lubridate` package.)

`lubridate` package

In many cases you can use functions from the `lubridate` package to parse dates pretty easily.

For example, if you have a character string with the date in the order of *year-month-day*, you can use the `ymd` function from `lubridate` to convert the character string to the `Date` class. For example:

```
library("lubridate")
my_date <- ymd("2008-10-13")
class(my_date)
```

```
## [1] "Date"
```

`lubridate` package

The functions in `lubridate` are pretty good at working with different ways of expressing date and time elements intelligently:

```
mdy("10-31-2017")
```

```
## [1] "2017-10-31"
```

```
dmy("31 October 2017")
```

```
## [1] "2017-10-31"
```

lubridate package

There are lubridate functions that can parse date-times, too:

```
ymd_hms("2017/10/31--17:33:10")
```

```
## [1] "2017-10-31 17:33:10 UTC"
```

```
mdy_hm("Oct. 31, 2017 5:33PM", tz = "MST")
```

```
## [1] "2017-10-31 17:33:00 MST"
```

Converting to `Date` class

We can use the `mdy_hms` function from `lubridate` to convert the `sample_time` column in the `beijing_pm` dataset to a date-time class ("POSIXct"):

```
beijing_pm <- beijing_pm %>%
  mutate(sample_time = mdy_hm(sample_time))

head(beijing_pm, 3)
```

```
## # A tibble: 3 x 4
##   sample_time         value qc    aqi
##   <dttm>              <dbl> <chr> <fct>
## 1 2017-01-01 00:00:00   505 Valid Beyond Index
## 2 2017-01-01 01:00:00   485 Valid Hazardous
## 3 2017-01-01 02:00:00   466 Valid Hazardous
```

Converting to `Date` class

Once you have an object in a date or date-time class, you can do things like plot by date, calculate the range of dates, and calculate the total number of days the dataset covers:

```
range(beijing_pm$sample_time)
```

```
## [1] "2017-01-01 00:00:00 UTC"
## [2] "2017-06-30 23:00:00 UTC"
```

```
diff(range(beijing_pm$sample_time))
```

```
## Time difference of 180.9583 days
```

lubridate package

The lubridate package also includes functions to pull out certain elements of a date. For example, we could use wday to create a new column with the weekday of each show:

```
beijing_pm <- mutate(beijing_pm,
                     sample_weekday = wday(sample_time,
                                           label = TRUE))
```

lubridate package

```
beijing_pm %>%
  select(sample_time, sample_weekday) %>%
  sample_n(size = 3)
```

```
## # A tibble: 3 x 2
##   sample_time         sample_weekday
##   <dttm>              <ord>
## 1 2017-06-21 15:00:00 Wed
## 2 2017-03-29 19:00:00 Wed
## 3 2017-03-31 19:00:00 Fri
```

`lubridate` package

The `wday` function created an **ordered factor** ("ord" below the column name in the tibble print-out). You can use this like other factors.

lubridate package

```
beijing_pm %>%
  group_by(sample_weekday) %>%
  summarize(mean_pm = mean(value))
```

```
## # A tibble: 7 x 2
##   sample_weekday mean_pm
##   <ord>             <dbl>
## 1 Sun                67.6
## 2 Mon                52.2
## 3 Tue                64.3
## 4 Wed                76.1
## 5 Thu                75.4
## 6 Fri                61.0
## 7 Sat                45.1
```

`lubridate` package

Other functions in `lubridate` for pulling elements from a date include:

- `mday`: Day of the month
- `yday`: Day of the year
- `month`: Month
- `quarter`: Fiscal quarter
- `year`: Year