Stat Computing - Exercises 01 - Dates and Date-times

Include your answers as code chunks within this document, and render (knit) it as a pdf.

To help the graders differentiate between the question statements and your own comments, please change the formatting of the question statements to italics.

Due to the many formatting conventions for dates and times, and due to time zone considerations, dates can be tricky to work with. These exercises will give you some practice working with dates and times in R.

1. Read the documentation for `as.Date` and answer the following questions:
    a. Why does `as.Date("2024-01-01")` work without supplying a format?
    b. Why doesn't `as.Date(365)` work and how can you make it work to produce the date 2024-01-01?
2. Using a combination of `seq` and `as.Date`, display the dates of our Tuesday lectures throughout the semester (you can ignore the fact that we don't meet during spring break).
3. Read the documentation for `strptime` and use `format` convert 2024-01-23 to these formats.
    a. 01/23/2024
    b. 23/01/2024
    c. 01/23/24
    d. January 23, 2024
    e. Jan 23, 2024
4. Use `format` to figure out what day of the week you were born on.
5. How many days are in the years 2000, 2022, 2024, and 2100? Why? Hint: R gives the right answer for all of these.
6. In a few sentences, describe the main differences between POSIXct and POSIXlt. Hint: `?POSIXlt`
7. Convert 2024-01-01 and 2024-07-01 to POSIXct format, and print them out. What do the three-letter abbreviations stand for?
8. Use `as.POSIXct` to create two date time objects, one referring to 2024-01-26 at noon in Ithaca's time zone and one referring 2024-01-26 at noon in UTC time. Calculate the difference between these two objects.
9. Use the `attr` function to set the `"tzone"` attribute for the UTC object to Ithaca time, and confirm that the two printed times differ by the amount in the previous question.
10. Create a vector that has the datetimes 2024-01-26 12:00:00 and 2024-07-26 12:00:00 in the America/New_York time zone. Then successively change the time zone to the below time zones, and print the result. Do you notice anything unexpected?

    America/Chicago, America/Denver, America/Phoenix, America/Los_Angeles
11. Share something interesting that you learned from reading the documentation for these functions, or found online.