Data science with R: tidyverse

IV Data Wrangle: dates / times (lubridate & hms)

Assignment

Create *R* script called *assignment_4.R*. From course sources, download file called **energy_consumption.zip**, extract its content in your **data** folder inside your *R*'s project folder.

Exercise 1

In the first exercise we will test our date / date time parsing skills using **lubridate** package helper functions. For each string below use an adequate function and parse string to date or date time object:

- "2021-01-15 23:05:30"
- "2030-01-01 05"
- "2000-28-02 10:15"
- "1990-15-03 04"
- "05/30/1995 9:15:45"
- "1 Nov 2040 01/02:00"
- "30 Jun 2035 20:45:00"
- "20000101"
- "January 1st 2029"
- "October 2nd 2028"
- "July 15th 2027"
- "30th March 25"
- "2015: Q2"

Exercise 2

In the given exercise, we would like to check which are the leap years between year 1 and year 3000 (AD / "Anno Domini"   after year 0). You won't have to write a procedure for leap year testing from scratch, we will use **lubridate**. Do the following:

- first create sequence of first days for each year (**HINT: seq.Date()**)
- convert sequence to **tibble**
- add column **year**
- check if difference between 2 rows in your **tibble** is 1 year!
- to **tibble** add flag **leap year** - use **lubridate**
- How many leap years are all together?
- Which are the leap years?
- count leap years per century
- Do all centuries have the same number of leap years?

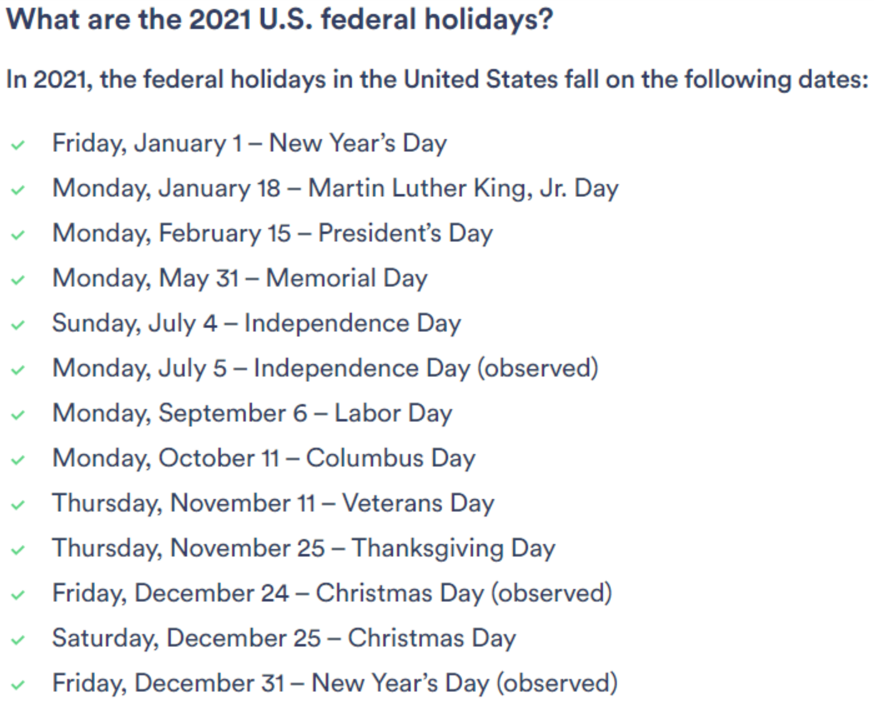
Exercise 3

In the third exercise we will inspect holidays in the USA. The figure 1 shows a list of federal holidays in USA (source:
`https://www.zenefits.com/workest/list-of-2021-federal-holidays-for-small-business-owners/`)

Use the data from the figure 1 / url, and do the following:

- first store all holidays in a **tibble**
- create two columns: **holiday** and **date**
- calculate durations: how many days / weeks / hours / seconds is between two successive holidays
- **HINT:** use **dplyr**'s **lag** or **lead** function
- **HINT:** date difference of two holidays concert to **period**
- **HINT:** divide with specific duration constructor function
- try answering the following questions:
- Is today a holiday?
- Which holiday was the last one?
- Which holiday will be the next one?

Figure 1: US holidays



What are the 2021 U.S. federal holidays?

In 2021, the federal holidays in the United States fall on the following dates:

- ✓ Friday, January 1 – New Year's Day
- ✓ Monday, January 18 – Martin Luther King, Jr. Day
- ✓ Monday, February 15 – President's Day
- ✓ Monday, May 31 – Memorial Day
- ✓ Sunday, July 4 – Independence Day
- ✓ Monday, July 5 – Independence Day (observed)
- ✓ Monday, September 6 – Labor Day
- ✓ Monday, October 11 – Columbus Day
- ✓ Thursday, November 11 – Veterans Day
- ✓ Thursday, November 25 – Thanksgiving Day
- ✓ Friday, December 24 – Christmas Day (observed)
- ✓ Saturday, December 25 – Christmas Day
- ✓ Friday, December 31 – New Year's Day (observed)

Exercise 4

In this exercise we will use dataset found on **kaggle** website. Data is related to hourly energy consumption in the USA - provided by the organization called **PJM Interconnection LLC (PJM)**. Data source provides hourly data about energy consumption in megawatts (MW) for given US regions.

Source of the data comes from:
`https://www.kaggle.com/robikscube/hourly-energy-consumption?select=PJME_hourly.csv`

First, from the course sources download the file called **energy_consumption.zip**, unzip the file into the folder **data** inside your project folder. Now do the following:

- import the **.csv** file **pjm_hourly_est.csv**
- keep only columns **Datetime** and **PJME**
- do not forget column parsing!
- remove rows where **PJME** data is missing!
- sort rows based on date time column
- check if data is for every hour in given time span?
- now add columns: **date**, **month**, **year** (**lubridate**)
- calculate time intervals: **year intervals**, **month intervals**, **day intervals**
- **HINT:** per **year** / **month** / **day** calculate **minimum** and **maximum** time stamp
- **HINT:** calculate intervals using **lubridate**
- now use your intervals and calculate **total** and **mean hourly energy consumption** per each **year** / each **month** / each **day**