# Intro to working with lubridate package

This project will introduce the basics around how R and lubridate handle dates. While this package is part of the tidyverse, it is not automatically loaded when we use the library(tidyverse) command. Thus, you will need to use the library(lubridate) command to use this package.

## Loading libraries needed for practice

library("tidyverse")  
library(lubridate)

## Basics around lubridate

### Working with Dates

this\_day <- today()  
this\_day

## [1] "2019-11-27"

date\_ex <- "2019-05-12"  
  
y <- year(date\_ex)  
m <- month(date\_ex)  
d <- day(date\_ex)  
  
y

## [1] 2019

m

## [1] 5

d

## [1] 12

rbind(y,m,d)

## [,1]  
## y 2019  
## m 5  
## d 12

#You can also get the exact day of the week (represented as a number where 1 - Sunday, 2 - Monday, 3 - Tuesday, 4 - Wednesday, 5 - Thursday, 6 - Friday, 7 - Saturday )  
w <- wday(date\_ex)  
w

## [1] 1

#You can also see what day of the year it is. The example below shows that May 12, 2019 is the 132nd day of the year  
dayofyear <- yday(date\_ex)  
dayofyear

## [1] 132

### Working with Times

this\_moment <- now()  
this\_moment

## [1] "2019-11-27 13:17:41 EST"

hr <- hour(this\_moment)  
minu <- minute(this\_moment)  
sec <- second(this\_moment)  
  
rbind(hr,minu,sec)

## [,1]  
## hr 13.00000  
## minu 17.00000  
## sec 41.70681

### Parsing Dates

#Lubridate recognizes the format of various dates entered  
my\_date <- ymd("1989-05-17")  
my\_date

## [1] "1989-05-17"

my\_date2 <- ymd("1989 May 17")  
my\_date3 <- mdy("March 12, 1975")  
my\_date4 <- dmy(25081985)

###Time Zones To find the current date in New York, we’’ll use the now() function again. This time, however, we’ll specify the time zone that we want: “America/New\_York”. “For a complete list of valid time zones for use with lubridate, check out the following Wikipedia <http://en.wikipedia.org/wiki/List_of_tz_database_time_zones>.

###

### Arithmetic

graduation\_date <- mdy(08152020)  
graduation\_date

## [1] "2020-08-15"

start\_date <- mdy(09012019)  
start\_date

## [1] "2019-09-01"

#We can combine these to create an interval of time from start to end  
how\_long <- interval(start\_date,graduation\_date)  
how\_long

## [1] 2019-09-01 UTC--2020-08-15 UTC

as.period(how\_long)

## [1] "11m 14d 0H 0M 0S"

### Changing existing data

install.packages("dslabs")  
install.packages("nycflights13")

library(nycflights13)  
flights

## # A tibble: 336,776 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 1 1 517 515 2 830  
## 2 2013 1 1 533 529 4 850  
## 3 2013 1 1 542 540 2 923  
## 4 2013 1 1 544 545 -1 1004  
## 5 2013 1 1 554 600 -6 812  
## 6 2013 1 1 554 558 -4 740  
## 7 2013 1 1 555 600 -5 913  
## 8 2013 1 1 557 600 -3 709  
## 9 2013 1 1 557 600 -3 838  
## 10 2013 1 1 558 600 -2 753  
## # ... with 336,766 more rows, and 12 more variables: sched\_arr\_time <int>,  
## # arr\_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,  
## # origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
## # minute <dbl>, time\_hour <dttm>

make\_datetime\_100 <- function(year, month, day, time) {  
 make\_datetime(year, month, day, time %/% 100, time %% 100)  
}  
  
flights\_dt <- flights %>%   
 filter(!is.na(dep\_time)) %>%   
 mutate(  
 dep\_time = make\_datetime\_100(year, month, day, dep\_time),  
 arr\_time = make\_datetime\_100(year, month, day, arr\_time),  
 sched\_dep\_time = make\_datetime\_100(year, month, day, sched\_dep\_time),  
 sched\_arr\_time = make\_datetime\_100(year, month, day, sched\_arr\_time)  
 ) %>%   
 select(origin, dest, ends\_with("delay"), ends\_with("time"))  
  
flights\_dt <-flights\_dt %>%  
 mutate (  
 actual\_air\_time = interval(dep\_time,arr\_time),  
 actual\_air\_time = as.period(actual\_air\_time),  
 actual\_air\_time\_minutes = arr\_time - dep\_time,  
 scheduled\_air\_time = interval(sched\_dep\_time,sched\_arr\_time),  
 scheduled\_air\_time = as.period(scheduled\_air\_time),  
 scheduled\_air\_time\_minutes = sched\_arr\_time - sched\_dep\_time  
 )