

# Assignment 7, Part I

Ellen Bledsoe

2024-02-29

## Assignment Details

### Purpose

The goal of this assignment is to work with dates and times using the `lubridate` package.

### Task

Write R code to successfully answer each question below.

### Criteria for Success

- Code is within the provided code chunks or new code chunks are created where necessary
- Code chunks run without errors
- Code chunks have brief comments indicating which code is answering which part of the question
- Code will be assessed as follows:
  - Produces the correct answer using the requested approach: 100%
  - Generally uses the right approach, but a minor mistake results in an incorrect answer: 90%
  - Attempts to solve the problem and makes some progress using the core concept, but returns the wrong answer and does not demonstrate comfort with the core concept: 50%
  - Answer demonstrates a lack of understanding of the core concept: 0%
- Any questions requiring written answers are answered with sufficient detail

### Due Date

March 11 at midnight MST

## Assignment Exercises

The assignment for week 7 is divided into 2 parts:

- Part 1: `lubridate`
- Part 2: `stringr`

## 1. Set-Up (5 pts)

Load in the tidyverse.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.4.4      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

## 2. When Did You Knit This Document? (5 pts)

Write code to do the following:

- The first line should print only the date that you knit this document.
- The second line should print the date and time that you knit this document.

```
#a
today()
```

```
## [1] "2024-02-29"
```

```
#b
now()
```

```
## [1] "2024-02-29 09:02:28 MST"
```

## 3. Plant Vouchers (20 pts)

During my PhD, I collected plant vouchers from the Portal Project that I eventually submitted to the UA Herbarium for identification. I also got DNA sequences for most of them (but not all). Read in that dataset by running the code chunk below.

```
vouchers <- read_csv("https://raw.githubusercontent.com/weecology/DNA_metabarcoding/master/data/collect
```

```
## Rows: 165 Columns: 17
## -- Column specification -----
## Delimiter: ","
## chr (11): season, sp_code, sci_name_fieldID, sci_name_profID, voucher, DNA, ...
## dbl (6): year, month, day, easting, northing, elevation (m)
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

vouchers

```
## # A tibble: 165 x 17
##   year month   day season sp_code  sci_name_fieldID  sci_name_profID voucher
##   <dbl> <dbl> <dbl> <chr>  <chr>    <chr>            <chr>          <chr>
## 1  2017     1    26 <NA>  <NA>    Panicum miliaceum  Panicum miliac~ <NA>
## 2  2016     3    20 winter cass bauh  Cassia bauhinoides  Senna bauhinoi~ Y
## 3  2016     3    20 winter spha hast  Sphaeralcea cocci~ Sphaeralcea ha~ Y
## 4  2016     3    20 winter amsi tess  Amsinckia tessell~ Amsinckia tess~ Y
## 5  2016     3    20 winter micr lene  Uropappus lindleyi  Uropappus lind~ Y
## 6  2016     3    20 winter erig conc  Erigeron concinnus  Erigeron conc~ Y
## 7  2016     3    20 winter atri cane  Atriplex canescens  Atriplex canes~ Y
## 8  2016     3    20 winter euro lana  Eurotia lanata      Krascheninniko~ Y
## 9  2016     3    20 winter pros glan  Prosopis glandulo~ Prosopis gland~ Y
## 10 2016     3    20 winter phac ariz  Phacelia arizonica  Phacelia arizo~ Y
## # i 155 more rows
## # i 9 more variables: DNA <chr>, label_number <chr>, collector <chr>,
## #   location <chr>, easting <dbl>, northing <dbl>, 'elevation (m)' <dbl>,
## #   vial_barcode <chr>, notes <chr>
```

- Using the `make_date` function and the `mutate()` function, create a new column called “collection\_date” that has the year, month, and day that I collected the voucher specimen for each plant. Save this new column in the `vouchers` dataframe.
- Using the `min()` function, find the earliest date that I collected a voucher specimen. Using the `summarize()` function is optional.
- Use code to find the last date that I collected a voucher specimen.
- Find the span of time (duration) that I was collecting voucher specimens.
- Create a column for the day of year that each specimen was collected.

```
#a
vouchers <- vouchers %>%
  mutate(collection_date = make_date(year = year,
                                     month = month,
                                     day = day), .after = day)
vouchers
```

```
## # A tibble: 165 x 18
##   year month   day collection_date season sp_code  sci_name_fieldID
##   <dbl> <dbl> <dbl> <date>    <chr>  <chr>    <chr>
## 1  2017     1    26 2017-01-26 <NA>  <NA>    Panicum miliaceum
## 2  2016     3    20 2016-03-20 winter cass bauh  Cassia bauhinoides
## 3  2016     3    20 2016-03-20 winter spha hast  Sphaeralcea coccinea
## 4  2016     3    20 2016-03-20 winter amsi tess  Amsinckia tessellata
## 5  2016     3    20 2016-03-20 winter micr lene  Uropappus lindleyi
## 6  2016     3    20 2016-03-20 winter erig conc  Erigeron concinnus
## 7  2016     3    20 2016-03-20 winter atri cane  Atriplex canescens
## 8  2016     3    20 2016-03-20 winter euro lana  Eurotia lanata
## 9  2016     3    20 2016-03-20 winter pros glan  Prosopis glandulosa
## 10 2016     3    20 2016-03-20 winter phac ariz  Phacelia arizonica
## # i 155 more rows
## # i 11 more variables: sci_name_profID <chr>, voucher <chr>, DNA <chr>,
## #   label_number <chr>, collector <chr>, location <chr>, easting <dbl>,
## #   northing <dbl>, 'elevation (m)' <dbl>, vial_barcode <chr>, notes <chr>
```

```

#b
min(vouchers$collection_date)

## [1] "2016-03-20"

#c
max(vouchers$collection_date)

## [1] "2019-04-01"

#d
as.duration(interval(min(vouchers$collection_date), max(vouchers$collection_date)))

## [1] "95644800s (~3.03 years)"

#e
vouchers %>%
  mutate(DOY = yday(collection_date), .after = collection_date)

## # A tibble: 165 x 19
##   year month   day collection_date DOY season sp_code sci_name_fieldID
##   <dbl> <dbl> <dbl> <date>         <dbl> <chr> <chr>    <chr>
## 1 2017     1    26 2017-01-26         26 <NA> <NA>    Panicum miliaceum
## 2 2016     3    20 2016-03-20         80 winter cass  bauh  Cassia bauhinoides
## 3 2016     3    20 2016-03-20         80 winter spha  hast  Sphaeralcea coccinea
## 4 2016     3    20 2016-03-20         80 winter amsi  tess  Amsinckia tessellata
## 5 2016     3    20 2016-03-20         80 winter micr  lene  Uropappus lindleyi
## 6 2016     3    20 2016-03-20         80 winter erig  conc  Erigeron concinnus
## 7 2016     3    20 2016-03-20         80 winter atri  cane  Atriplex canescens
## 8 2016     3    20 2016-03-20         80 winter euro  lana  Eurotia lanata
## 9 2016     3    20 2016-03-20         80 winter pros  glan  Prosopis glandulosa
## 10 2016     3    20 2016-03-20         80 winter phac  ariz  Phacelia arizonica
## # i 155 more rows
## # i 11 more variables: sci_name_profID <chr>, voucher <chr>, DNA <chr>,
## #   label_number <chr>, collector <chr>, location <chr>, easting <dbl>,
## #   northing <dbl>, 'elevation (m)' <dbl>, vial_barcode <chr>, notes <chr>

```

#### 4. NDVI from the Santa Rita Experimental Range (20 pts)

There is a large network of phenocams (cameras set up to take daily images of a landscape to monitor plant phenology) across the US. The Santa Rita Experimental Range (SRER) has one such camera.

One value that we can calculate from these phenocam images is the NDVI of the landscape in the picture, which gives us a measurement of vegetation “greenness.” This value is likely to change through time due to seasonal changes in temperature, precipitation, etc.

Run the following code chunk to bring in a subset of the SRER phenocam data.

```

phenocam <- read_csv("neon_srер_ndvi_phencom.csv", skip = 17, col_names = TRUE,
  col_types = "ccncc") %>%
  select(date, local_std_time, contains("mean"), NDVI_c)

```

- Create a new column in `phenocam` that uses the `unite()` function to join the date and time columns together into one column. Separate the date and time with a space.
- Convert the datetime column you created in (a) to a POSIXct format. Save this new column to `phenocam`. You can use whichever function you would like.
- Calculate the duration of the phenocam dataset provided here.
- Create new columns in `phenocam` for the year, the month, and the day of year.
- Using the year and month columns you created in (d), calculate the average NDVI value for each month in the dataset (each month in each year).

```
#a create datetime column using unite
phenocam <- phenocam %>%
  unite("datetime", date, local_std_time, sep = " ")
phenocam
```

```
## # A tibble: 110,270 x 6
##   datetime      r_mean g_mean b_mean ir_mean NDVI_c
##   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2017-02-24 17:15:05    66    57    33    93 -0.0136
## 2 2017-02-24 17:30:05    67    56    31    92  0.0244
## 3 2017-02-24 17:45:06    72    58    30    91  0.0893
## 4 2017-02-24 18:00:05    77    58    28    87  0.179
## 5 2017-02-24 18:15:06    66    67    42    85  0.0668
## 6 2017-02-24 18:30:06    70    63    44    73 -0.792
## 7 2017-02-24 18:45:06    38    24    30    66 -0.0219
## 8 2017-02-24 19:00:06    21    12    15    19 -0.694
## 9 2017-02-25 06:15:05    24    14    19    19 -0.909
## 10 2017-02-25 06:30:05    48    37    43   104  0.127
## # i 110,260 more rows
```

```
#b convert to POSIXct using lubridate
phenocam <- phenocam %>%
  mutate(datetime = ymd_hms(datetime))
phenocam
```

```
## # A tibble: 110,270 x 6
##   datetime      r_mean g_mean b_mean ir_mean NDVI_c
##   <dtm>      <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2017-02-24 17:15:05    66    57    33    93 -0.0136
## 2 2017-02-24 17:30:05    67    56    31    92  0.0244
## 3 2017-02-24 17:45:06    72    58    30    91  0.0893
## 4 2017-02-24 18:00:05    77    58    28    87  0.179
## 5 2017-02-24 18:15:06    66    67    42    85  0.0668
## 6 2017-02-24 18:30:06    70    63    44    73 -0.792
## 7 2017-02-24 18:45:06    38    24    30    66 -0.0219
## 8 2017-02-24 19:00:06    21    12    15    19 -0.694
## 9 2017-02-25 06:15:05    24    14    19    19 -0.909
## 10 2017-02-25 06:30:05    48    37    43   104  0.127
## # i 110,260 more rows
```

```
#c duration of dataset
as.duration(interval(min(phenocam$datetime), max(phenocam$datetime)))
```

```
## [1] "220585501s (~6.99 years)"
```

```
#d make separte columns for year, month, and DOY
```

```
phenocam <- phenocam %>%
```

```
  mutate(year = year(datetime),
         month = month(datetime),
         DOY = yday(datetime))
```

```
phenocam
```

```
## # A tibble: 110,270 x 9
```

```
##   datetime          r_mean g_mean b_mean ir_mean NDVI_c year month DOY
##   <dtm>          <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 2017-02-24 17:15:05      66     57     33     93 -0.0136 2017     2    55
## 2 2017-02-24 17:30:05      67     56     31     92  0.0244 2017     2    55
## 3 2017-02-24 17:45:06      72     58     30     91  0.0893 2017     2    55
## 4 2017-02-24 18:00:05      77     58     28     87  0.179 2017     2    55
## 5 2017-02-24 18:15:06      66     67     42     85  0.0668 2017     2    55
## 6 2017-02-24 18:30:06      70     63     44     73 -0.792 2017     2    55
## 7 2017-02-24 18:45:06      38     24     30     66 -0.0219 2017     2    55
## 8 2017-02-24 19:00:06      21     12     15     19 -0.694 2017     2    55
## 9 2017-02-25 06:15:05      24     14     19     19 -0.909 2017     2    56
## 10 2017-02-25 06:30:05      48     37     43    104  0.127 2017     2    56
## # i 110,260 more rows
```

```
#e calculate mean NVDI per month per year
```

```
phenocam %>%
```

```
  group_by(year, month) %>%
  summarise(mean_NVDI = mean(NDVI_c, na.rm = TRUE))
```

```
## 'summarise()' has grouped output by 'year'. You can override using the
## '.groups' argument.
```

```
## # A tibble: 85 x 3
```

```
## # Groups:   year [8]
```

```
##   year month mean_NVDI
##   <dbl> <dbl>    <dbl>
## 1 2017     2 -0.0907
## 2 2017     3 -0.0802
## 3 2017     4 -0.0739
## 4 2017     5 -0.0734
## 5 2017     6 -0.0976
## 6 2017     7 -0.0810
## 7 2017     8  0.00581
## 8 2017     9 -0.0542
## 9 2017    10 -0.105
## 10 2017    11 -0.129
## # i 75 more rows
```