CSI 2300: Intro to Data Science

In-Class Exercise 10: Some tools for data wrangling.

1. In the Eagle Mountain data the **rep** function was used to repeat the dates several times. This function very flexible. Explain the difference between the examples:

```
rep(1:10, 5) #repeats the numbers 1-10 5 times
#  [1]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
# [26]  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10
rep(1:10, rep(5, 10)) #repeats the numbers 1-10 5 times
#  [1]  1  1  1  1  1  2  2  2  2  2  3  3  3  3  3  4  4  4  4  4  5  5  5  5  5
# [26]  6  6  6  6  6  7  7  7  7  7  8  8  8  8  8  9  9  9  9  9 10 10 10 10 10
rep(1:10, each = 5) #repeats the numbers 1-10 5 times
#  [1]  1  1  1  1  1  2  2  2  2  2  3  3  3  3  3  4  4  4  4  4  5  5  5  5  5
# [26]  6  6  6  6  6  7  7  7  7  7  8  8  8  8  8  9  9  9  9  9 10 10 10 10 10
rep(1:10, 1:10) #varies the number of times we'd repeat the numbers
#  [1]  1  2  2  3  3  3  4  4  4  4  5  5  5  5  5  6  6  6  6  6  6  7  7  7  7
# [26]  7  7  7  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9 10 10 10 10 10
# [51] 10 10 10 10 10

#TODO comment on what each line is doing
```

2. Suppose that **A** is a matrix, for example

```
A <- matrix(1:48, nrow = 6, ncol = 8) #syntax for a matrix
```

The R command

```
b <- c(A) #vector
#it stacks the columns
```

will convert this matrix into a vector. Does it do it by stacking rows or columns? What R code can you use to convert **b** back into **A**?

```
A <- matrix(b, 6, 8)
```

What happens if you apply the `c()` command to a dataframe?

```r
A_df <- as.data.frame(A)
b2 <- c(A_df)#This is a list as opposed to a vector

#NOTES: access pieces in a list
b2[1]#this is the info in column 1
# $V1
# [1] 1 2 3 4 5 6
b2[6]
# $V6
# [1] 31 32 33 34 35 36
```

3. Another way of formatting the Eagle Mountain Lake data is to order the observations with all the times for the first depth, then the second depth, and so forth. Create the correct depth and date columns for this type of ordering and include the temperature variable to create a 3 column data frame (the `rep()` and `seq()` functions may be useful here). To format the data this way, do you need to convert the original **temp** data frame to a matrix?

```r
temp <- read.csv('dat/EagleMountain/temp_through_09_12_2019.csv')
N <- nrow(temp)
#from the WIDE form
#vector the dates
date <- temp$DateTime
#vector with the depths
depth <- seq(0, 10, by = 0.5)
M <- length(depth)
#vector that's a smushed matrix of the temps


#to the LONG form
#big depths
bigDepths <- rep(depth, each = N)
#big dates
bigDates <- rep(date, M)
#vector that's a smushed matrix of the temps
bigTemp <- c(as.matrix(temp[,3:23]))
new.df <- data.frame(DateTime = bigDates,
                     depth = bigDepths,
                     temp = bigTemp)
```

4. Read in the wrangled and formatted data from the binary file `EML_through_09_12_2019.rda` using the load function. The .rda ending indicates this format. The first column are the dates, but are they date objects? Show how you can tell.

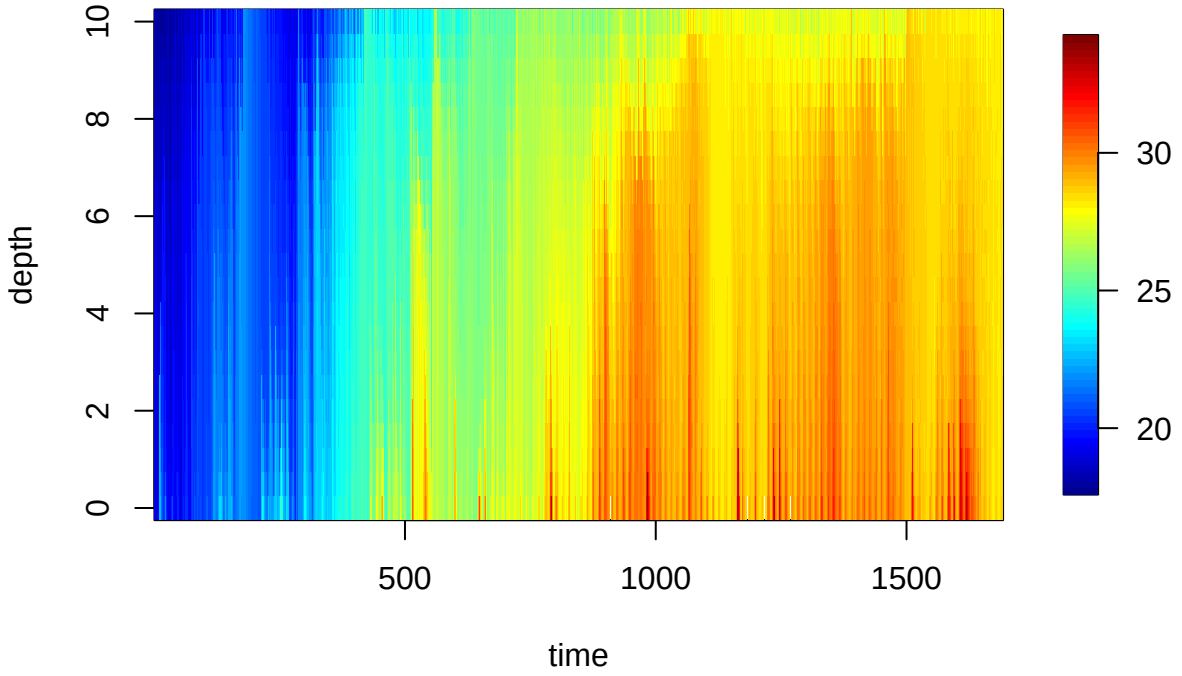
```r
library(lubridate)
#
# Attaching package: 'lubridate'
# The following objects are masked from 'package:base':
#
#     date, intersect, setdiff, union
load('dat/EML_through_09_12_2019.rda')
#day(all.data$DateTime)#this throws an error, bc col is a character

all.data$DateTime <- mdy_hm(all.data$DateTime)
#day(all.data$DateTime)
```

5. Here is a nice way to visualize data that has two dimensions, in this case time (about the first two years) and depth.

```r
library(fields)
# Loading required package: spam
# Spam version 2.11-1 (2025-01-20) is loaded.
# Type 'help( Spam)' or 'demo( spam)' for a short introduction
# and overview of this package.
# Help for individual functions is also obtained by adding the
# suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
#
# Attaching package: 'spam'
# The following objects are masked from 'package:base':
#
#     backsolve, forwardsolve
# Loading required package: viridisLite
#
# Try help(fields) to get started.
temp  <-read.csv(
  file = "dat/EagleMountain/temp_through_09_12_2019.csv", header=T)

tempImage <- as.matrix(temp[ ,(1:21)+ 2])
depth <- seq(0, 10, by=0.5)
time <- 1:nrow(tempImage)
image.plot(time, depth, tempImage)
```



How must the arguments to `image.plot()` be organized?

vector with time data and depth data (not replicated),third argument takes the matrix of temperature data.

Do you see any outliers or unusual time periods in these data to investigate? Is there any evidence of seasonality in these water temperatures?

There is a seasonality trend where red after 500 looks like hotter than usual weather.

water is cooler in April, warmer in the sumer months, the water gets cooler as yo go deeper, yellow streak at ~ 500 looks like an outlier