

## STA 234 HW-1

**Due Date: 2/4/2022 by 10:00PM**

### Problem 1 [30 points]

Report the output of the following R codes and explain what each part did.

```
1. c(1, FALSE)
2. c("a", 1)
3. c(list(1), "a")
4. c(TRUE, 1L)
5. seq(4,10)
6. seq(4,10,2)
7. oops = c(7,9,13)
   rep(oops,3)
8. rep(1:2,c(10,15))
9. x = matrix(1:12,nrow=3,byrow=T)
   t(x)
10. weight_kg <- c(60,72,57,90,95,72)
    height_m <- c(1.75,1.80,1.65,1.90,1.74,1.91)
    bmi <- weight_kg/(height_m^2)
    bmi>25
```

### Problem 2 [5 points]

Create a vector with three different names (characters).

## Factors

Factors are a special data structure for working with categorical data. Categorical data represents data that only differs by label (such 'yes'/'no') or ranks (such as '1st', '2nd', etc.). In R, factors are a special type of labeled integer vector. Factors are created with the `factor()` function. This function takes as arguments, a vector, the levels of the factor, and the labels of the factor. Think of factors as a way to label your data. Factors should only be used when you have a pre-determined number of categories. Next few problems will help you learn and practice factors in R.

### Problem 3 [15 points]

Report the outputs for f1 before and after modifying the levels of a factor.

```
f1 <- factor(letters)
levels(f1) <- rev(levels(f1))
```

What does the following code do? How do f2 and f3 differ from f1 in problem 3?

```
f2 <- rev(factor(letters))
f3 <- factor(letters, levels = rev(letters))
```

## Problem 4 [5 points] Practice

Creating a factor vector for days of the week as follows:

```
weekday_factor <- factor(c('M', 'T', 'W', 'Th', 'F', 'M', 'W'),  
                        levels = c('M', 'T', 'W', 'Th', 'F'),  
                        labels = c('Monday', 'Tuesday', 'Wednesday',  
                                  'Thursday', 'Friday'))  
weekday_factor
```

The `str()` function will tell us that the vector is factor, display some of the levels, and show the underlying mapping of levels to integer values that happened behind the scenes. The levels of a factor are always mapped to a sequence of numbers starting at 1 and increasing by 1 for every level. This mapping is based on the order in which the levels are entered in `factor()`

```
str(weekday_factor)
```

The `summary()` function will automatically count the occurrence of factor labels.

```
summary(weekday_factor)
```

Factors can also be created with numeric vectors as input. Let's say that we have a vector of 1s and 0s where 1 represents the occurrence of an event and 0 otherwise. The code below shows how to create a labeled factor from the data.

## Problem 5 [10 points]

```
event_indicator <- c(1, 0, 0, 1, 0, 0)  
  
event_fct <- factor(event_indicator,  
                   levels = c(0, 1),  
                   labels = c('No', 'Yes'))  
  
summary(event_fct)
```

Note that the order in which the levels are entered affects how they are stored in the factor.

```
event_fct_2 <- factor(event_indicator,  
                     levels = c(1, 0),  
                     labels = c('Yes', 'No'))  
  
summary(event_fct_2)
```

To access the levels of any factor and see their order, use the `levels()` function.

```
levels(event_fct)  
levels(event_fct_2)
```

By default, if we do not provide input to the `levels` and `labels` arguments in `factor()`, levels are automatically assigned in alphabetic order (for character vectors) or numeric order. The labels are then set to the levels values.

```
fct_from_chr <- factor(c('Yes', 'No', 'No', 'Yes'))
str(fct_from_chr)

fct_from_num <- factor(c(1, 1, 1, 4, 5))
str(fct_from_num)
```

### Problem 6 [15 points]

The survey vector below represents survey responses where people indicated their level of comfort with data analysis.

```
survey <- c(1, 3, 3, 2, 2, 1, 1, 1, 1)
```

The numeric values have the following meaning: 1 represents 'not comfortable' 2 represents 'moderately comfortable' 3 represents 'very comfortable'

Use the `factor()` function to label this vector. You should get the results below if you pass your factor into the `summary()` function.

not comfortable moderately comfortable very comfortable

5                      2                      2

### Problem 7 [20 points]

Suppose you track your commute times for 10 days and you find the following times in minutes: 17, 16, 20, 24, 22, 15, 21, 15, 17, 22.

- Enter this data into R and find the longest commute time (use function `max`), minimum commute time (use function `min`) and the average commute time (use function `mean`).
- Oops you realize that 24 was a mistake and it should have been 18. Correct one value in the data. Find the new maximum, minimum and average commute time after fixing this error.

**Project problem: [25 points]:** Share your broad areas of interest, and some specific research questions of interest in these areas with me. Investigate the datasets available in your area(s) of interest. I have shared data resources on Moodle Share the potential datasets with me.