| | |
|---|---|
| **CSDE 502 2021**<br>**Assignment 2**<br>**Data structures and tidyverse**<br>**Instructor: Phil Hurvitz**<br>**phurvitz@uw.edu** | **My Name: answer key**<br>**My UWNetID:** |

**Due Date: 2021-01-21 09:00**

**Instructions:**

1. Fill out your name and UWNetID at the top of this page.
2. Add your answers to this document.
3. Use the "00Answers" Word style for your answers so they will be clearly discernible from the questions.
4. When you are completed with your work, create a PDF from the Word document.
5. Name your PDF document with the pattern UWNetID_HW_n.pdf where UWNetID is your UWNetID and n is the week number when the homework was assigned. For example, the first assignment by me would be named *phurvitz_HW_01.pdf*. Upload your completed document to Canvas.
6. Upload any specified data or code files to Canvas. You should be uploading this document (completed) and an HTML file (output from a rendered Rmd file).

**Explanation:**

The questions here are to help you understand R data structures, to explore some of the `tidyverse` functions, and to practice editing R Markdown for generating informative outputs.

Feel free to use any reference materials at your disposal to answer these questions (i.e., do not consider the course materials a complete set of reference materials).

**Questions:**

1. Is `as.logical(NA)` a vector? What is its length?

**as.logical(NA) is a vector of length 1, as shown:**

```
> x <- as.logical(NA)
> class(x)
[1] "logical"
> str(x)
 logi NA
> is(x)
[1] "logical" "vector"
> length(x)
[1] 1
```

2. Is `as.logical(NULL)` a vector? What is its length?

**as.logical(NULL) is a vector of length 0, as shown:**

```
> y <- as.logical(NULL)
> class(y)
[1] "logical"
> str(y)
 logi(0)
```

```
> is(y)
[1] "logical" "vector"
> length(y)
[1] 0
```

3. The following code creates an object whose value is `NA`:

```
x <- NA
```

3.1. What type of data structure is `x`?

**x is a vector:**

```
> is(x)
[1] "logical" "vector"
```

3.2. What data type is `x`?

**x is a *logical* data type:**

```
> is(x)
[1] "logical" "vector"
```

3.3. How would you create an NA object with data type integer?

```
> x <- as.integer(NA)
> is(x)
[1] "integer"              "double"               "numeric"
[4] "vector"               "data.frameRowLabels"
```

**Creating NA values with specific data type is important for combining data. For example, if some values were instantiated using "x <- NA", the data type is logical, and in some instances, attempting to combine these values with existing data can induce failures. For example, adding records from R to an SQL database in which an existing column is type "float" (i.e., "numeric" in R), but records to be added included logical NAs.**

4. Explain how a data frame in R is a type of R list.

**A data frame in R has a row x column rectangular structure, similar to a matrix. However, a matrix can only have columns of one data type, whereas a data frame can have columns that are different data types. Similarly, a list can have multiple data types. Indeed, a data frame *is* a type of list.**

**Looking at the built-in *iris* data frame, this shows that it is a list of 5 elements:**

```
> is(iris)
[1] "data.frame" "list"       "oldClass"   "vector"
> is.list(iris)
[1] TRUE
> length(iris)
[1] 5
```

**Each column is a vector and a list element. If we get the first list element's first element, we get the first cell of the data frame.**

```
> iris[[1]][1]
[1] 5.1
> head(iris, 1)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
```

**Compare this with a matrix; a 10 x 10 matrix has length of 100:**

```
> m <- matrix(1:100, nrow = 10)
> m
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    1   11   21   31   41   51   61   71   81    91
 [2,]    2   12   22   32   42   52   62   72   82    92
 [3,]    3   13   23   33   43   53   63   73   83    93
 [4,]    4   14   24   34   44   54   64   74   84    94
 [5,]    5   15   25   35   45   55   65   75   85    95
 [6,]    6   16   26   36   46   56   66   76   86    96
 [7,]    7   17   27   37   47   57   67   77   87    97
 [8,]    8   18   28   38   48   58   68   78   88    98
 [9,]    9   19   29   39   49   59   69   79   89    99
[10,]   10   20   30   40   50   60   70   80   90   100
> length(m)
[1] 100
```

5. Use the R Markdown file generated at the end of last week's class session as an example/template for presenting the following results. The output HTML file should contain all of the code that generated the output as well as the requested output. Use `tidyverse` functions as much as possible to generate readable code. Include commentary as necessary for interpretation of your methods and/or results.

   Using the built-in `iris` data frame:

   5.1. Present three tables that have all of the original variables, one for each species of iris.
   5.2. Present a table with only sepal length and sepal width for spp. *virginica* that also has columns indicating whether the sepal length and sepal width are greater than the mean for this species.

   Use the general health of respondents, question "S3Q1 GENERAL HEALTH-W1" from the Add Health table we used in class:

   5.3. Create a new variable that classifies health, stratified at the break between "(2) Very good" and better versus "(3) Good" and worse. Make sure you explicitly handle missing or unknown values.
   5.4. Tabulate (count and percent) this new variable for all respondents.
   5.5. Tabulate (count and percent) this new variable for those who self-identified as White versus those who self-identified as African American.