

Assignment 1 solutions

BIOSTAT 213: Introduction to Computing in the R Software Environment

Due: Wednesday, October 9, 2019

Collaboration with other students in this course is permitted on this assignment. However, you must write all work—including R code—in your own voice, as much as possible, and acknowledge your collaborators as thoroughly as possible.

The assignment is due online, in the [Collaborative Learning Environment](#), by 11:59 pm, Wednesday, October 9, 2019. You must submit a Word-compatible document or a PDF file.

For each of the below, share your code as well as the output (if there is any output). If your answer to a question is unclear from your R output, or difficult to identify, include an answer in text form (in addition to the code and output).

1. Imagine that you are trying to estimate an individual's body mass index (BMI). The individual is in your office. You measure the individual to weigh 140 pounds. However, for whatever reason, though your office has a scale, it lacks a measurement instrument for height/length. Using your experience (and perhaps your own height), you guess the individual to be between 5 ft 7 in and 5 ft 10 in tall. Because BMI will necessarily decrease as height increases (in mathematical terms: the relationship is monotonic), you can use this range to calculate a range for the individual's BMI. Write code to do this, all in one line and without conducting any math outside of R (whether in your head or otherwise). BMI is defined as: $(\text{pounds})/(\text{inches})^2 \times 703$.

```
> 140/c(5*12+10,5*12+7)^2*703
[1] 20.08571 21.92470
```

2. Write one line of code that will output every multiple of 5, beginning with 0 and ending with 100.

```
> seq(0,100,5)
[1] 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80
[18] 85 90 95 100
```

3. A common tongue twister is to try to say “toy boat” quickly many times. Write code that will output 10 iterations of the phrase. Your code should output 1 element with a space between the phrases, not a vector with 10 elements.

```
> paste(rep('toy boat',10),collapse=' ')
[1] "toy boat toy boat toy boat toy boat toy boat toy boat toy boat toy boat toy boat toy boat"
```

You will be using the `presidencies.csv` data file for the following problems. The data set contains data on prior US presidents.

4. Without counting, how many unique birth states (`state`) have the presidents represented? You should use only one line of code. In case you noticed and are curious: one state is “South/North Carolina.” You may and should count this as a unique state.

```
> length(unique(ii$state))
[1] 21
```

5. In a single line, write code that will return the number of characters in each variable name.

```
> nchar(names(ii))
[1] 5 4 5 3 5 3 5
```

6. Write code that will test whether any presidents were born in the state of Oregon. Your code should return the output `FALSE`. You should use only one line of code.

```
> is.element('Oregon',ii$state)
[1] FALSE
```

You will be using the `nhanes.csv` data file for the following problems.

7. The variable `seqn` represents a unique ID. Write code that will return the total number of unique IDs. Write code that will return the total number of rows in the data set. Verify, in your head, that these numbers match. The output in both cases should be 4209.

```
> length(unique(ii$seqn))
[1] 4209
> nrow(ii)
[1] 4209
```

8. Add a new ID variable to the data set. Name the new variable `id`. The values of `id` should be the sequential integers, beginning with 1. For example, the first value should be 1, the second value should be 2, and so forth. Use `head()` and `tail()` to verify your work. For example: `head(ii$id)`. You should use only one line of code.

```
> ii$id<-1:nrow(ii)
> head(ii$id)
[1] 1 2 3 4 5 6
> tail(ii$id)
[1] 4204 4205 4206 4207 4208 4209
```

9. The variable `ridageyr` stores age in years. You can use `head()` to display the first few values of the variable. For example: `head(ii$ridageyr)`. Write code that will output these values *sorted in decreasing order*. You should use only one line of code.

```
> sort(head(ii$ridageyr),decreasing=TRUE)
[1] 62 56 53 42 32 22
```