NR 592
Assignment 1

INSERT YOUR NAME HERE

INSERT THE DATE HERE

Last Updated: 2019-03-28

Instructions

Show your work. The response to each question should include all of the commands necessary to reproduce your results, and should be commented to give a naive reader an idea of the intent behind each section of code. Answers to questions should be clearly shown in the code output or spelled out in text or comments.

Before submitting your assignment, ensure that your code for each question works correctly when the lines are run in sequence, starting with an empty environment. You may submit either this R Markdown document with your code entered into the appropriate code blocks, or a regular R script with the question numbers as comments and the answers following. If you choose to Knit the R Markdown document to HTML or PDF, include those files with your submission (but it is not required that it Knit successfully).

Question 1

1.1

Create a vector named `v1` that contains the integers 1 through 5 (without using the `seq()` function).

1.2

Using the `seq()` function, create a vector named `v2` that contains the integers 6 through 10.

Divide every number in `v2` by 2 and reassign the result to `v2`.

1.3

Square each number in `v1` and assign the result to a vector called `v1sq`.

Add `v1sq` to `v2` and assign the result to `v3`.

Remove `v1sq` from the environment.

1.4

Inspect the class of each vector (`v1`, `v2`, and `v3`). If any of these vectors are not of the integer class, print (show) their values, coerce (change) their class to integer, and reassign them under the same names.

1.5

Print the new values of any vector that was coerced to integer.

What changed when these values were coerced to integers?

Why would it be dangerous to coerce real data without inspecting it first?

`Delete this and enter your response here.`

1.6

Find and use a function that will reverse the order of `v3`, and reassign the result to `v3`.

1.7

Coerce `v1` to an ordered factor (ordinal variable). Inspect the structure of `v1`.

How many levels does `v1` have? Which is the lowest level?

`Delete this and enter your response here.`

1.8

Identify which values of `v1` are less than the corresponding values of `v2`.

Question 2

2.1

Create a matrix named `m1`, using `v1`, `v2`, and `v3` as the three columns of that matrix.

Print the resulting matrix.

(Hint: When you're finished, the first row should read `1  3  30`.)

2.2

Create a vector named `m1cols` containing the strings "age", "height", and "pred".

Coerce `m1` to a data frame named `df1` and assign `m1cols` as its column names.

Print the resulting data frame.

2.3

Inspect the structure of `df1`.

If the classes of any of the columns are different than the vectors they were created from, coerce them back to their original types.

2.4

Add a new column to `df1` called "id". The values of this column should be strings, made up of the letter "a" followed by the number "0" and the corresponding value of the `age` column. (Hint: Check out the documentation for the `paste()` function, specifically `paste0()`.)

Coerce the `id` column to a factor (nominal/categorical variable).

2.5

Rearrange `df1` so that `id` is the first column, followed by `age`, `height`, and `pred`.

Make sure that the class of each column is the same as it was before being reordered.

Question 3 (Extra Credit)

3.1

Write a function to calculate and return the mean of any five input numbers (without using the base `mean()` function).

(Hint: You can write your function to handle either: one input variable that is a vector of five numbers; or five different input variables, each of which is a number. Both could work. Read the next part of the problem and think about which will be easier for your input data.)

3.2

Plug each variable (column) from your `df1` data frame (except `id`) into your function to show that it works. For practice doing both, reference some of these variables by column number and some by column name.

(Hint: You may have to coerce the class of one of your variables to numeric when you plug it in.)

3.3

Write a loop that will do the following:

- For each column number in `df1` after `id` until the last column...
  - Store the column's values in a vector called `varVals`.
  - Check if `varVals` is numeric...
    - If not, coerce `varVals` to numeric and reassign.
  - Run your function to calculate the mean of `varVals` and store the value in a variable called `varMean`.
  - Print a string that reads "Variable ", followed by the value of the column number, followed by a colon (":").
  - Print the value of `varMean`.

When it is finished, your output should look like this:

```
Variable 2:
3
Variable 3:
```

```
3.8
Variable 4:
14.8
```

(pdf / Rmd)