Introduction to R

Session 4 exercises

Statistical Consulting Centre

1 March, 2017

1. Generate a one-way frequency table for `q1a`.
2. Create a new variable called `q1a.sc` (meaning *q1a score*), where `q1a.sc` is of type numeric/integer rather than of type `factor`.
3. Generate a one-way frequency table of `q1a.sc` and compare it with the one you generated in question 1. Their frequencies should be identical.
4. Repeat the steps in questions 1 – 3 for variables `q1b` to `q1e`, thereby creating new variables `q1b.sc` – `q1e`.
5. Create a data frame called `mean.df` containing all five score variables (`q1a.sc` – `q1e.sc`) which you've created.
6. Use `apply()` on `mean.df` to calculate each participant's mean score across variables `q1a.sc` – `q1e.sc`. Name this new variable `nerdy.sc`, meaning *nerdy score*.
7. Add the variable `nerdy.sc` to the `mean.df` data frame and use `summary()` to generate the five-number-summary of all *six* variables in `mean.df`.
8. Add the columns of `nerdy.sc` to `sports.df` for future use.
9. Use `tapply()` to calculate the mean nerdy score for all ten income levels.
10. Income level 1 is shown first in the output of question 9 while income level 10 is shown last. Do you agree with R's default ordering of `income` levels? If not, appropriately order the levels of `Income`.
11. Repeat question 9 to check that *your chosen* ordering of `Income` levels has been correctly set.
12. You were introduced to the following function, `mytab()`, in the Session 4 lecture slides.

```
mytab <- function(someinput){
 n <- length(someinput)
 n.missing <- na.check(someinput)
 n.complete <- n - n.missing
 mymean <- round(mean(someinput, na.rm = T), 2)
 mysd <- round(sd(someinput, na.rm = T), 2)
 mystder <- round(mysd/sqrt(n.complete), 2)
 Lower.CI <- round(mymean - 1.96*mystder, 2)
 Upper.CI <- round(mymean + 1.96*mystder, 2)
```

```
 c(Complete.obs = n.complete, Missing.obs = n.missing,
   Mean = mymean, Std.Error = mystder,
   Lower.CI = Lower.CI, Upper.CI = Upper.CI)
}
```

It depends on the `na.check()` function, defined earlier, to calculate the number of missing values, i.e., `mytab()` depends on the availability of `na.check()` in order for it to work. Modify `mytab()` so it does *no longer* depends on `na.check()` to calculate the number of missing values. Let's call the modified function `mytab1()`.

13. Use `mytab1()` to produce a summary table for all six variables in `mean.df`.

14. Use `mytab1()` to produce a summary table of nerdy scores for all ten income levels.