

Module 8: exercise solutions

BIOSTAT 213: Introduction to Computing in the R Software Environment

```
> ii<-read.csv('nhanes.csv')
```

1. Use `tapply()` to find the median systolic blood pressure (`bpxsy1`) in each `bmecat` group. Preferably, first re-order the levels of `bmecat` so they are in scientific order.

```
> ii$bmecat<-factor(ii$bmecat,levels(ii$bmecat)[c(5,4,1:3)])
> tapply(ii$bpxsy1,ii$bmecat,median,na.rm=TRUE)
Underweight or normal weight      Overweight
                        116                120
      Class I obesity      Class II obesity
                        122                124
      Class III obesity
                        126
```

2. Use one of the apply functions to find the maximum systolic blood pressure among the 4 possible measurements (`bpxsy1`, `bpxsy2`, `bpxsy3`, and `bpxsy4`) for the first 10 individuals in the data. You should obtain 10 values (each represents the maximum measurement for a unique individual). Hint: you may want to include indexing.

```
> apply(ii[1:10,c('bpxsy1','bpxsy2','bpxsy3')],2,max,na.rm=TRUE)
bpxsy1 bpxsy2 bpxsy3
    178    180    180
```

I modified this to include just the first 3 variables, since all 10 of the first 10 individuals are missing `bpxsy4`.

3. Write a function that implements the [quadratic formula](#):

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The symbol $\pm$ means plus or minus. For example, 2 ± 3 means $2 - 3$ and $2 + 3$.

The function should accept 3 arguments and output 1 value.

Check your function with a few values of a , b , and c .

```
quadratic<-function(a,b,c){
  left<--b/(2*a)
  right<-sqrt(b^2-4*a*c)/(2*a)
  c(left-right,left+right)
}
```

```
> quadratic(5,6,1)
[1] -1.0 -0.2
```

4. Use `tapply()` to find the median systolic blood pressure (`bpxsy1`) in each `bpcat` group. Preferably, first re-order the levels of `bpcat` so they are in scientific order. Note that `bpcat` depends on `bpxsy1` (as well as `bpxdi1`) so you should notice a pattern.

```
> ii$bpcat<-relevel(ii$bpcat,'Normal')
> tapply(ii$bpxsy1,ii$bpcat,median,na.rm=TRUE)
      Normal      Elevated Hypertension
        110         124         136
```

5. Use one of the apply functions to output the number of missing values in each variable.

```
> apply(ii,2,function(cc){sum(is.na(cc))})
```

6. The `format()` function can be used to format dates. For example:

```
> format(Sys.Date(), '%Y-%m-%d')
[1] "2019-11-22"
```

Write a function that will output a formatted version of `Sys.Date()` (a function that outputs the current date). The function should accept an argument that indicates whether the date should be outputted in a month-day-year format (as in the United States) or a day-month-year format (the convention in most other places). You might consider the [list of codes](#) shared in Module 7. Here's an example function:

```
> today()
[1] "11/22/2019"
> today(us=FALSE)
[1] "22 November 2019"
```

```
today<-function(us=TRUE){
  if(us){
    format(Sys.Date(), '%m/%d/%Y')
  } else {
    format(Sys.Date(), '%d %B %Y')
  }
}
```