Introduction to R

Answers to Session 3 exercises

Statistical Consulting Centre

1 March, 2017

1 Missing values

(i) In question 3(ii) of exercise 2 you identified the "Can't choose" cases in `q1a`. Now, replace these cases by `NA`.

```
exclude.q1a <- which(sports.df$q1a ==  "Can?t choose")
sports.df$q1a[exclude.q1a] = NA
```

(ii) Repeat 1(i) for `q1b` – `q1e`, so that all cases of "Can't choose" are replaced by `NA`.

```
exclude.q1b <- which(sports.df$q1b ==  "Can?t choose")
sports.df$q1b[exclude.q1b] <- NA
exclude.q1c <- which(sports.df$q1c ==  "Can?t choose")
sports.df$q1c[exclude.q1c] <- NA
exclude.q1d <- which(sports.df$q1d ==  "Can?t choose")
sports.df$q1d[exclude.q1d] <- NA
exclude.q1e <- which(sports.df$q1e ==  "Can?t choose")
sports.df$q1e[exclude.q1e] <- NA
```

(iii) Produce a one-way frequency table of `ethnicity`.

```
table(sports.df$ethnicity)


  China,Cantonese,Hakka,Mandarin            Europe,White/European
                              19                              817
 India,Hindi,Urdu,Gujarati,Tamil                Maori+New Zealand
                               8                               90
                   NA, dont know               Other,mixed origin
                              11                               26
PACIFIC,Polynesian,Chamorro/Guam
                              25
```

(iv) Repeat **1**(iii) after replacing all cases of "NA, dont know" with `NA`.

```
exclude.ethnicity <- which(sports.df$ethnicity == "NA, dont know")
sports.df$ethnicity[exclude.ethnicity] <- NA
```

(v) There are only two possible values for `partner`: `Yes` and `No`. Replace any values which are not `Yes` or `No` with `NA`.

```
exclude.partner <- with(sports.df, which(partner != "Yes" & partner !="No"))
sports.df$partner[exclude.partner] <- NA
```

2 Factor

(i) Produce a two-way frequency table of `q1a` versus `gender`.

```
with(sports.df, table(q1a, gender))


                                    gender
q1a                                  Female Male
  Cant choose                           112  123
  Daily                                   1    1
  Several times a month                  44   22
  Several times a week                    4    4
  Several times a year or less often    352  297
```

(ii) Table 1 shows the appropriate ordering of the levels of the values in `q1a` – `q1e`.

Table 1: The right levels for `q1a` to `q1e`

q1a	Factor(q1a)
Daily	1
Several times a week	2
Several times a month	3
Several times a year or less often	4

Convert `q1a` – `q1e` into factors with their levels ordered as shown in Table 1. Then generate two-way frequency tables between `q1a` to `q1e`, respectively, versus `gender` to check that you've appropriately ordered these factors' levels.

```
sports.df$q1a <- factor(sports.df$q1a,
                        levels = c("Daily", "Several times a week",
                                   "Several times a month",
                                   "Several times a year or less often"))
with(sports.df, table(q1a, gender))


                                    gender
q1a                                  Female Male
  Daily                                   1    1
  Several times a week                      4    4
  Several times a month                    44   22
  Several times a year or less often      352  297

sports.df$q1b <- factor(sports.df$q1b,
                        levels = c("Daily", "Several times a week",
                                   "Several times a month",
                                   "Several times a year or less often"))
with(sports.df, table(q1b, gender))

                                    gender
q1b                                  Female Male
  Daily                                   3    0
  Several times a week                    3    1
  Several times a month                  39   24
  Several times a year or less often    362  287

sports.df$q1c <- factor(sports.df$q1c,
                        levels = c("Daily", "Several times a week",
                                   "Several times a month",
                                   "Several times a year or less often"))
with(sports.df, table(q1c, gender))

                                    gender
q1c                                  Female Male
  Daily                                  21   14
  Several times a week                  184   93
  Several times a month                 224  213
  Several times a year or less often     81  113

sports.df$q1d <- factor(sports.df$q1d,
                        levels = c("Daily", "Several times a week",
                                   "Several times a month",
                                   "Several times a year or less often"))
with(sports.df, table(q1d, gender))

                                    gender
q1d                                  Female Male
  Daily                                  34   28
  Several times a week                  159  122
  Several times a month                 240  200
  Several times a year or less often     80   97

sports.df$q1e <- factor(sports.df$q1e,
                        levels = c("Daily", "Several times a week",
                                   "Several times a month",
                                   "Several times a year or less often"))
with(sports.df, table(q1e, gender))
```

```
                                    gender
q1e                                  Female Male
  Daily                                 136  108
  Several times a week                  220  151
  Several times a month                  89   98
  Several times a year or less often     48   62
```

(iii) Create a new variable which categorises all participants into one of three age groups: “Under 40”, “41 to 60” and “Over 61”.

```
age.group <- with(sports.df, ifelse(age <= 40, "Under 40",
                             ifelse(age > 40 & age <=60, "41 to 60", "Over 61")))
```

(iv) Convert the variable created in **2**(iii) into factors with appropriate levels.

```
age.group <- factor(age.group, levels = c("Under 40", "41 to 60", "Over 61"))
```

(v) Add the factor into `sports.df` and name it `age.group`

```
sports.df$age.group <- age.group
```

3 Challenge

We mentioned in Exercise 2 that the function `mystder` calculates the standard error of the mean (SEM), i.e.

```
mystder <- function(x){
       mysd <- sd(x, na.rm = T)
       n <- length(x)
       mysd/sqrt(n)
}
```

This function only calculates the standard error correctly if the input does NOT contain missing values. This is because the `length()` function counts the number of elements in the variable, including missing values. For example:

```
test <- c(1, 2, 3, 4, NA)
length(test)

[1] 5
```

So, `length(test)` returns 5 instead of 4. Suppose you repeat an experiment 5 times, resulting in one missing value; your real/valid sample size is 4. Thus, when you calclate your standard error, use $n = 4$ instead of 5. For example,

```
mysd <- sd(test, na.rm = T)
mysd

[1] 1.290994

n <- 4
n

[1] 4

mysd/sqrt(n)

[1] 0.6454972
```

The real SEM for `test` should be 0.6454972; however, if we use `mystder()` to calculate it we get:

```
mystder(test)

[1] 0.5773503
```

Thus, calculating the sample size using `length()` will lead to an incorrect solution when there are missing values in the data.

(i) Now that you know what is wrong with `mystder()`, modify it so it gives the correct SEM even if the input contains missing values.

```
#There are many many ways of doing this. Here is just one example:
mystder <- function(x){
       mysd <- sd(x, na.rm = T)
       n <- sum(!is.na(x))
       mysd/sqrt(n)
}
```

(ii) Apply your modified `mystder` function to `test` to see whether it returns the correct answer, i.e. 0.6454972.

```
mystder(test)


[1] 0.6454972
```

(iii) Create `test2`, as shown below, and test your function on this new variable.

```
test2 <- c(1:100, rep(NA, 30))
mystder(test2)


[1] 2.901149
```

The correct value for the SEM should be 2.9011492.