Introduction to R

Session 2 – Data subsetting

Statistical Consulting Centre

19 July, 2017

1. Installing an R package

R packages are collections of user-defined functions. The function `std.error`, for example, is contained in the `plotrix` package.

1. Let's look at what happens when we try to use a function before actually installing on our computer the package in which it is contained. E.g. Calculate the SEM of age using `std.error`.

```
std.error(lake.df$pH)
```

```
## Error in std.error(lake.df$pH): could not find function "std.error"
```

2. Install the package `plotrix` while in your R session by following the instructions below:
   (a) Select `Packages` from the bottom right panel of your Rstudio interface.
   (b) Click on the `Install Packages` icon just below `Packages`.
   (c) Type `plotrix` in the blank space provided below "`Packages (separate multiple with space or comma):`"
   (d) Click on `No` if you are asked you to restart R.
   (e) Submit the code `library(plotrix)` to the R console to make the functions contained in `plotrix` available in the current R session.

```
library(plotrix)
```

3. Now, use `std.error` to calculate the standard error of the pH.

```
std.error(lake.df$pH)
```

```
## [1] 0.1769821
```

4. Try writing your own code to calculate the standard error of the pH. Hint: This only requires one line of code. Use online resources if you cannot remember how the SEM is calculated.

```
with(lake.df, sd(pH, na.rm = TRUE)/sqrt(length(pH)))
```

```
## [1] 0.1769821
```

2. Write your own function

1. In Session 2 you were shown a simple function to calculate the standard error of the mean (SEM), i.e.

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

Type the above code into your R script and submit it to the R console.

2. Modify the function in 2.1 so that the output will have only 2 decimal places.

```
mystder <- function(x){
       mysd <- sd(x, na.rm = TRUE)
       n <- length(x)
       round(mysd/sqrt(n), 2)
}
```

3. Calculate the SEM of pH using the function you created in 2.2.

```
mystder(lake.df$pH)
```

```
## [1] 0.18
```

2. Subsetting datasets

1. Print the following to the console:

- The pH of the first lake.

```
lake.df$pH[1]
```

```
## [1] 6.1
```

- The pH of the last lake.

```
lake.df$pH[53]
```

```
## [1] 7.9
```

```
#or:
lake.df$pH[nrow(lake.df)]
```

```
## [1] 7.9
```

- The pH values of the first and last lakes.

```
lake.df$pH[c(1, 53)]
```

```
## [1] 6.1 7.9
```

- All measurements made on the third lake.

```
lake.df[3, ]
```

```
##   ID   Lake  pH Calcium Chlorophyll
## 3  3 Apopka 9.1    High       128.3
```

- All pH values.

```
lake.df[, "pH"]
```

```
##  [1] 6.1 5.1 9.1 6.9 4.6 7.3 5.4 8.1 5.8 6.4 5.4 7.2 7.2 5.8 7.6 8.2 8.7
## [18] 7.8 5.8 6.7 4.4 6.7 6.1 6.9 5.5 6.9 7.3 4.5 4.8 5.8 7.8 7.4 3.6 4.4
## [35] 7.9 7.1 6.8 8.4 7.0 7.5 7.0 6.8 5.9 8.3 6.7 6.2 6.2 8.9 4.3 7.0 6.9
## [52] 5.2 7.9
```

2. Calculate:

- The average pH of lakes with low Calcium concentrations.

```r
with(lake.df, mean(pH[Calcium == "Low"]))
```

```
## [1] 5.344444
```

- The average pH of lakes with low Calcium concentrations and Chlorophyll concentrations lower than 10.

```r
with(lake.df, mean(pH[Calcium == "Low" & Chlorophyll < 10]))
```

```
## [1] 4.933333
```

4. Challenge

Modify the function given in 2., so that the function will return a 95% confidence interval (with 2 decimal places). Hint: A 95% confidence interval of a variable `x` is given by the mean of `x` $\pm$ 1.96 $\times$ SEM of `x`. You might find the `paste()` function useful.

```r
mystder <- function(x){
       mymean <- mean(x, na.rm = TRUE)
       mysd <- sd(x, na.rm = TRUE)
       n <- length(x)
       mystder = mysd/sqrt(n)
       upperCI = round(mymean + 1.96*mystder, 2)
       lowerCI = round(mymean - 1.96*mystder, 2)
       paste("(", lowerCI, " , ", upperCI, ")", sep = "")
}
mystder(lake.df$pH)
```

```
## [1] "(6.24 , 6.94)"
```