1B:Intro_to_RMarkdown

2024-09-16

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
depth <- 10
```

Writing and running R

You can type code directly into your console at the > prompt, like we did to install the tidyverse package in the previous document.

Let's take another example. Say we have a measurement that was taken in the water column at 10 m depth. We need to tell R the water depth was 10 m. We do this by typing the following straight into the R terminal:

```
depth <- 10
```

Alternatively, you can type it inline within this R markdown document, which allows you to take notes and run code within the same document. To do so, you'll write your code like this:

```
depth <- 10
```

Place your cursor in the cell, on top of your command, and press "command-return" on a mac, or "control-enter" on a PC to run the line. You'll see that the command is sent to the R console.

In R-talk we have "assigned" the value of 10 to the "object" called depth. Note, many other programming languages use "variable" in place of object and I may slip into that vocabulary.

The power of using R markdown here is that we can check our code and print out results directly into our document. For example, if we want to check what the value of 'depth' is at this point in our workflow we can print it into the document like this, and when we run this line, you can see a print out of the result below it:

```
print(depth)
```

```
## [1] 10
```

In R markdown, you can also print an object to your notebook without using the print function:

```
depth
```

```
## [1] 10
```

Let's record another measurement at a second depth of 50 m, and calculate our mean depth measurement:

```
depth2 <- 50
meanDepth <- (depth+depth2)/2
meanDepth
```

```
## [1] 30
```

What if we took a measurement every 10m from the surface (i.e. depth = 0 m) to 100 m? Do we want to have 11 different objects each containing a different depth?

What we can do is create a vector, which contains all the depth information in one object.

You can think of vectors as a list or series of objects. They are written with a 'c' followed by a parenthesis and a series of values separated by commas:

```
depths <- c(0,10,20,30,40,50,60,70,80,90,100)
depths
```

```
##  [1]   0  10  20  30  40  50  60  70  80  90 100
```

There's a function we can use to make writing this vector easier for ourselves, since there is a pattern to the depths we are reporting. We'll use the 'seq' function.

```
depths <- seq(0,100,10)
depths
```

```
##  [1]   0  10  20  30  40  50  60  70  80  90 100
```

The function is working like so:

First you call the function `seq`

Then, you put in the parameters, in this case, the first value is the starting number ('0' in this example), the second value is the ending number ('100' in this example) and the third value is the value by which to space the numbers in the sequence ('10' in this example')