USE R! HAVE FUN! _ Solutions to week 1 exercises

Hey guys! Sorry about the lateness, because I get a serious stomachache so I cannot get down to work until this afternoon. I hope that you have already finished the exercise. And do feel free to argue with me.

And game time now!

Prepared by: Yuan Tian(2016-4-17)

Solutions to week 1 exercises

1. Besides R, there are a lot of software tools used in data science. Can you list **2** other tools? What are their advantages and disadvantages? And in which industries are those tools often used?

- SPSS
  Advantages: SPSS has a user friendly UI(User interface), and the outputs are usually good looking with even default options.
  Disadvantages: SPSS is not so compatible with office software. And typically a UI will reduce the reproducibility of your analysis.
  Mainly applied: Sociology, economy.
- SAS
  Advantages: SAS has dozens of portable packages suitable for different functions. It is also capable for operations research questions.
  Disadvatages: The license for SAS is rather expensive.
  Mainly applied: Almost every area.
- Minitab
  Advatages: Minitab has a user friendly UI, while capable for lots of questions in quality control.
  Disadvantages: The apperance of Minitab's outputs are usually not good-looking. And a UI will reduce the reproducibility of your analysis.

2. Install R on your computer. Download R (https://www.r-project.org/).
   Just do it!

3. RStudio is a powerful and productive user interface for R. I highly recommend you to install RStudio. Download RStudio (https://www.rstudio.com/products/rstudio/download/).
   Just do it!

4. You can also download the R tutorials from R-bloggers (http://www.r-bloggers.com/).
   Just do it!

5. Please find out how many data types are legal in R. List all of them along with their features. **There are at least 10 of them.**

- Logical

- Integer
- Doulbe
- Complex
- Character
- Matrix
- Factor
- List
- NA
- NaH
- Data.frame

6. Using your knowledge about data type, type the following code:

```
x <- c(2)
y <- as.logical(x)
z <- as.character(y)
x
```

```
## [1] 2
```

```
y
```

```
## [1] TRUE
```

```
z
```

```
## [1] "TRUE"
```

x = 2 because the code “c(2)” means creating a number 2, and the operator “<-” means assigning 2 to x.
y = TRUE because we coerce 2 to a logical number, namely “True” or “False”
z = “TRUE” because we coerce y to a character, which means we store the character “TRUE” in z.
Then type the following code:

```
x <- 1:6
class(x)
dim(x) = c(3,2)
class(x)
```

Why is the class of x changed?
This is because we coerce a vector into a matrix by assigning its dimensions.
7. Study the control structures of R (if else, for loops, while loops, repeat, next, break).
—————
if(your condition)
{

your code
}
else
{
your code
}
—————
for(your condition)
{
your code
}
————— while(your condition)
{
your code
}
—————
continue
* Stop the current loop, and excute the next loop.*
—————
break
* Stop the whole loop.*

8. Using the knowledge you’ve learned, program a function that can do the division. Remember that the denominator cannot be zero. (I understand this is may be a bit tough, so fill your code in the following example).

```
division <- function(nominator, denominator){
  if(denominator == 0)
  {
    return("The denominator cannot be zero")
  }
  else
  {
      return(nominator/denominator)
    }
}
```

To test your function, type in “division(2, 3)”, “division(2, 0)”. The results will be 0.6667 and “Denominator is zero”.

9. Use your **division** function to solve the following equations:
- 10/2
- 6/5
- 9/0

```
  division(10,2)
```

```
## [1] 5
```

```
division(6,5)
```

```
## [1] 1.2
```

```
division(9,0)
```

```
## [1] "The denominator cannot be zero"
```

10. Program a function called **innerproduct**, whose purpose is to calculate the inner product of two vectors. To practice the use of **for** loops, try both multiplication and summation operations. **Do make sure the two vectors are of the same length.** Test your function. Tips: you can generate a vector in R by typing the following:

```
x <- c(1, 2, 3, 4, 5)
```

```
innerproduct <- function(a,b){
  if(length(a)!= length(b))
  return("The two vectors are not of the same length")
  else
  {sum = 0
    for(i in 1:length(a)){
      sum = sum + a[i]*b[i]
      i = i + 1}
  }
  return(sum)
}
x <- c(1, 2 ,3)
y <- c(3, 2, 1)
z <- c(1, 2, 3, 4)
innerproduct(x, y)
```

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

```
innerproduct(x, z)
```

```
## [1] "The two vectors are not of the same length"
```