QUIZ #1 SOLUTIONS

MSAN 593

July 19, 2018

Instructions: No computer, no notes or electronic devices permitted in this quiz. You may only use a pencil and eraser or pen. You have 30 minutes to complete the quiz.

Question 1 (2 pts)

What a line of code that creates a vector of the integer numbers 1 to 5 (inclusive), storing it in the variable `myIntegers`.

```
myIntegers <- c(1, 2, 3, 4, 5) #although technically this generates a double
```

OR

```
myIntegers <- 1:5
```

Question 2 (2 pts)

```
myAtomicVector_01 <- c(99, 98, 97, 96)
typeof(myAtomicVector_01)
```

What does the last line of code return?

```
myAtomicVector_01 <- c(99, 98, 97, 96)
typeof(myAtomicVector_01)
```

```
## [1] "double"
```

Question 3 (2 pts)

```
myAtomicVector_01 <- c(99, 98, 97, 96)
class(myAtomicVector_01)
```

What does the last line of code return?

```
myAtomicVector_01 <- c(99, 98, 97, 96)
class(myAtomicVector_01)
```

```
## [1] "numeric"
```

Question 4 (2 pts)

What are two reasons I suggested to you to use `R` (other than the fact that you are required to use `R` in MSAN 593)?

- Free
- Deep set of statistical packages

Question 5 (2 pts)

If you wanted to call a function from a package that **is downloaded** but **not loaded**, how would you accomplish this? In your answer, write the actual code you would use if you want to call a function named `myFunction` from a package named `myPackage`, where `myPackage` is not loaded into memory but is downloaded.

```
myPackage::myFunction
```

Question 6 (2 pts)

What is the difference between the functions `require()` and `library()`? Be explicit.

1. `require()` will throw a **warning** if the package is not downloaded
2. `library()` will throw an **error** if the package is not downloaded

Question 7 (2 pts)

```
myAtomicVector_01 <- c(9, 8, 7, 6)
```

Write code which stores the mean of `myAtomicVector_01` in the variable `mySum`.

```
mySum <- mean(myAtomicVector_01)
```

Question 8 (1 pt)

`str` in the function `str()` is an abbreviation for what word?

Structure.

Question 9 (3 pts)

The data frame `mtcars` from base `R` looks like this

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Write the following sentence in RMarkdown syntax

"The average cylinder size is `xxx`"

where `xxx` is dynamically computed inline `R` code. **n.b.** Write out the entire sentence in `R`Markdown syntax as if it were going to be knitted.

The average cylinder size is `mean(mtcars$cyl, na.rm = T)`

Question 10 (2 pts)

You want to generate an `RMarkdown` code chunk where the code is evaluated, *but* the results are not printed to the page. What code chunk option(s) do you need to set to generate this result?

`include = F`

Question 11 (3 pts)

`R` and Python are both interpreted languages. What does **interpreted** mean? What other style of computer language is there that is not interpreted? Give one example.

Code is executed line by line instead of being compiled. C or Java are two programming lanaguages, not interpreted languages.

Question 12 (2 pts)

You are writing an `RMarkdown` document, and need to load three libraries: `Hmisc`, `dplyr` and `tibble`. Your audience is not interested in seeing any of the libraries being loaded. Write the **entire** `RMarkdown` code chunk that will load your libraries in a fashion that is transparent to the user.

```
# option for the code chunk: include = F
library(dplyr)
library(Hmisc)
library(tibble)
```

Question 13 (2 pts)

What line(s) of code should you include at the beginning of all you `R` scripts to ensure that your environment is clear of all data before running any code?

`rm(list=ls())`

Question 14 (3 pts)

Explain what functional masking is and provide an example (example may be fake).

Functions from various packages who share the same name.

1. Packaage 1, `myPak_01` loads a function called `abc()`
2. Packaage 2, `myPak_02` loads a function called `abc()`
3. When calling `abc()`, it will be called from the package `myPak_02` because it was loaded most recently.

Question 15 (2 pts)

Using code, how would you see which packages are currently loaded into memory?

```
(.packages())
search()
```

Question 16 (2 pts)

What is the purpose of global chunk options? Can these setting be suppressed? If so, how?

1. To set options which are the default vales for all code chunks in the document.
2. For each individual code chunk, the individual setting will override global settings.

Question 17 (2 pts)

You want to write code for demonstrative purposes, i.e., you want to show the end user your code, but you don't actually want to run it in `RMarkdown`. What code chunk option(s) do you need to set to generate this result?

`eval = F, echo = T`

Question 18 (4 pts)

Using the following code

```r
myAtomicVector <- c(99, 98, 97, 96)
```

write the line of `RMarkdown` code, including text, code and `RMarkdown` syntax, that generates the following text:

The **mean** of the vector `myAtomicVector` is 97.5, which is pretty *amazing*.

The `**mean**` of the vector `` `myAtomicVector` `` is `` `r mean(myAtomicVector)` ``, which is pretty `*amazing*`.