Microsoft: DAT209x Programming in R for Data Science

Bookmarks

- 0. Start Here
- 1. Introduction
- 2. Functions and Data Structures
- 3. Loops and Flow Control
- 4. Working with Vectors and Matrices
- 5. Reading in Data
- **6. Writing Data to Text Files**

Lecture

6. Writing Data to Text Files > Lab > Lab

Bookmark

Consider the following data frame for this lab:

```
set.seed(9007)
my.data<-data.frame(x=rnorm(10),y=rnorm(10)+5,z=rchisq(10,1))
```

Question 1

(1/1 point)
When using `write.table()`, R by defaults prints 15 digits after the decimal point. You want to write the data frame into a text file, but only with two digits after decimal point.

Which command should you use to perform the task?

○ write.table(my.data,digits = 2,"Assignment 6a.txt",row.names=FALSE)

● write.table(round(my.data,digits = 2),"Assignment 6a.txt",row.names=FALSE) ✓

○ write.table(my.data,precision = 2,"Assignment 6a.txt",row.names=FALSE)

Knowledge Checks
Quiz

Lab
Lab

7. Reading Data from SQL Databases

○ write.table(digits(my.data,2),"Assignment 6a.txt",row.names=FALSE)

EXPLANATION

You can use the following command to perform the task:

```
write.table(round(my.data,digits = 2),"Assignment 6a.txt",row.names=FALSE)
```

Question 2

(1/1 point)
Creating more digits is a little bit difficult. Study the `format()` function, and figure out a way to export my.data with 20 digits after the decimal point.

Which command should you use to perform the task?

○ write.table(my.data,precision = 20,"Assignment 6b.txt",row.names=FALSE)

○ write.table(digits(my.data,20),"Assignment 6b.txt",row.names=FALSE)

○ write.table(my.data,digits = 20,"Assignment 6b.txt",row.names=FALSE)

● write.table(format(my.data,digits = 20),"Assignment 6b.txt",row.names=FALSE) ✓

EXPLANATION

You can use the following command to perform the task:

```
write.table(format(my.data,digits = 20),"Assignment 6b.txt",row.names=FALSE)
```

Question 3

(2/2 points)

Now, let's try something different. Multiply `my.data` by 10e5 and store it to `my.data2`

Write `my.data2` to "Assignment 6c.txt" with no row names.

Read the data back into R in the object with the `read.table()` function, with header=TRUE specified, and divide the dataframe with 10e5.

Store the result in `my.data3`.

What is the difference between my.data[;1,1] and my.data3[1,1]

○ 0

○ 10e5

● very close to zero, something like 1.1102230246251565e-16 ✓

○ 1

How many digits after the decimal point that my.data[1,1] and my.data3[1,1] agree on? (use options(digits=20)).

○ 0

● 15 ✓

○ 20

○ 22

EXPLANATION

While the answer could vary on the system, visual inspection reveals that the two objects have the same digits for the first 15 positions on this system.

What is 1 - 0.9 - 0.1?

Typing the expression in the R console gives the value

```
> 1.0-0.9-0.1
[1] -2.77555756156289135l059e-17
```

A very small number but not 0. This is because of that the expression involved arithmetic operations, and this is as far as the floating point precision goes for R’s standard double precision. The same thing has happened in c): Multiplying with 10e5 doesn’t just move the decimal point, it is an actual calculation, and therefore a tiny difference can be seen when you look at 15-20 digits after the decimal point. There is a reason for R’s 15 digits; if you need to export a higher precision, you should explore R’s capabilities in that respect. As with so much else, R can handle the problem.