mda14jx_week1

August 21, 2018

```
In [4]: help(library)
```

1 Exercise 1

The code help(library) displays the *Loading/Attaching and Listing of Packages*. How to use the function.

```
In [23]: library
```

The library command code executes the command. The library( ) executes an empty command.

2 Exercise 2

```
In [9]: getwd()
```

```
In [36]: setwd("~/Autumn2016/Week1")
```

Marked in red as it's a string/ path, which has no value.

3 Exercise 3

```
In [14]: x<-3
         y<-10
         z<-15
```

```
In [16]: x+y+z
```

```
In [17]: (y-x)/z
```

```
In [21]: x*y*z
```

```
In [22]: (x+y+z)^2
```

```
In [25]: v<-c(x,y,z)
```

```
In [26]: sum(v)
```

The question states **calculate the sum of the vector raised to the power of 4**, which can be interperated in two ways. The first method calculates the (sum of the vector) raised to the power of 4, and the second method calculates the sum of (vector raised to the power of 4)

```
In [29]: (sum(v))^4
```

```
In [30]: sum(v^4)
```

```
In [33]: sqrt(z-x)
```

4 Exercise 4

```
In [3]: myname <-"Jingyi"
        email <-"jxie7@sheffield.ac.uk"
        module <-"BMS353"
        message <- paste(myname,email,module,sep=",")
        print (message)
```

```
[1] "Jingyi,jxie7@sheffield.ac.uk,BMS353"
```

5 Exercise5

```
In [67]: seq(1,30,by=2)
```

```
In [53]: seq(2,30,by=2)
```

```
In [68]: a<-seq(1,30,by=2)
```

```
In [64]: b<-seq(2,30,by=2)
```

```
In [69]: length(seq(1,30,by=2))
```

```
In [61]: length(seq(2,30,by=2))
```

```
In [70]: sum(a)
```

```
In [71]: sum(b)
```

```
In [72]: sum(seq(1,30))
```

The conclusion is that the sum of the entire sequence of numbers ranging from 1-30 is equal to the total sum of the odd numbers from 1-30 and even numbers from 2-30.

```
In [1]: sample(1:100,12,replace=TRUE)
```

6 Exercise 6

```
In [21]: dept<- "BMS"
         code<- 353
         BMSmodule<- c(dept,as.character(code))
         print(BMSmodule)
```

```
[1] "BMS" "353"
```

```
In [12]: dept<- "APS"
         code<- 227
         APSmodule<- c(dept,as.character(code))
         print(APSmodule)
```

```
[1] "APS" "227"
```

```
In [14]: dept<- "MBB"
         code<- 253
         MBBmodule<- c(dept,as.character(code))
         print(MBBmodule)
```

```
[1] "MBB" "253"
```

```
In [18]: mergevector<- c(BMSmodule,APSmodule,MBBmodule)
         print(mergevector)
```

```
[1] "BMS" "353" "APS" "227" "MBB" "253"
```

```
In [24]: stringsvector<- c("BMS","APS","MBB")
         print(stringsvector)
```

```
[1] "BMS" "APS" "MBB"
```

```
In [28]: x<-c("BMS","APS","MSS")
         y<-c(353,227,253)
         z<-c(x,as.character(y))
         print(z)
```

```
[1] "BMS" "APS" "MSS" "353" "227" "253"
```

```
In [30]: x<-c("BMS","APS","MSS")
         y<-c(353,227,253)
         z<- paste(x,y,sep="")
         print(z)
```

```
[1] "BMS353" "APS227" "MSS253"
```

7 Exercise 7

```
In [40]: Mat1<- matrix(1:20,nrow=4,ncol=5,byrow=TRUE)
         M
```

In R, matrices are created by column by defult, therefore byrow=TRUE will create the matrix by row.

```
In [33]: rownames(M)<-c("A","B","C","D")
         colnames(M)<-c("1st", "2nd", "3rd","4th","5th")
         M
```

```
In [53]: M=matrix(1:20,nrow=4,ncol=5,byrow=TRUE)
         M
```

```
In [88]: M[c(1,2),c(1,2)]
```

```
In [90]: M[c(1,2),c(1,2,4,5)]
```

```
In [87]: M[2,]
```

8 Exercise 8

```
In [92]: Mat2=matrix(c(32,42,18,20,33,38,25,28,26),nrow=3,ncol=3,byrow=TRUE)
         Mat2
```

```
In [94]: rownames(Mat2)<- c("BMS353","APS227","MBB253")
         colnames(Mat2)<- c("2013-14","2014-15","2015-16")
         Mat2
```

```
In [96]: Mat2["BMS353",]
```

9 Exercise 9

```
In [97]: Mat3=matrix(sample(1:100,12,replace=TRUE),nrow=3,ncol=4)
         Mat3
```

```
In [98]: Mat4=matrix(sample(1:100,12,replace=TRUE),nrow=3,ncol=4)
         Mat4
```

```
In [99]: Mat3+Mat4
```

```
In [100]: Mat4-Mat3
```

```
In [101]: Mat3-Mat4
```

```
In [108]: Mat3 %*% t(Mat4)
          t(Mat3)%*%Mat4
```

```
In [111]: sqrt(Mat3)
          sqrt(Mat4)
```

10 Exercise 10

```
In [20]: myFunction <- function(x) {
           ux <- x^3-1
           return(ux)
         }

         test<-myFunction(2)
         print(test)
```

```
[1] 7
```

2 is the value of the input, and 7 is the value of the output.

```
In [1]: x<- c(5,4,3,2,1)
        print(var(x))
```

```
[1] 2.5
```

The command var(x) computes the variance of x, whcih can be a numeric vector, matrix or data frame.

```
In [11]: y<- sum(x)
         z<- mean(x)
         n<- length(x)
         m<- (x-z)^2
         a<- sum(m)
         sd<- (1/(n-1))*a
         print(sd)
```

```
[1] 2.5
```

y= sum of x z= mean of x n= number of x m= square of x minus mean a= sum of the bars sd= variance

11 Exercise 11

```
In [21]: BMI <- function(h,w){ux<- h/(w^2)
                              return(ux)}
         test <- BMI(55,1.65)
         print(test)
```

```
[1] 20.20202
```

w= weight of body in kg h= height of body in m

12 Exercise 12

Week 1 practical content: * Basic operation in R * Use of markdown cells * Built in help functions in R * Changing path and verifying location of workspace * Working with vairables and objects, to perform simple calculations * Assigning value to object * The print command * Creating vectors and matrices, as well as their manipulation and calculations. * Transform numbers to characters. * Rearranging dimensions of matrices * Create user defined functions * Calculation of variance