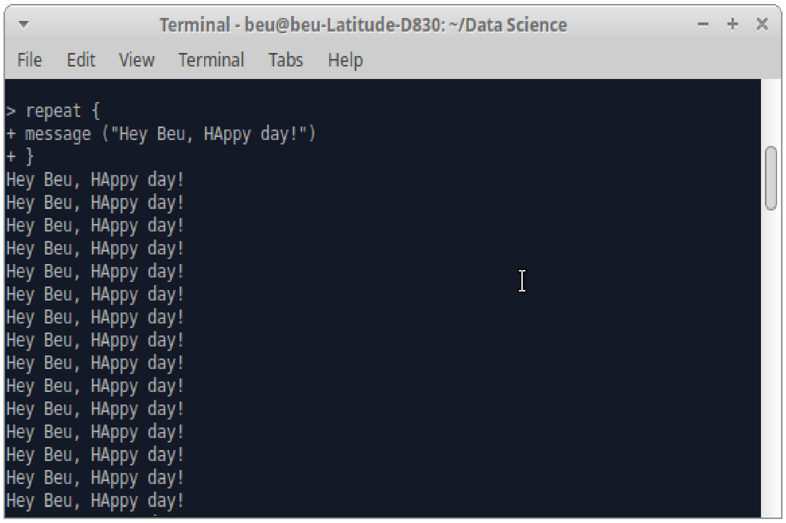
LABWORK – 5

Repeat loop in R

# ‘repeat’ executes the same code over and over until we tell it to stop

1.
```
> repeat {
+ message ("Hey Beu, HAppy day!")
+ }
```

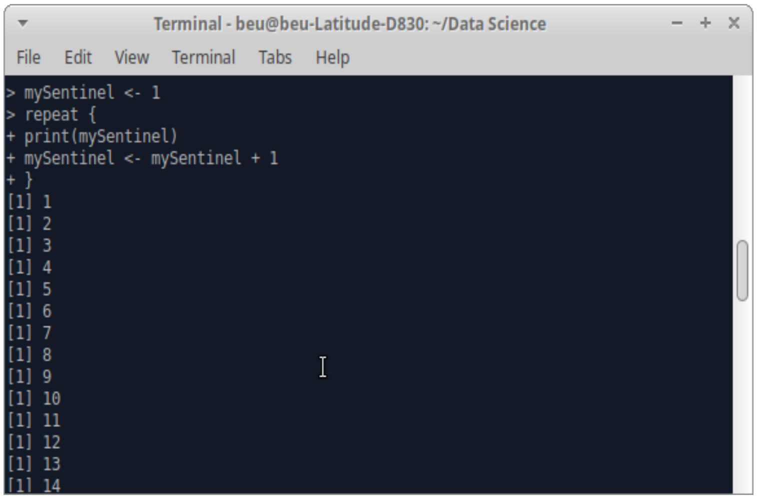
# infinite loop



```
> repeat {
+ message ("Hey Beu, HApppy day!")
+ }
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
Hey Beu, HApppy day!
```

2.
```
> mySentinel <- 1
> repeat {
+ print(mySentinel)
+ mySentinel <- mySentinel + 1
+ }
```

# infinite loop starts from 1



```
> mySentinel <- 1
> repeat {
+ print(mySentinel)
+ mySentinel <- mySentinel + 1
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
```

3.
```
> x <-  1
> repeat {
+ print(x)
+ x = x-1
+ if (x == -3) {
+ break }}
```
breaking out the infinite loop by including a break statement

```
[1] 1
[1] 0
[1] -1
[1] -2
```

4.

```
> myVector <- c ("hi", "hello")
> n <- 0
> repeat {
+ print(myVector)
+ n = n+1
+ if (n > 5) {
+ break
+ }
+ }
```

```
[1] "hi"    "hello"
[1] "hi"    "hello"
[1] "hi"    "hello"
[1] "hi"    "hello"
[1] "hi"    "hello"
[1] "hi"    "hello"
```

5.

```
> repeat {
+ message ("happy day!")
+ action <- sample (
+ c (
+ "Write R assignment",
+ "Do JAVA assignment",
+ "Go and SLEEP"
+ ),
+ 1
+ )
+ message("you should ",action)
+ if (action =="Go and SLEEP")
+ break
+ }
```

```
happy day!
you should Write R assignment
happy day!
you should Do JAVA assignment
happy day!
you should Go and SLEEP
```

6.

```
> repeat {
+ message ("happy day!")
+ action <- sample (
+ c ( "Write R assignment",
+ "Do JAVA assignment",
+ "Go and SLEEP"
+ ), 1 )
+ if (action =="Go and SLEEP")
+{ message ("skip! its time to
study")
+ next }
+ message("you should ",action)
+}
```

```
happy day!
you should Do JAVA assignment
happy day!
you should Write R assignment
happy day!
skip! its time to study
happy day!
you should Do JAVA assignment
happy day!
you should Do JAVA assignment
happy day!
skip! its time to study
happy day!
.
.
```

While loop in R

1.

```
> while (TRUE) {
+ print("Infinite loop!")
+ }
# its an infinite while loop
```

```
[1] "Infinite loop!"
[1] "Infinite loop!"
[1] "Infinite loop!"
[1] "Infinite loop!"
[1] "Infinite loop!"
[1] "Infinite loop!"
[1] "Infinite loop!"
[1] "Infinite loop!"
[1] "Infinite loop!"
        .
        .
```

2.

```
> mySentinel <- 1
> while (TRUE) {
+ print(mySentinel)
+ mySentinel <- mySentinel + 1
+ }
# infinite loop starts from 1
```

```
 [1] 1
 [1] 2
 [1] 3
 [1] 4
 [1] 5
 [1] 6
 [1] 7
 [1] 8
 [1] 9
[1] 10
[1] 11
[1] 12
   .
   .
```

3.

```
> x <- 1
> while (TRUE) {
+ print (x)
+ x = x-1
+ if (x == -5) {
+ break }}   # breaking out the
infinite while loop by including
a break statement
```

```
 [1] 1
 [1] 0
[1] -1
[1] -2
[1] -3
[1] -4
```

4.

```
> myVector <- c("While”,“Loop")
> n <- 0
> while (TRUE) {
+ print(myVector)
+ n = n+1
+ if (n > 5) {
+ break
+ }
+ }
```

```
[1] "While”,“Loop"
[1] "While”,“Loop"
[1] "While”,“Loop"
[1] "While”,“Loop"
[1] "While”,“Loop"
[1] "While”,“Loop"
```

5.

```
> myNum <- 0
> while (myNum < 10) {
+ print (myNum)
+ myNum = myNum + 1
+ }
```

```
[1] 0
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
```

6.

```
> action <- sample (
+ c (
+ "Write R assignment",
+ "Do JAVA assignment",
+ "Go and SLEEP"),1)
> while (action != "Go and SLEEP") {
+ message ("Hard Day!")
+ action <- sample (
+ c (
+ "Write R assignment",
+ "Do JAVA assignment",
+ "Go and SLEEP"),1)
+  message ("",action) }
```

```
Hard Day!
Do JAVA assignment
Hard Day!
Write R assignment
Hard Day!
Do JAVA assignment
Hard Day!
Go and SLEEP
```

For loop in R

# for loop is to be used when we know exactly how many times we want the code to repeat

1.

```
> for (i in 1:5)
+ message("hi")
```

```
hi
hi
hi
hi
hi
```

# iterates for 5 times

2.

```
> for (i in 1:5)
+ print(i)
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

3.

```
> for (i in 1:5) {
+ mySquare <- i^2
+ print(mySquare)
+ }
```

```
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
```

4.

```
> for (month in month.name) {
+ message(month)
+ }
```

```
January
February
March
April
May
June
July
August
September
October
November
December
```

5.
```
> for (myVector in c ("Data
Science", "Java",
"Cryptography")) {
+ message ("I like ", myVector)
+ }
```

```
I like Data Science
I like Java
I like Cryptography
```

6.
```
> myList <- list (
+ "hi, my name is beu",
+ pi,
+ LETTERS[1:10]
+ )
> myList
[[1]]
[1] "hi, my name is beu"

[[2]]
[1] 3.141593

[[3]]
 [1] "A" "B" "C" "D" "E" "F"
"G" "H" "I" "J"

> for (i in myList) {
+ print (i)
+ }
```

```
[1] "hi, my name is beu"
[1] 3.141593
[1] "A" "B" "C" "D" "E" "F" "G" "H"
"I" "J"
```

Loop Control Statements

# Loop-control statements are part of control statements in R programming that are used to change the execution of a loop from its normal execution sequence.

1.
```
> myVector <- c (1:10)
> myVector
 [1]  1  2  3  4  5  6  7  8  9 10
> for (i in myVector) {
+ if (i == 7) {
+ break  }
+ print(i)
+ }  # break statement
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
```

2.

```
> x <-  1
> repeat {
+ print(x)
+ x = x-1
+ if (x == -3) {
+ break }}     # breaking out the
infinite loop by including a
break statement
```

```
[1] 1
[1] 0
[1] -1
[1] -2
```

3.

```
> x <- 1
> while (TRUE) {
+ print (x)
+ x = x-1
+ if (x == -5) {
+ break }}    # breaking out the
infinite while loop by including
a break statement
```

```
[1] 1
[1] 0
[1] -1
[1] -2
[1] -3
[1] -4
```

4.

```
> myVector <- c (1:10)
> myVector
 [1]  1  2  3  4  5  6  7  8  9 10
> for (i in myVector) {
+ if (i == 7) {
+ next
+ }
+ print(i)
+ }
# next statement
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 8
[1] 9
[1] 10
```

5.

```
x <-  1
repeat {
print(x)
x = x-1
if (x == -3) {
next }}     # Skipping out the
value '-3'  from the infinite loop
by including a next statement
```

```
[1] 1
[1] 0
[1] -1
[1] -2
[1] -4
[1] -5
[1] -6
  .
  .
```

6.

```
> x <- 1
> while (TRUE) {
+ print (x)
+ x = x+1
+ if (x == 2) {
+ next }}   # Skipping out the
value ‘2’  from the infinite loop
by including a next statement
```

```
[1] 1
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
   .
   .
```