Introduction to R.

1. R syntax

```
> "Hello World!"
[1] "Hello World!"
> 5+5
[1] 10
> 5
[1] 5
> 10
[1] 10
```

2. R print

```
> "Hello World!"
[1] "Hello World!"
> print("Hello World!")
[1] "Hello World!"
> for (x in 1:10) {
+     print(x)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

3. R comment

```
> # This is a comment
> "Hello World!"
[1] "Hello World!"
>
> "Hello World!"  # This is a comment
[1] "Hello World!"
>
```

4. R variables

```
> name <- "John"
> age <- 40
>
> name
[1] "John"
>
> age
[1] 40
```

```
> name <- "Sadik"
>
> for (x in 1:10) {
+     print(name)
+ }
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
[1] "Sadik"
```

5. Data Types

```
> x <- 10.5
> class(x)
[1] "numeric"
>
> x <- 1000L
> class(x)
[1] "integer"
>
>
> x <- 9i + 3
> class(x)
[1] "complex"
>
> x <- "R is exciting"
> class(x)
[1] "character"
>
> x <- TRUE
> class(x)
[1] "logical"
```

6. R numbers

```
> x <- 10.5    # numeric
> y <- 10L     # integer
> z <- 1i
>
> class(x)
[1] "numeric"
> class(y)
[1] "integer"
> class(z)
[1] "complex"
```

7. R string

```
> # Print string
> "Hello World"
[1] "Hello World"
>
> #Assign a string to a variable
> str <- "Hello"
>
> str # print the value of str
[1] "Hello"
>
> # cat function is used to print a newline
> str <- "Lorem ipsum dolor sit amet,
+ consectetur adipiscing elit,
+ sed do eiusmod tempor incididunt
+ ut labore et dolore magna aliqua."
>
> cat(str) # print the value of str
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

8. R Booleans

```
> 10 > 9    # TRUE because 10 is greater than 9
[1] TRUE
> 10 == 9   # FALSE because 10 is not equal to 9
[1] FALSE
> 10 < 9    # FALSE because 10 is greater than 9
[1] FALSE
```

```
> a <- 10
> b <- 9
> a > b
[1] TRUE
> a <- 200
> b <- 33
> if (b > a) {
   print ("b is greater than a")
 } else {
  Print ("b is not greater than a")
}
[1] "b is not greater than a"
```

9. R operators

```
> #Addition
> 5+5
[1] 10
> #Substraction
> 5-5
[1] 0
> #Multiplication
> 5*5
[1] 25
> #Division
> 5/5
[1] 1
> #Greater than
> 5 > 10
[1] FALSE
> #Less than
> 5 < 10
[1] TRUE
```

```
> #Equal
> 5==5
[1] TRUE
> #Modulus
> 5%%4
[1] 1
```

10. R if…else

```
> a <- 200
> b <- 33
> if (b > a) {
+   print("b is greater than a")
+ } else if (a == b) {
+   print("a and b are equal")
+ } else {
+   print("a is greater than b")
+ }
[1] "a is greater than b"
```

11. R while loop with break statement

```
> i <- 1
> while (i < 6) {
+   print(i)
+   i <- i + 1
+   if (i == 4) {
+     break
+   }
+ }
[1] 1
[1] 2
[1] 3
```

12. R for loop

```
> for(x in  1:5){
+   print(x)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

13. R function

```
> my_function <- function() {
+   print("Hello World!")
+ }

> my_function()
[1] "Hello World!”
```

14. R global and sciope variable

```
> #This txt is a global variable
> txt <- "global variable"
> my_function <- function() {
+   #This txt is a local variable
+   txt = "fantastic"
+   paste("R is", txt)
+ }
> my_function()
[1] "R is fantastic"
> txt # print txt
[1] "global variable"
```

15. R Vectors:

```
> fruits <- c("banana", "apple", "orange")
> fruits
[1] "banana" "apple"  "orange"
```

16. R Vector of numeric sequence:

```
> numbers1 <- 1.5:6.5
> numbers1
[1] 1.5 2.5 3.5 4.5 5.5 6.5
>
> numbers2 <- 1.5:6.3
> numbers2
[1] 1.5 2.5 3.5 4.5 5.5
```

17. Vector Length

```
> fruits <- c("banana", "apple", "orange")
>
> length(fruits)
[1] 3
```

18. Vector sort

```
> fruits <- c("banana", "apple", "orange", "mango", "lemon")
> numbers <- c(13, 3, 5, 7, 20, 2)
>
> sort(fruits)  # Sort a string
[1] "apple"  "banana" "lemon"  "mango"  "orange"
> sort(numbers) # Sort numbers
[1]  2  3  5  7 13 20
```

19. Vectors other property

```
> fruits <- c("banana", "apple", "orange", "mango", "lemon")
>
> fruits[1] <- "pear"
>
> fruits
[1] "pear"   "apple"  "orange" "mango"  "lemon"
>
> repeat_each <- rep(c(1,2,3), each = 3)
>
> repeat_each
[1] 1 1 1 2 2 2 3 3 3
```

20. R Lists

```
> thislist <- list("apple", "banana", "cherry")
> thislist
[[1]]
[1] "apple"
[[2]]
[1] "banana"
[[3]]
[1] "cherry"
>
> thislist <- list("apple", "banana", "cherry")
> thislist[1]
[[1]]
[1] "apple"
>
> thislist <- list("apple", "banana", "cherry")
> thislist[1] <- "blackcurrant"
>
> thislist
[[1]]
[1] "blackcurrant"
[[2]]
[1] "banana"
[[3]]
[1] "cherry"
>
> thislist <- list("apple", "banana", "cherry")
> length(thislist)
[1] 3
```

21. R matrix

```
> thismatrix <- matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)
>
> thismatrix

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
>
> thismatrix <- matrix(c("apple", "banana", "cherry", "orange"),
nrow = 2, ncol = 2)
>
> thismatrix
     [,1]     [,2]

[1,] "apple"  "cherry"

[2,] "banana" "orange"
>
> thismatrix <- matrix(c("apple", "banana", "cherry", "orange"),
nrow = 2, ncol = 2)
>
> thismatrix[1, 2]

[1] "cherry"
>
> thismatrix <- matrix(c("apple", "banana", "cherry", "orange"),
nrow = 2, ncol = 2)
>
> thismatrix[2,]

[1] "banana" "orange"
>
> thismatrix <- matrix(c("apple", "banana", "cherry", "orange"),
nrow = 2, ncol = 2)
>
> thismatrix[,2]

[1] "cherry" "orange"
```