**Vectors in R**

A vector is a collection of elements, all of the same type. For instance, **c(1, 3, 2, 1, 5)** is a vector consisting of the numbers 1, 3, 2, 1, 5, in that order.

Similarly**, c("R", "Excel", "SAS", "Excel")** is a vector of the character elements “R,” “Excel,” “SAS” and “Excel.” A vector cannot be of mixed type.

Vectors play a crucial, and helpful, role in R. More than being simple containers, vectors in R are special in that R is a vectorized language. That means operations are applied to each element of the vector automatically, without the need to loop through the vector. This is a powerful concept that may seem foreign to people coming from other languages, but it is one of the greatest things about R.

**Implementing Vector**

X <-c(1,2,3,4,5,6,7,8,9,10)

X

X\*3

X+2

X-2

X/4

X^2

Sqrt(x)

**Accessing the values of Vectors**

* 1:10

Output will be: **(1,2,3,4,5,6,7,8,9,10)**

* 10:1

Output will be: **(10 9 8 7 6 5 4 3 2 1)**

* **-2:3**

Output will be: **(-2,-1,0,1,2,3)**

* **-5: 2**

Output will be: **(-5,-4,-3,-2,-1,0,1,2)**

**Adding two Vectors**

X <-c(1,2,3,4,5)

Y <-c(6,7,8,9,10)

Output will be: **(7,9,11,13,15)**

**Vectors with Strings**

q <- **c**("Hockey", "Football", "Baseball", "Curling",

"Rugby")

* **nchar(q)**

**Missing Data**

Missing data plays a critical role in both statistics and computing, and R has two types of missing data, NA and NULL. While they are similar, they behave differently and that difference needs attention.

**N.A**

Often we will have data that has missing values for any number of reasons. Statistical programs use varying techniques to represent missing data such as a dash, a period or even the number 99. R uses NA. NA will often be seen as just another element of a vector. is.na tests each element of a vector for missingness.

* **Z <-c(1,2,NA,8,3,NA,3)**
* **Z**

**To check NULL values**

To check the Null values in the dataset **is.na(Z)** is used for checking it will return TRUE if data contains NULL values.

Handling missing data is an important part of statistical analysis. There are many techniques depending on field and preference. One popular

technique is **multiple imputation**.

**NULL**

NULL is the absence of anything. It is not exactly missingness, it is nothingness. Functions can sometimes return NULL and their arguments can be NULL. An important difference between NA and NULL is that NULL is atomical and cannot exist within a vector. If used inside a vector it simply disappears.

* Z <-c(1,NULL,3,4,5,NULL)

Z

The test for a NULL value is is.null.

D <- NULL

Is.null(D)

The output will be true.

**Conclusion**

Data come in many types, and R is well equipped to handle them. In addition to basic calculations, R can handle numeric, character and time-based data. One of the nicer parts of working with R, although one that requires a different way of thinking about programming, is vectorization. This allows operating on multiple elements in a vector simultaneously, which leads to faster and more

mathematical code.

**Vector Creation**

### **Single Element Vector**

Even when you write just one value in R, it becomes a vector of length 1 and belongs to one of the above vector types.

print(“abc”);

print(12.4)

print(64L)

print(True)

print(2+3i)

print(charToRaw(‘hello’))

### **Multiple Elements Vector**

#creating a sequence from 5 to 12

V <- 5:12

Print(v)

V <- 5.3: 12.3

Print(V)

V<- 3.8: 11.3

If final does not belong to the sequence it will be discarded.

Using Seq Operator

Print(seq(5,9,by=0.4))

**Using the c() function**

The non-character values are coerced to character type if one of its elements is a character.

S <- c(‘apple’, ‘red’,5,’grapes’)

Print(s)

## Accessing Vector Elements

Elements of a Vector are accessed using indexing. The **[ ] brackets** are used for indexing. Indexing starts with position 1. Giving a negative value in the index drops that element from result.**TRUE**,**FALSE** or **0** and **1** can also be used for indexing

## Vector Manipulation

### **Vector arithmetic**

Two vectors of same length can be added, subtracted, multiplied or divided giving the result as a vector output.

V1 <- c(3,8,4,5,0,11)

V2 <- c(4,11,0,8,1,2)

V1 + V2

### **Vector Element Recycling**

If we apply arithmetic operations to two vectors of unequal length, then the elements of the shorter vector are recycled to complete the operations

V1 <- c(3,4,5,6)

V2 <-c(2,3)

V1 + V2

### **Vector Element Sorting**

Elements in a vector can be sorted using the **sort()** function

V <-c(3,4,5,65,1,-9,304)

print(Sort(v))

v <-c(2,3,567,322)

print(sort(v,decreasing = TRUE))

colors <-c(“Red”,”Blue”,”Yellow”,Violet”)

print(sort(colors))

## How to modify a vector in R?

X <-c(1,2,3,4,6,7,90)

X[2] <- 100

X

X[4] <- 200

X

## How to delete a Vector?

X <-c (1,2,3,4,5,67)

X <- NULL