Special Values , Classes and Coercion

Special values have specific meanings and behaviors in R, and can be used in conditional statements and calculations.These include Infinity (Inf and -Inf), Not a Number (NaN),Not Available (NA) and NULL.

Infinity

1. Infinity in R is represented by Inf (positive infinity) and -Inf (negative infinity).
2. You can create infinity by dividing a positive number by 0 or using the Inf constant.
3. You can check if a value is infinite using the is.infinite() function.
4. You can check if a value is finite(not infinite or NaN) using the is.finite() function.

```
1/0
```

```
## [1] Inf
```

```
is.infinite(Inf)    # check the term is infinity.
```

```
## [1] TRUE
```

```
is.infinite(1)      # check the term is infinity.
```

```
## [1] FALSE
```

```
is.finite(1)        # check the term is finite.
```

```
## [1] TRUE
```

```
is.finite(Inf)      # check the term is finite.
```

```
## [1] FALSE
```

NaN

NaN stands for "Not a Number" and represents an invalid numeric result, such as 0/0 or the square root of a negative number.You can check if a value is NaN using the is.nan() function.

```
0/0           # to create a not a number.
```

```
## [1] NaN
```

```
is.nan(NaN)   # to check if is nan.
```

```
## [1] TRUE
```

```
is.nan(1)     # to check if is nan.
```

```
## [1] FALSE
```

NA

1. NA represents a missing value in a data set.
2. You can check for NA (or NaN) using the is.na() function.
3. The na.omit() function removes any NA and NaN values from a vector or data frame.

```
vec_1 <- c(1,2,NA,4)
vec_1
```

```
## [1]  1  2 NA  4
```

```
is.na(vec_1)
```

```
## [1] FALSE FALSE  TRUE FALSE
```

```
na.omit(vec_1)
```

```
## [1] 1 2 4
## attr(,"na.action")
## [1] 3
## attr(,"class")
## [1] "omit"
```

NULL

1. NULL represents the absence of a value.
2. It is often used as a placeholder or to indicate an empty result.
3. You can check for NULL using the is.null() function.

```
foo <- NULL
is.null(foo)
```

```
## [1] TRUE
```

```
bar <- c(1, NULL, 3)
is.null(bar)
```

```
## [1] FALSE
```

Classes and Coercion

1. **Classes** are used to categorize objects based on their data type and structure, such as numeric, logical, character, factor, matrix, or data frame.
2. **Coercion** converts an object from one class to another using functions like as.numeric(), as.character(), as.factor(), etc.

```
y<-c(1,2,3,4,5)
as.character(y)
```

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

```
as.factor(y)
```

```
## [1] 1 2 3 4 5
## Levels: 1 2 3 4 5
```

Explicit Attributes

Explicit attributes provide additional information about an object,such as dimensions, names, and class.You can access attributes using the attributes() or attr() functions.The dimnames() function specifically retrieves dimension names for arrays.

```
x <- matrix(1:4, nrow=2)
dimnames(x) <- list(c("row1","row2"), c("col1","col2"))
attributes(x)
```

```
## $dim
## [1] 2 2
##
## $dimnames
## $dimnames[[1]]
## [1] "row1" "row2"
##
## $dimnames[[2]]
## [1] "col1" "col2"
```

Object Class

You can check the class of an object using the class() function.

```
vec <- 1:3
mat <- matrix(1:6, nrow=2)

class(vec)
```

```
## [1] "integer"
```

```
class(mat)
```

```
## [1] "matrix" "array"
```

Object-Checking Functions

1. R provides specific functions to check if an object belongs to a particular class.
2. These functions start with is. followed by the class name,such as is. numeric() ,is.character(),is.matrix(), etc.

```
is.numeric(1)
```

```
## [1] TRUE
```

```
is.character("hello")
```

```
## [1] TRUE
```

```
is.matrix(matrix(1:4, nrow=2))
```

```
## [1] TRUE
```

Object-Coercion Functions

You can convert objects between classes using functions that start with as. followed by the desired class name.

```
as.character(1:3)
```

```
## [1] "1" "2" "3"
```

```
as.data.frame(matrix(1:4, nrow=2))
```

```
##   V1 V2
## 1  1  3
## 2  2  4
```