

Foundations of R Software

Lecture 7

Introduction

⋮⋮⋮

Some more Basic Operations in R

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Assignment operator

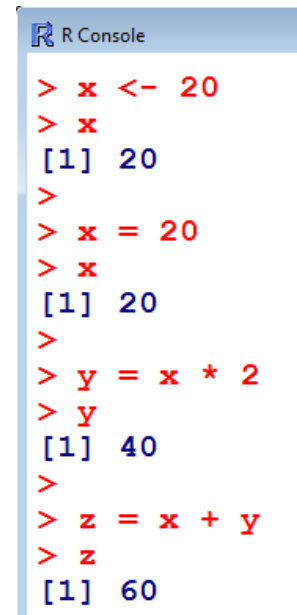
- **>** is the prompt sign in R.
- The assignment operators are the left arrow with dash **<-** and equal sign **=**

> x <- 20 assigns the value **20** to **x**.

> x = 20 assigns the value **20** to **x**.

Initially only **<-** was available in R.

- **> x = 20** assigns the value **20** to **x**.
- **> y = x * 2** assigns the value **2*x** to **y**.
- **> z = x + y** assigns the value **x + y** to **z**.



```
R Console
> x <- 20
> x
[1] 20
>
> x = 20
> x
[1] 20
>
> y = x * 2
> y
[1] 40
>
> z = x + y
> z
[1] 60
```

Assignment of numbers and characters

- The numbers are assigned in the usual way

> **x <- 20** assigns the value 20 to **x**.

> **x = 20** assigns the value 20 to **x**.

- The characters assigned within double or single quotes

> **x = "apple"** assigns the apple to **x**.

> **x <- "apple1"** assigns the apple to **x**.

> **x = 'apple'** assigns the apple to **x**.

> **x <- 'apple'** assigns the apple to **x**.

Assignment of numbers and characters

```
R Console
> x = 20
> x
[1] 20
> x <- 20
> x
[1] 20
> x=apple
Error: object 'apple' not found
> x = "apple"
> x
[1] "apple"
> x = 'apple'
> x
[1] "apple"
> |
```

Knowing numbers and characters

To know if a value is a number, use `is.numeric()`

To know if a value is a character, use `is.character()`

```
x = 20
```

```
> is.numeric(x)
```

```
[1] TRUE
```

```
> is.character(x)
```

```
[1] FALSE
```

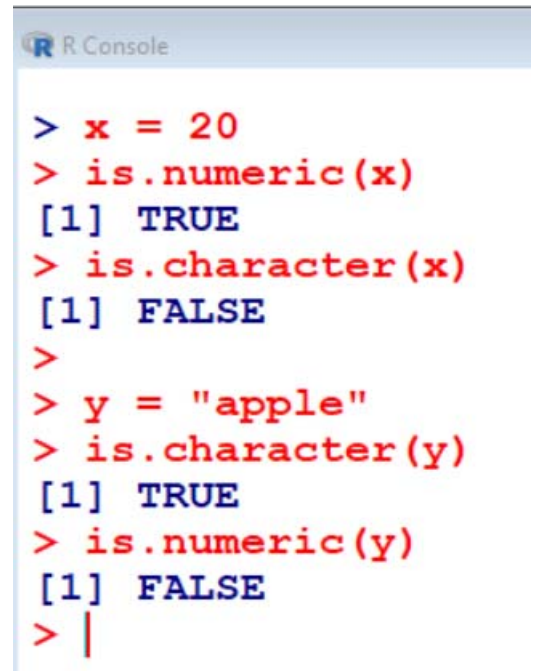
```
y = "apple"
```

```
> is.character(y)
```

```
[1] TRUE
```

```
> is.numeric(y)
```

```
[1] FALSE
```



```
R Console  
  
> x = 20  
> is.numeric(x)  
[1] TRUE  
> is.character(x)  
[1] FALSE  
>  
> y = "apple"  
> is.character(y)  
[1] TRUE  
> is.numeric(y)  
[1] FALSE  
> |
```

Converting numbers and characters

To convert a value as a number, use `as.numeric()`

To convert a value as a character, use `as.character()`

Converting a number into a character:

```
> x = 20
```

```
> is.numeric(x)
```

```
[1] TRUE
```

```
> y = as.character(x)
```

```
> is.numeric(y)
```

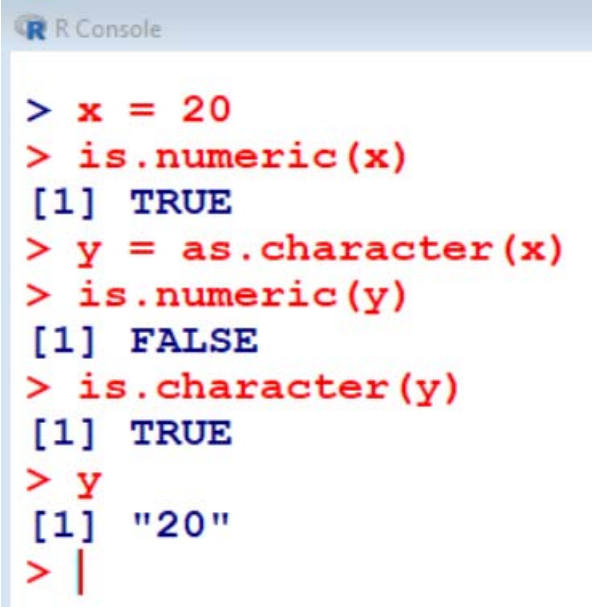
```
[1] FALSE
```

```
> is.character(y)
```

```
[1] TRUE
```

```
> y
```

```
[1] "20"
```



```
R Console
> x = 20
> is.numeric(x)
[1] TRUE
> y = as.character(x)
> is.numeric(y)
[1] FALSE
> is.character(y)
[1] TRUE
> y
[1] "20"
> |
```

Converting numbers and characters

Converting a character into a number:

```
> y = "apple"
> is.numeric(y)
[1] FALSE

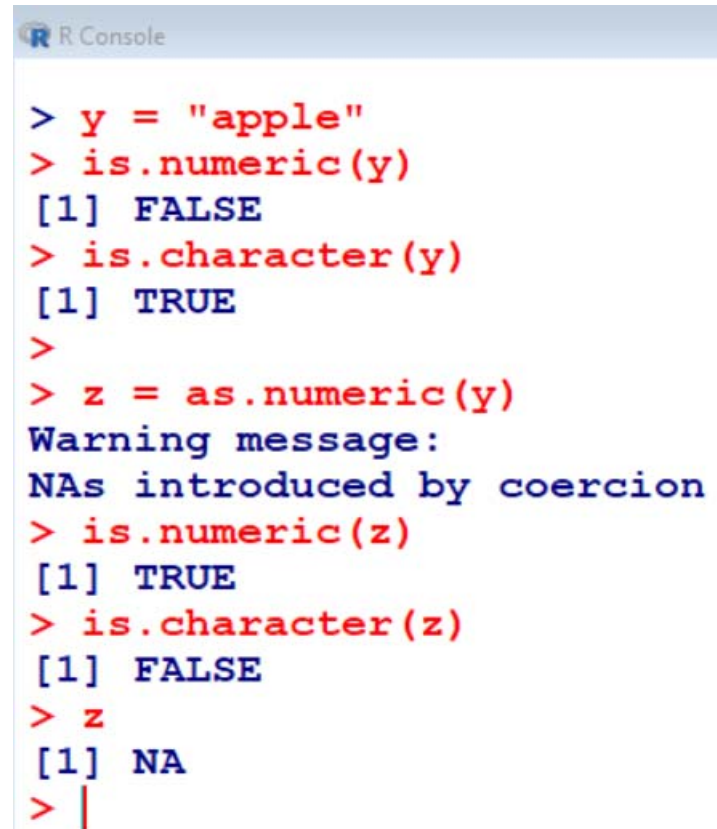
> is.character(y)
[1] TRUE

> z = as.numeric(y)
Warning message:
NAs introduced by coercion

> is.numeric(z)
[1] TRUE

> is.character(z)
[1] FALSE

> z
[1] NA
```

A screenshot of an R console window showing the same R code and output as the previous block. The window has a title bar that says "R Console". The code is entered in red text, and the output is in blue text. The warning message "NAs introduced by coercion" is displayed in blue text. The prompt character ">" is shown at the end of the last line of code.

```
R Console

> y = "apple"
> is.numeric(y)
[1] FALSE
> is.character(y)
[1] TRUE
>
> z = as.numeric(y)
Warning message:
NAs introduced by coercion
> is.numeric(z)
[1] TRUE
> is.character(z)
[1] FALSE
> z
[1] NA
> |
```

Comment operator

: The character # marks the beginning of a comment.

All characters until the end of the line are ignored.

> # mu is the mean

> # x <- 20 is treated as comment only

```
R Console
> x <- 20
> x
[1] 20
>
```

Case sensitivity in R

- Capital and small letters are different.

> **X** <- 20 and > **x** <- 20 are different

```
R R Console
> x <- 20
> x
[1] 20
>
> X
Error: object 'X' not found
>
> X <- 30
> X
[1] 30
> x
[1] 20
,
```

Combining values in a data vector

- The command `c(1,2,3,4,5)` combines the numbers 1,2,3,4 and 5 to a vector.

```
R Console
> y=1,2,3,4,5
Error: unexpected ',' in "y=1,"
>
> y=(1,2,3,4,5)
Error: unexpected ',' in "y=(1,"
>
> y=c(1,2,3,4,5)
>
> y
[1] 1 2 3 4 5
```

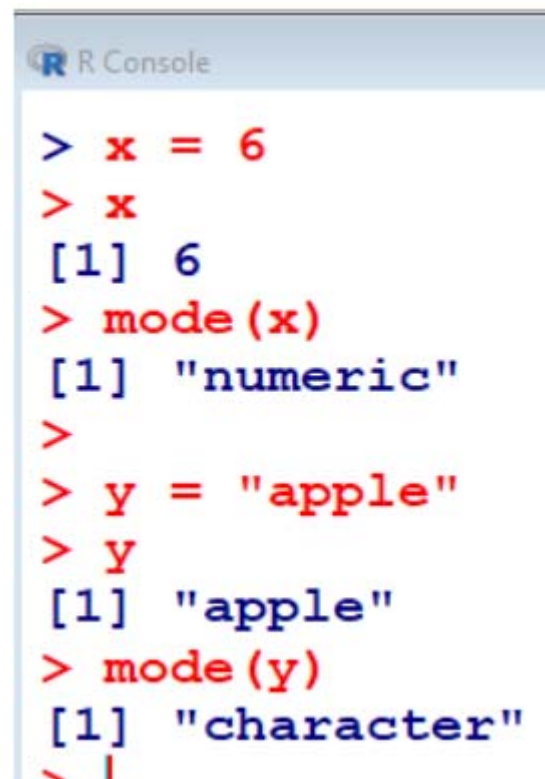
Mode

Command `mode()` explains the type or storage mode of an object.

[**Be watchful**: It's not like mean, median, mode]

```
> x = 6
> x
[1] 6
> mode(x)
[1] "numeric"
```

```
> y = "apple"
> y
[1] "apple"
> mode(y)
[1] "character"
```



```
R Console
> x = 6
> x
[1] 6
> mode(x)
[1] "numeric"
>
> y = "apple"
> y
[1] "apple"
> mode(y)
[1] "character"
~ |
```

Mode

Following modes are available:

- `"logical",`
- `"integer",`
- `"double",`
- `"complex",`
- `"raw",`
- `"character",`
- `"list",`
- `"expression",`
- `"name",`
- `"symbol" and`
- `"function".`

Mode

Command `storage.mode()` returns the storage mode of its argument.

It is generally used when calling functions written in another language, such as C or FORTRAN, to ensure that R objects have the data type expected by the routine being called.

Mode

mode	storage.mode
logical	logical
numeric	integer
numeric	double
complex	complex
character	character
raw	raw

Mode

Examples:

```
> x = 6 #numeric
```

```
> storage.mode(x)
```

```
[1] "double"
```

```
> x = TRUE #logical
```

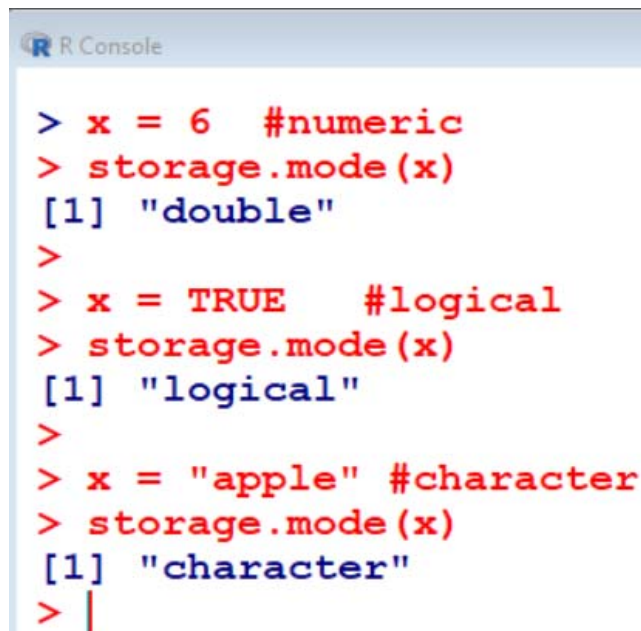
```
> storage.mode(x)
```

```
[1] "logical"
```

```
> x = "apple" #character
```

```
> storage.mode(x)
```

```
[1] "character"
```

A screenshot of an R console window with a light blue header bar containing the R logo and the text "R Console". The console shows three examples of setting a variable and checking its storage mode. The first example sets x to 6, the second to TRUE, and the third to "apple". Each time, the storage.mode(x) function is called, returning "double", "logical", and "character" respectively. The prompt character is a blue ">".

```
> x = 6 #numeric
> storage.mode(x)
[1] "double"
>
> x = TRUE #logical
> storage.mode(x)
[1] "logical"
>
> x = "apple" #character
> storage.mode(x)
[1] "character"
> |
```

Infinity

Some calculations yield result as infinity (∞). For example, $3/0$. In R

`is.finite()` and `is.infinite()` are used to know if an

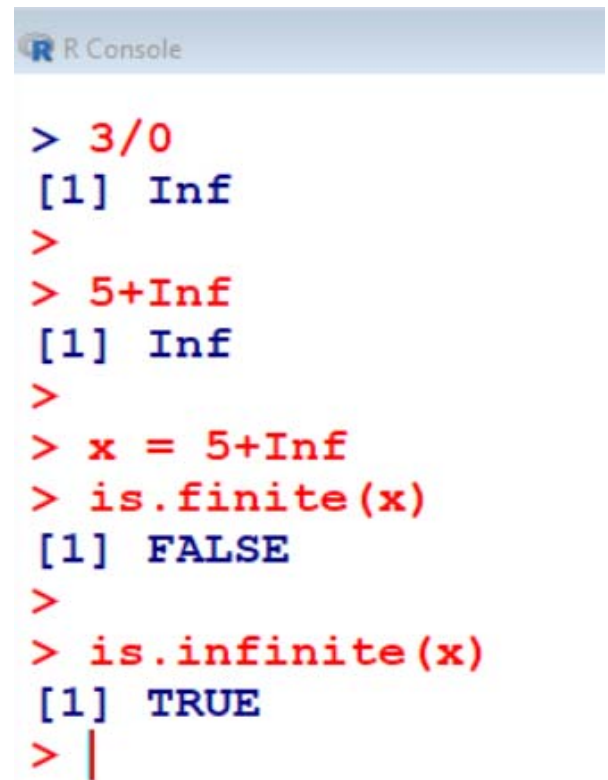
outcome is finite or infinite, respectively.

```
> 3/0  
[1] Inf
```

```
> 5+Inf  
[1] Inf
```

```
> x = 5+Inf  
> is.finite(x)  
[1] FALSE
```

```
> is.infinite(x)  
[1] TRUE
```

A screenshot of an R console window with a light blue header bar containing the R logo and the text "R Console". The console shows a series of commands and their outputs in a monospaced font. The commands are: 3/0, 5+Inf, x = 5+Inf, is.finite(x), and is.infinite(x). The outputs are: Inf, Inf, FALSE, and TRUE respectively. The prompt character ">" is used for each line.

```
R Console  
> 3/0  
[1] Inf  
>  
> 5+Inf  
[1] Inf  
>  
> x = 5+Inf  
> is.finite(x)  
[1] FALSE  
>  
> is.infinite(x)  
[1] TRUE  
> |
```