Introduction to R Software

Strings – Display and Formatting
:::
Replacement and Manipulations with Alphabets

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Operations with Strings

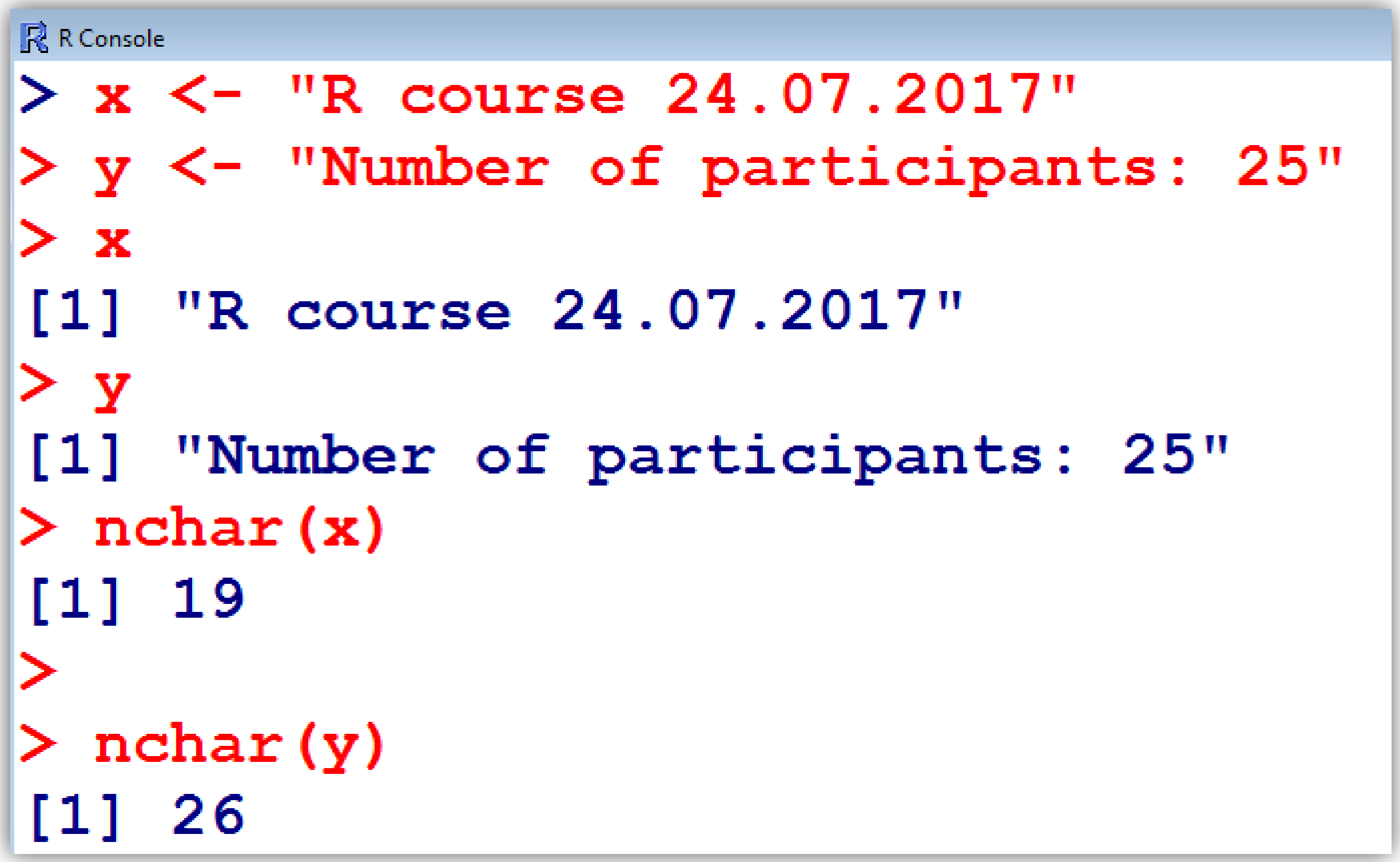
There are a variety of commands that can be used for strings.

Examples:

Count of number of characters:

```
> x <- "R course 24.07.2017"

> y <- "Number of participants: 25"


> nchar(x) #Count the Number of Characters in x
[1] 19

> nchar(y) #Count the Number of Characters in y
[1] 26
```

Operations with Strings



```
> x <- "R course 24.07.2017"
> y <- "Number of participants: 25"
> x
[1] "R course 24.07.2017"
> y
[1] "Number of participants: 25"
> nchar(x)
[1] 19
>
> nchar(y)
[1] 26
```

Operations with Strings

`sub` and `gsub` Functions:

Within a string, we want to replace one substring with another.

Use `sub` and `gsub` to replace the first instance of a substring:

```
sub(old, new, string)
```

The `sub` function finds the first instance of the old substring within string and replaces it with the new substring.

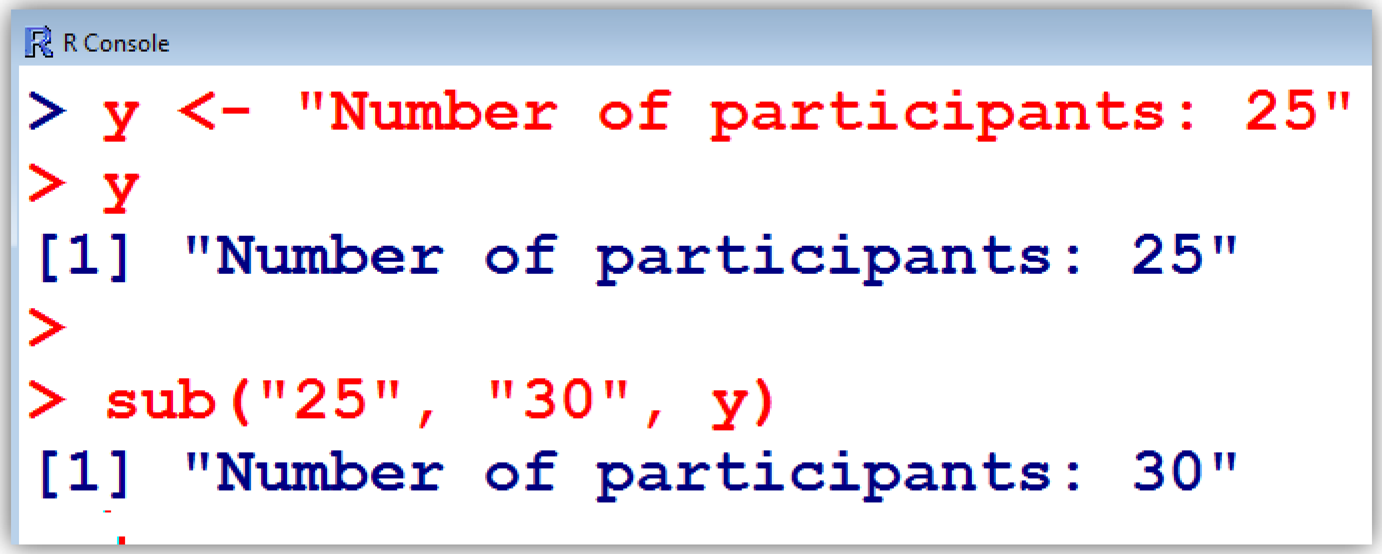
`gsub` does the same thing, but it replaces all instances of the substring (a global replace), not just the first.

```
gsub(old, new, string)
```

Operations with Strings

Examples:

```
> y <- "Number of participants: 25"

> sub("25", "30", y)
[1] "Number of participants: 30"
```



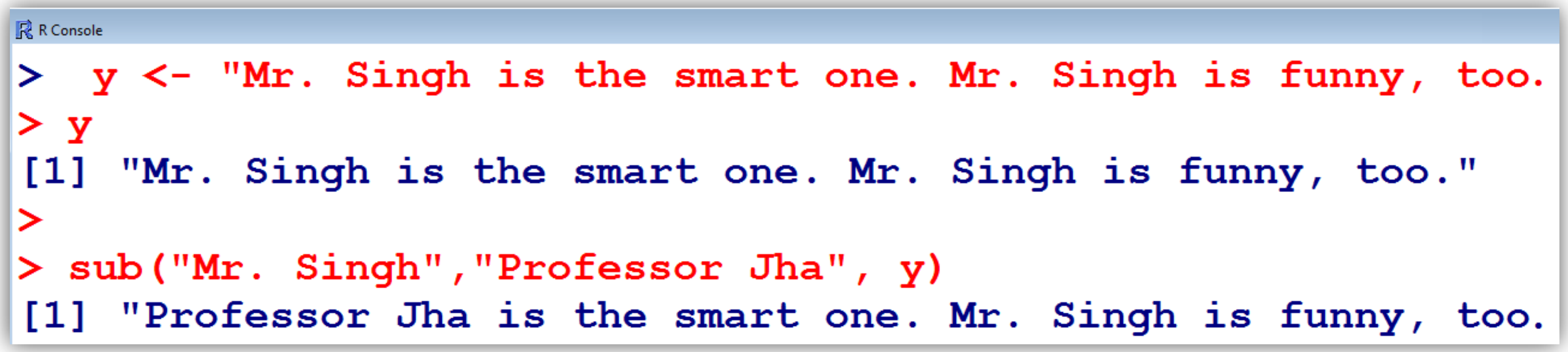
```
> y <- "Number of participants: 25"
> y
[1] "Number of participants: 25"
>
> sub("25", "30", y)
[1] "Number of participants: 30"
```

Operations with Strings

Examples:

```
>  y <- "Mr. Singh is the smart one. Mr. Singh
is funny, too."

> y
[1] "Mr. Singh is the smart one. Mr. Singh is
funny, too."

> sub("Mr. Singh","Professor Jha", y)
[1] "Professor Jha is the smart one. Mr. Singh
is funny, too."
```

Operations with Strings



```
>  y <- "Mr. Singh is the smart one. Mr. Singh is funny, too.
> y
[1] "Mr. Singh is the smart one. Mr. Singh is funny, too."
>
> sub("Mr. Singh","Professor Jha", y)
[1] "Professor Jha is the smart one. Mr. Singh is funny, too.
```

Operations with Strings

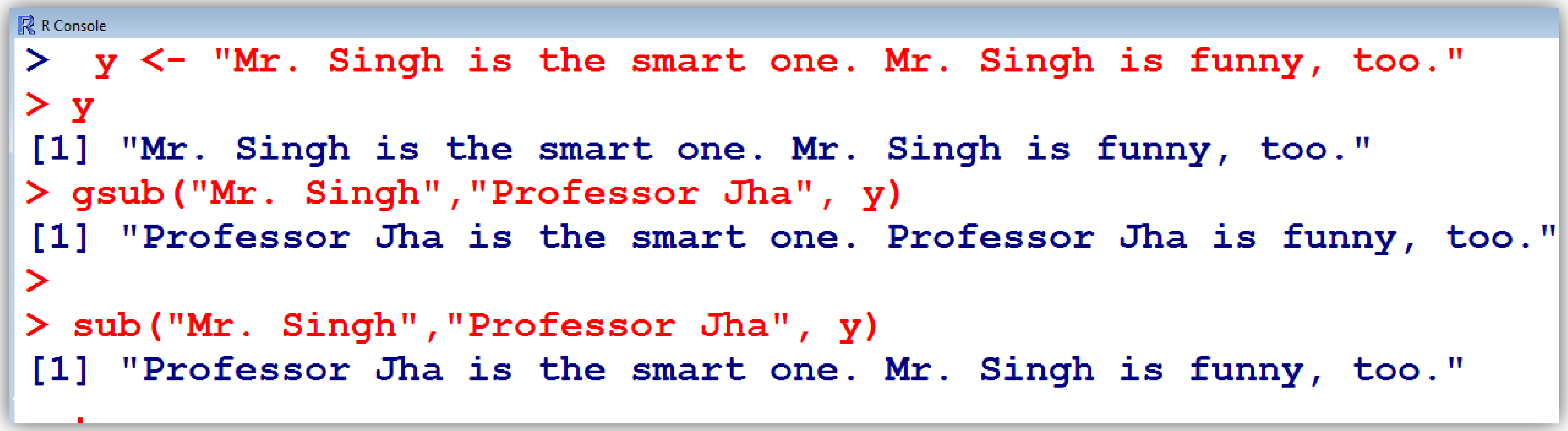
Examples:

```
>  y <- "Mr. Singh is the smart one. Mr. Singh
is funny, too."

> gsub("Mr. Singh","Professor Jha", y)
[1] "Professor Jha is the smart one. Professor
Jha is funny, too."
```

Recall

```
> sub("Mr. Singh","Professor Jha", y)
[1] "Professor Jha is the smart one. Mr. Singh
is funny, too."
```

Operations with Strings



```
>  y <- "Mr. Singh is the smart one. Mr. Singh is funny, too."
> y
[1] "Mr. Singh is the smart one. Mr. Singh is funny, too."
> gsub("Mr. Singh","Professor Jha", y)
[1] "Professor Jha is the smart one. Professor Jha is funny, too."
>
> sub("Mr. Singh","Professor Jha", y)
[1] "Professor Jha is the smart one. Mr. Singh is funny, too."
```

Operations with Strings

`tolower(x)` and `toupper(x)` Functions:

`tolower(x)` and `toupper(x)` convert upper-case characters in a character vector to lower-case, or vice versa.

Non-alphabetic characters are left unchanged.

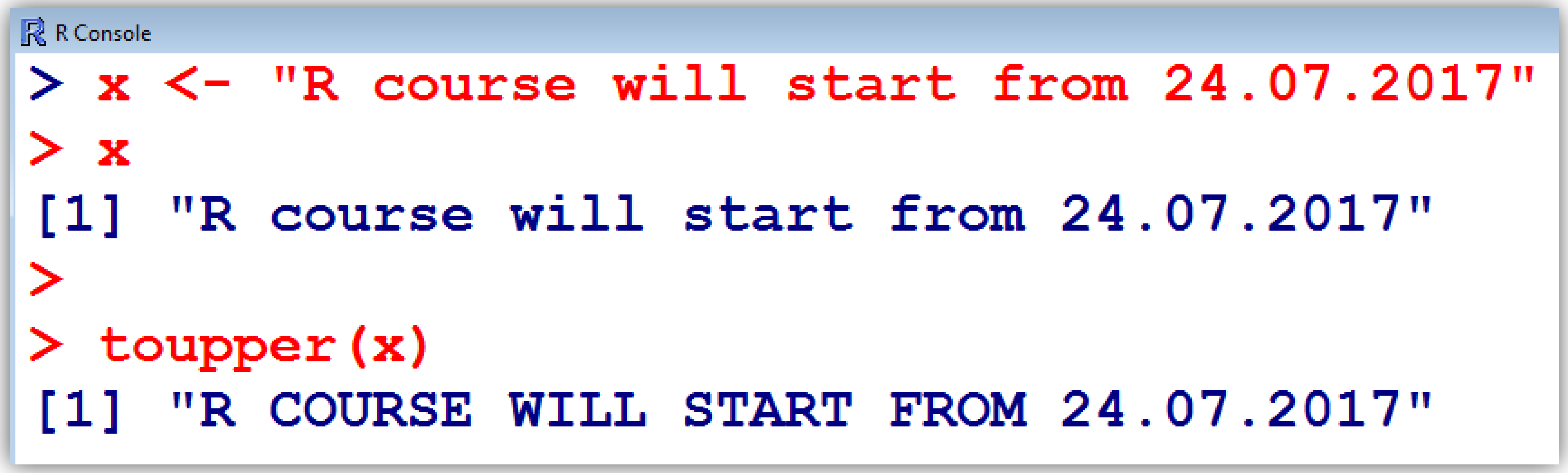
Operations with Strings

`tolower(x)` **and** `toupper(x)` **Functions:**

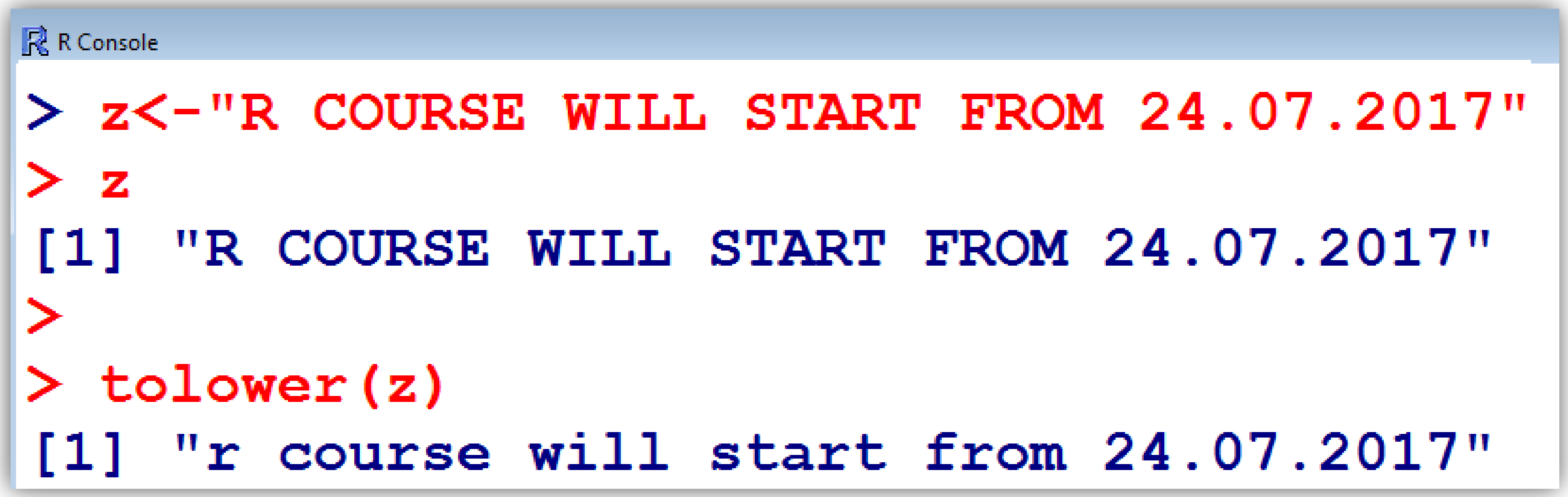
Examples:

```
> x <- "R course will start from 24.07.2017"

> toupper(x)
[1] "R COURSE WILL START FROM 24.07.2017"

> z<-"R COURSE WILL START FROM 24.07.2017"

> tolower(z)
[1] "r course will start from 24.07.2017"
```

Operations with Strings



```
> x <- "R course will start from 24.07.2017"
> x
[1] "R course will start from 24.07.2017"
>
> toupper(x)
[1] "R COURSE WILL START FROM 24.07.2017"
```



```
> z<-"R COURSE WILL START FROM 24.07.2017"
> z
[1] "R COURSE WILL START FROM 24.07.2017"
>
> tolower(z)
[1] "r course will start from 24.07.2017"
```