Introduction to R Software

Strings – Display and Formatting
:::
Paste Function

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Formatting and Display of Strings

paste function

- The `paste()` function concatenates several strings together.

- It creates a new string by joining the given strings end to end.

- The result of `paste()` can be assigned to a variable

  (in contrast to the function `cat()`).

Formatting and Display of Strings

Usage

```
paste (..., sep = " ", collapse = NULL).
```

`collapse` is an optional character string to separate the results.

- The parameter `sep` is a string that serves as a separation between the strings that are given as input.
- `paste` inserts a single space between pairs of strings.
- A desired line break can be achieved with `"\n"` (newline).

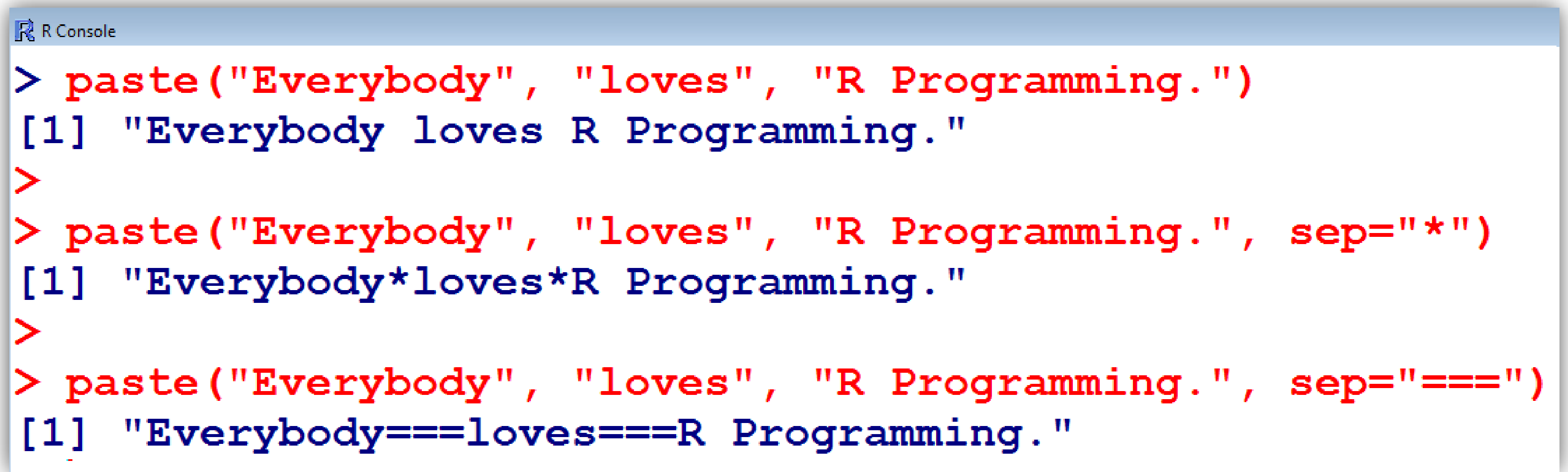
Formatting and Display of Strings

```
> paste("Everybody", "loves", "R Programming.")
[1] "Everybody loves R Programming."
```

```
> paste("Everybody", "loves", "R Programming.",
sep="*")
[1] "Everybody*loves*R Programming."
```

```
> paste("Everybody", "loves", "R Programming.",
sep="===")
[1] "Everybody===loves===R Programming."
```

Formatting and Display of Strings



```
> paste("Everybody", "loves", "R Programming.")
[1] "Everybody loves R Programming."
>
> paste("Everybody", "loves", "R Programming.", sep="*")
[1] "Everybody*loves*R Programming."
>
> paste("Everybody", "loves", "R Programming.", sep="===")
[1] "Everybody===loves===R Programming."
```

Formatting and Display of Strings

If one or more arguments are vectors of strings, `paste` will generate all combinations of the arguments:

```
> names <- c("Prof. Singh", "Mr.  Venkat", "Dr.
Jha")

> paste(names, "is", "a good", "person.")
[1] "Prof. Singh is a good person."
[2] "Mr.  Venkat is a good person."
[3] "Dr. Jha is a good person."
```

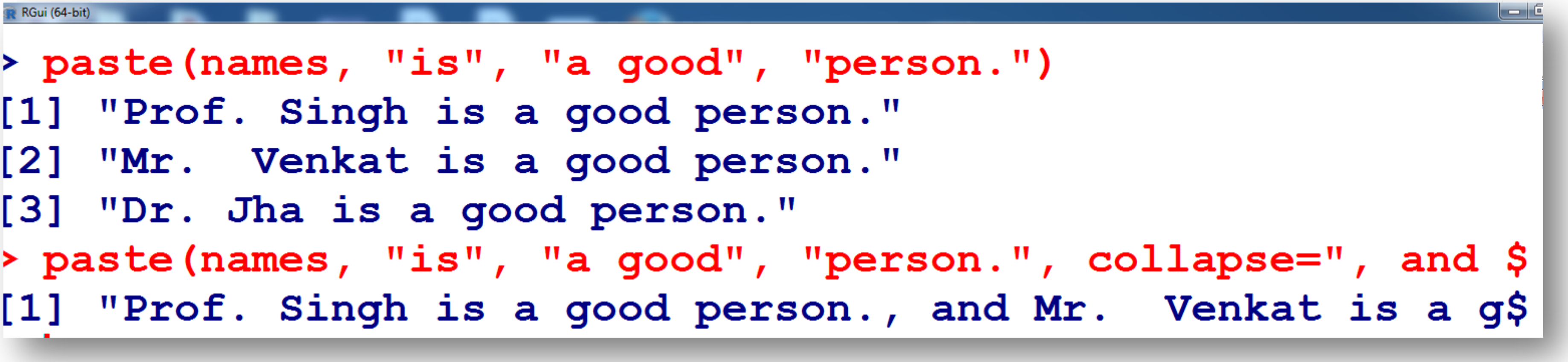
Formatting and Display of Strings

When we want to join even those combinations into one, big string.

The `collapse` parameter defines a top-level separator and instructs `paste` to concatenate the generated strings using that separator:

```
> names <- c("Prof. Singh", "Mr.  Venkat", "Dr.
Jha")

> paste(names, "is", "a good", "person.",
collapse=", and ")
[1] "Prof. Singh is a good person., and Mr.
Venkat is a good person., and Dr. Jha is a good
person.”
```

Formatting and Display of Strings



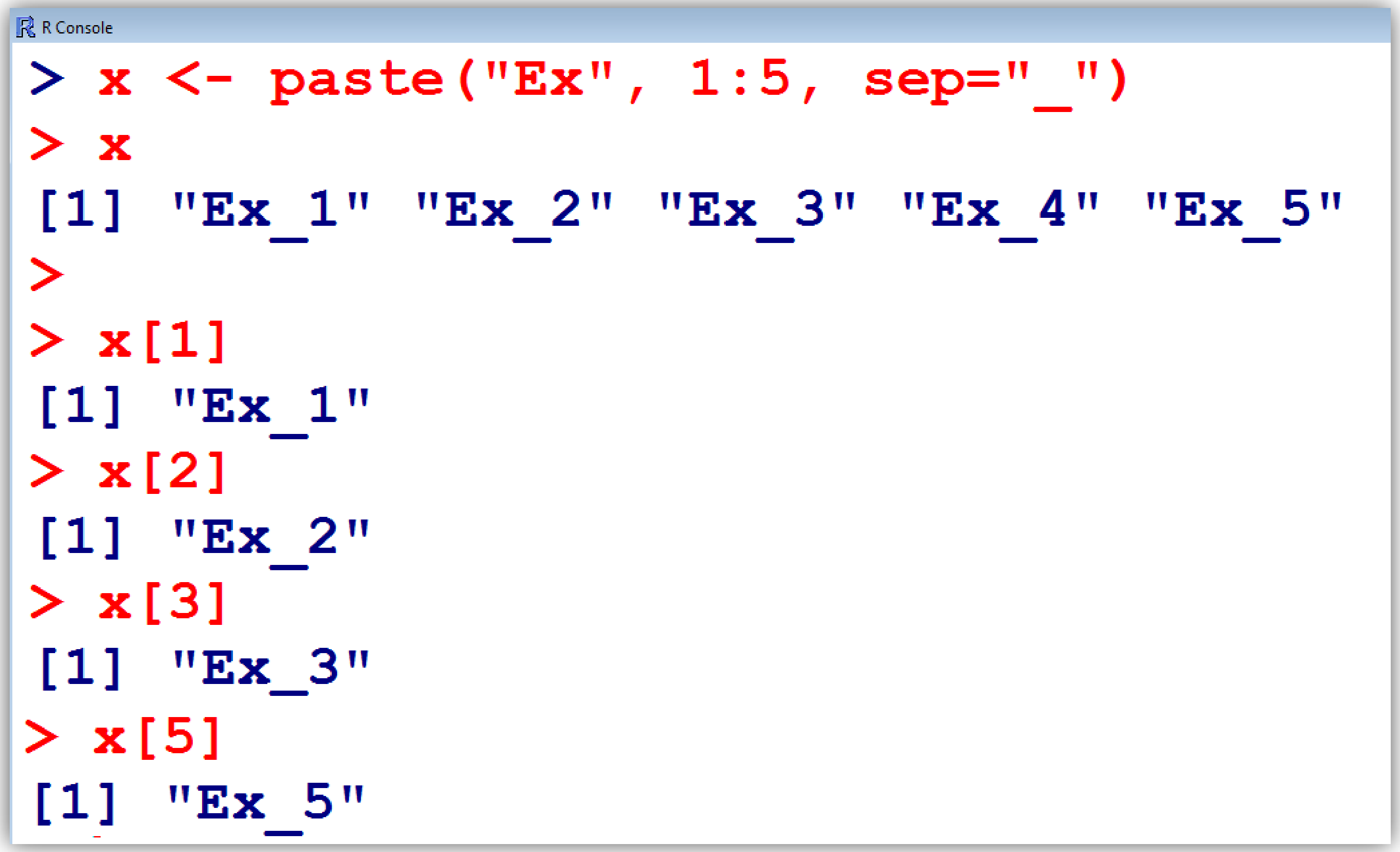
```
> paste(names, "is", "a good", "person.")
[1] "Prof. Singh is a good person."
[2] "Mr.  Venkat is a good person."
[3] "Dr. Jha is a good person."
> paste(names, "is", "a good", "person.", collapse=", and $
[1] "Prof. Singh is a good person., and Mr.  Venkat is a g$
```

Operations with Strings

Example:

```
> x <- paste("Ex", 1:5, sep="_")
>x
[1] "Ex_1" "Ex_2" "Ex_3" "Ex_4" "Ex_5"

> x[1]
[1] "Ex_1"

> x[2]
[1] "Ex_2"

> x[3]
[1] "Ex_3"

> x[5]
[1] "Ex_5"
```

Operations with Strings



```
> x <- paste("Ex", 1:5, sep="_")
> x
[1] "Ex_1" "Ex_2" "Ex_3" "Ex_4" "Ex_5"
>
> x[1]
[1] "Ex_1"
> x[2]
[1] "Ex_2"
> x[3]
[1] "Ex_3"
> x[5]
[1] "Ex_5"
```

Operations with Strings

`x` is a vector of strings.

If we use the parameter `collapse`, a single string, instead of a vector of strings, is created:

```
> x <- paste("Ex", 1:5, sep="_", collapse="")
> x[1]
[1] "Ex_1Ex_2Ex_3Ex_4Ex_5"
```

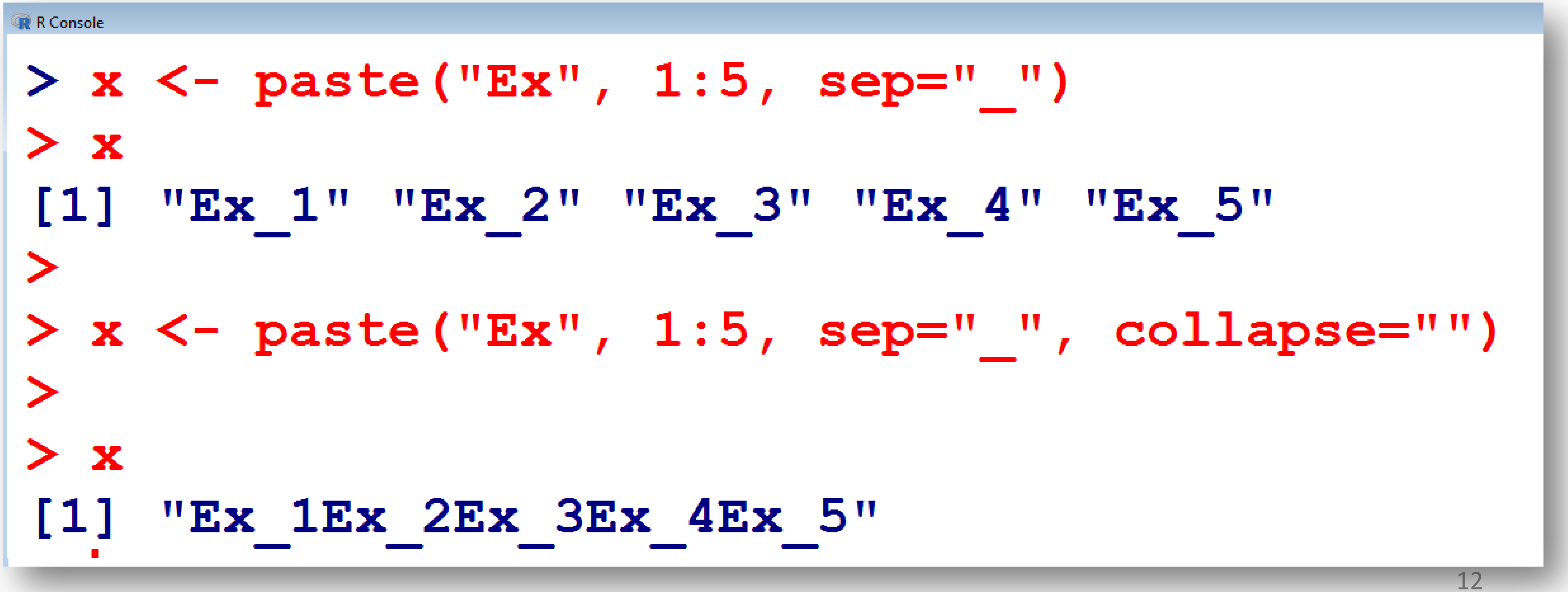
Operations with Strings

Note the difference between

```
x <- paste("Ex", 1:5, sep="_")
```

and

```
x <- paste("Ex", 1:5, sep="_", collapse="")
```



```
> x <- paste("Ex", 1:5, sep="_")
> x
[1] "Ex_1" "Ex_2" "Ex_3" "Ex_4" "Ex_5"
>
> x <- paste("Ex", 1:5, sep="_", collapse="")
>
> x
[1] "Ex_1Ex_2Ex_3Ex_4Ex_5"
```