

Foundations of R Software

Lecture 35

Strings – Display and Formatting

:::

Print and Format Function

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Strings

- **Formatting and Display of Strings**
- **Operations with Strings**

We need formatting and display of strings to obtain the results of specific operations in required format.

Formatting and Display of Strings

Important commands regarding formatting and display are `print`, `format`, `cat`, and `paste`.

`print` function prints its argument.

Usage

`print()`

`print()` is a generic command that is available for every object class.

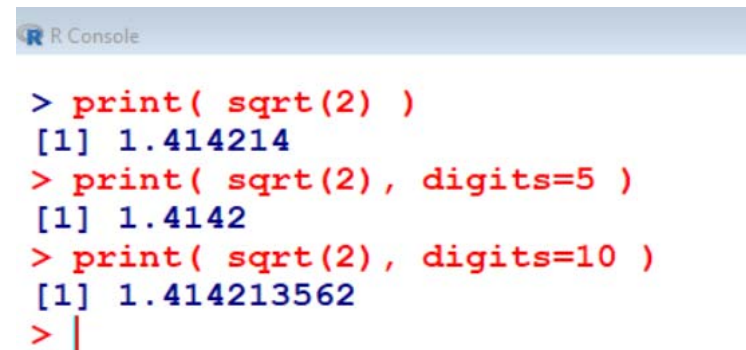
Formatting and Display of Strings

Examples: Print numbers

```
> print( sqrt(2) )  
[1] 1.414214
```

```
> print( sqrt(2), digits=5 )  
[1] 1.4142
```

```
> print( sqrt(2), digits=10 )  
[1] 1.414213562
```

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 the same three R commands and their outputs as the previous blocks, with the prompt character ">" appearing on the final line.

```
> print( sqrt(2) )  
[1] 1.414214  
> print( sqrt(2), digits=5 )  
[1] 1.4142  
> print( sqrt(2), digits=10 )  
[1] 1.414213562  
> |
```

Formatting and Display of Strings

Examples: Print characters

```
> print("apple")  
[1] "apple"
```

```
> print(c("apple", "banana"))  
[1] "apple" "banana"
```

```
> print(c("apple", "banana", 6, 10))  
[1] "apple" "banana" "6" "10"
```

R Console

```
> print("apple")  
[1] "apple"  
> print(c("apple", "banana"))  
[1] "apple" "banana"  
> print(c("apple", "banana", 6, 10))  
[1] "apple" "banana" "6" "10"  
~|
```

Formatting and Display of Strings

Format an R object for pretty printing.

Usage

`format(x, ...)`

`x` is any R object; typically numeric.

Formatting and Display of Strings

Usage

```
format(x, trim = FALSE, digits = NULL, nsmall  
= 0L, justify = c("left", "right", "centre",  
"none"), width = NULL, ...)
```

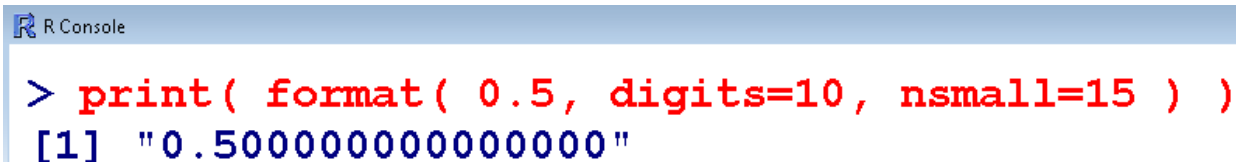
digits shows how many significant digits are to be used

nsmall shows the minimum number of digits to the right of
the decimal point

justify provides left-justified (the default), right-justified, or
centred.

Formatting and Display of Strings

```
> print( format( 0.5, digits=10, nsmall=15 ) )  
[1] "0.5000000000000000"
```



```
R Console  
> print( format( 0.5, digits=10, nsmall=15 ) )  
[1] "0.5000000000000000"
```

Formatting and Display of Strings: Use of width

```
> format(c("A", "BB", "CCC", "DDDD"),  
width = 7, justify = "centre")  
[1] "    A    " "    BB    " "    CCC    " "    DDDD    "
```

```
> format(c("A", "BB", "CCC", "DDDD"),  
width = 14, justify = "centre")  
[1] "          A          " "          BB          " "          CCC          " "          DDDD          "
```

R Console

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify = "centre")  
[1] "    A    " "    BB    " "    CCC    " "    DDDD    "  
>  
> format(c("A", "BB", "CCC", "DDDD"), width = 14, justify = "centre")  
[1] "          A          " "          BB          " "          CCC          " "          DDDD          "  
> |
```

Formatting and Display of Strings: Use of justify

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify  
= "centre")
```

```
[1] "      A      " "      BB      " "      CCC      " "      DDDD      "
```

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify  
= "left")
```

```
[1] "A      " "BB     " "CCC    " "DDDD   "
```

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify  
= "right")
```

```
[1] "      A" "      BB" "      CCC" "      DDDD"
```

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify  
= "none")
```

```
[1] "A"      "BB"     "CCC"    "DDDD"
```

Formatting and Display of Strings: Use of justify

R Console

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify = "centre")
[1] "  A  " " BB " " CCC " " DDDD "
```

>

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify = "left")
[1] "A      " "BB     " "CCC    " "DDDD   "
```

>

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify = "right")
[1] "      A" "     BB" "    CCC" "   DDDD"
```

>

```
> format(c("A", "BB", "CCC", "DDDD"), width = 7, justify = "none")
[1] "A"      "BB"     "CCC"    "DDDD"
```

> |

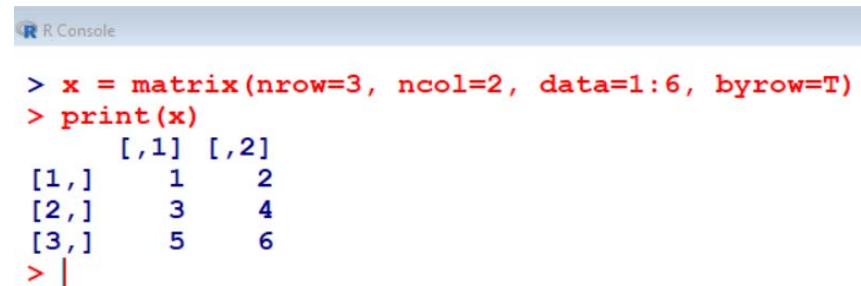
Formatting and Display of Strings

Example:

```
> x = matrix(nrow=3, ncol=2, data=1:6, byrow=T)
```

```
> print(x)
```

```
      [,1] [,2]  
[1,]    1    2  
[2,]    3    4  
[3,]    5    6
```



```
R Console  
  
> x = matrix(nrow=3, ncol=2, data=1:6, byrow=T)  
> print(x)  
      [,1] [,2]  
[1,]    1    2  
[2,]    3    4  
[3,]    5    6  
> |
```

Here, a matrix is displayed in the R command window.

One can specify the desired number of digits with the option **digits**.

Formatting and Display of Large Quantities

Use command `big.mark` for printing large quantities with `format`.

Example, we can print a number with sequences separated by a comma `",`

```
> format(1234567, big.mark = ",")
```

```
[1] "1,234,567"
```

```
> format(12345678, big.mark = ",")
```

```
[1] "12,345,678"
```

```
> format(123456789, big.mark = ",")
```

```
[1] "123,456,789"
```

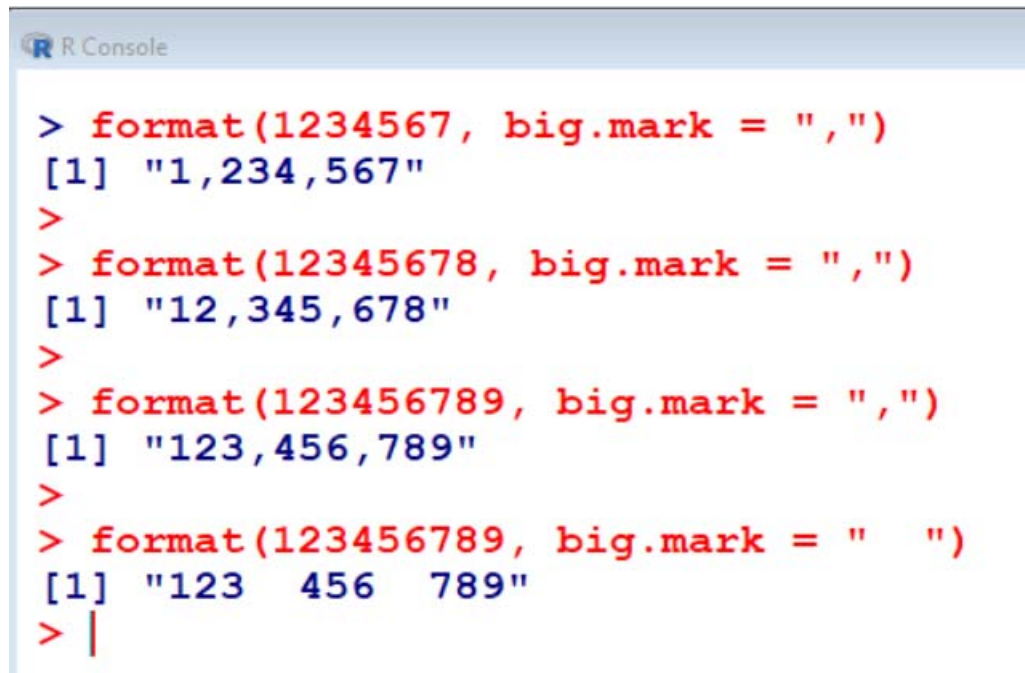
Formatting and Display of Large Quantities

Example, we can print a number with sequences separated by spaces

" "

```
> format(123456789, big.mark = " ")
```

```
[1] "123  456  789"
```



```
R Console
> format(1234567, big.mark = ",")
[1] "1,234,567"
>
> format(12345678, big.mark = ",")
[1] "12,345,678"
>
> format(123456789, big.mark = ",")
[1] "123,456,789"
>
> format(123456789, big.mark = " ")
[1] "123  456  789"
> |
```