

Introduction to R Software

Swayam Prabha

Lecture 25

Display using Print and Format Functions with Concatenate

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Slides can be downloaded from
<http://home.iitk.ac.in/~shalab/sp>



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

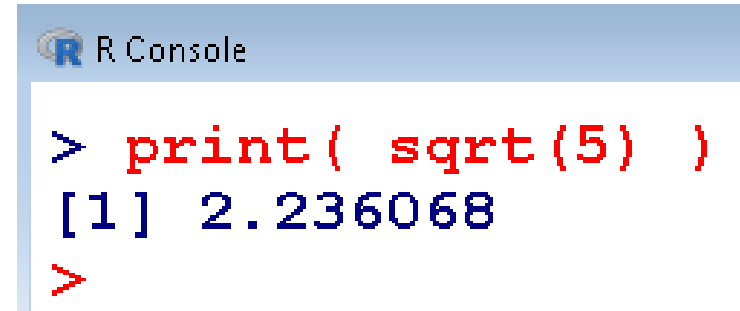
```
print()
```

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

Formatting and Display of Strings

Examples:

```
> print( sqrt(5) )  
[1] 2.236068
```

A screenshot of an R console window. The title bar reads "R Console". The console shows the command `> print(sqrt(5))` in red text, followed by the output `[1] 2.236068` in blue text. A red prompt character `>` is visible on the next line.

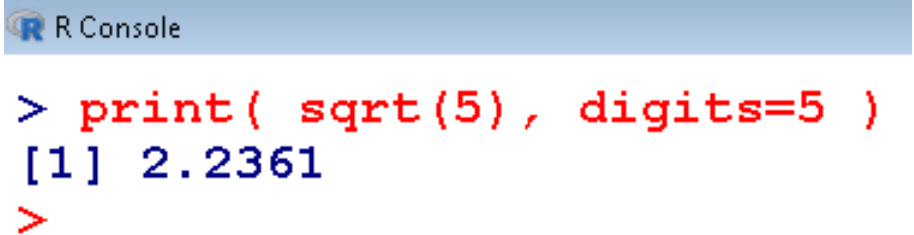
```
R Console  
> print( sqrt(5) )  
[1] 2.236068  
>
```

Formatting and Display of Strings

Examples:

```
> print( sqrt(5), digits=5 )
```

```
[1] 2.2361
```

A screenshot of an R console window. The title bar is light blue and contains the R logo and the text "R Console". The console shows the command `> print(sqrt(5), digits=5)` in red text, followed by the output `[1] 2.2361` in blue text. A red prompt character `>` is visible on the line below the output.

```
> print( sqrt(5), digits=5 )  
[1] 2.2361  
>
```

Formatting and Display of Strings

Format an R object for meaningful 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, ...)
```

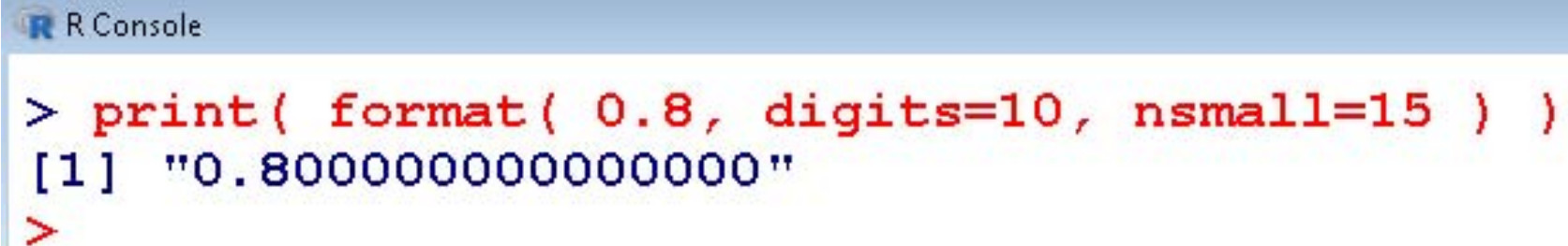
digits shows how many significant digits are to be used

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( 0.8, digits=10, nsmall=15 ) )  
[1] "0.8000000000000000"
```



```
R Console  
> print( format( 0.8, digits=10, nsmall=15 ) )  
[1] "0.8000000000000000"  
>
```


Formatting and Display of Strings

Example:

```
> x <-matrix(nrow=3, ncol=2, data=3:8, byrow=T)
```

```
> print(x)
```

```
      [,1] [,2]  
[1,]    3    4  
[2,]    5    6  
[3,]    7    8
```

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 Strings

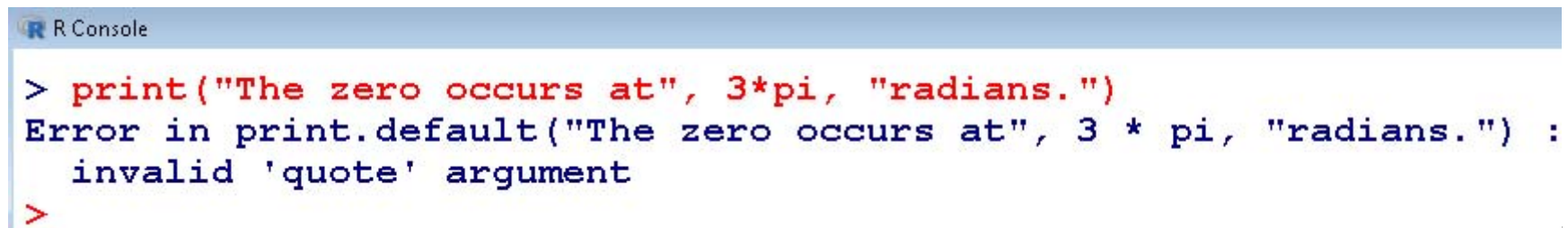
```
R Console  
> x <-matrix(nrow=3, ncol=2, data=3:8, byrow=T)  
>  
> x  
      [,1] [,2]  
[1,]    3    4  
[2,]    5    6  
[3,]    7    8  
>
```

Formatting and Display of Strings

The `print` function has a significant limitation that it prints only one object at a time.

Trying to print multiple items gives error message:

```
> print("The zero occurs at", 3*pi, "radians.")  
Error in print.default("The zero occurs at", 3 *  
pi, "radians.") : invalid 'quote' argument
```



```
R Console  
> print("The zero occurs at", 3*pi, "radians.")  
Error in print.default("The zero occurs at", 3 * pi, "radians.") :  
  invalid 'quote' argument  
>
```

Formatting and Display of Strings

The only way to print multiple items is to print them one at a time

```
> print("The zero occurs at"); print(3*pi);  
print("radians")
```

```
[1] "The zero occurs at"
```

```
[1] 9.424778
```

```
[1] "radians"
```

The `cat` function is an alternative to `print` that lets you combine multiple items into a continuous output.

Formatting and Display of Strings

R Console

```
> print("The zero occurs at"); print(3*pi); print("radians")  
[1] "The zero occurs at"  
[1] 9.424778  
[1] "radians"  
> |
```

Formatting and Display of Strings

cat function

- The function `cat()` concatenate (link in the same sequence) and print the arguments in strings, concatenates them and prints the entire string in the command window.
- `cat` is useful for producing output in user-defined functions.
- It converts its arguments to character vectors, link them together in the same sequence to a single character vector, appends the given `sep = string(s)` to each element and then outputs them.

Formatting and Display of Strings

cat function

Usage

```
cat(... , file = "", sep = " ", ...)
```

`cat` puts a space between each item by default.

One must provide a newline character (`\n`) (newline) to terminate the line.

Formatting and Display of Strings

The only way to print multiple items is to print them one at a time

```
> print("The zero occurs at"); print(3*pi);  
print("radians")  
[1] "The zero occurs at"  
[1] 9.424778  
[1] "radians"
```

The `cat` function is an alternative to print that lets you combine multiple items into a continuous output:

```
> cat("The zero occurs at", 3*pi, "radians.",  
"\n")  
The zero occurs at 9.424778 radians.
```


Formatting and Display of Strings

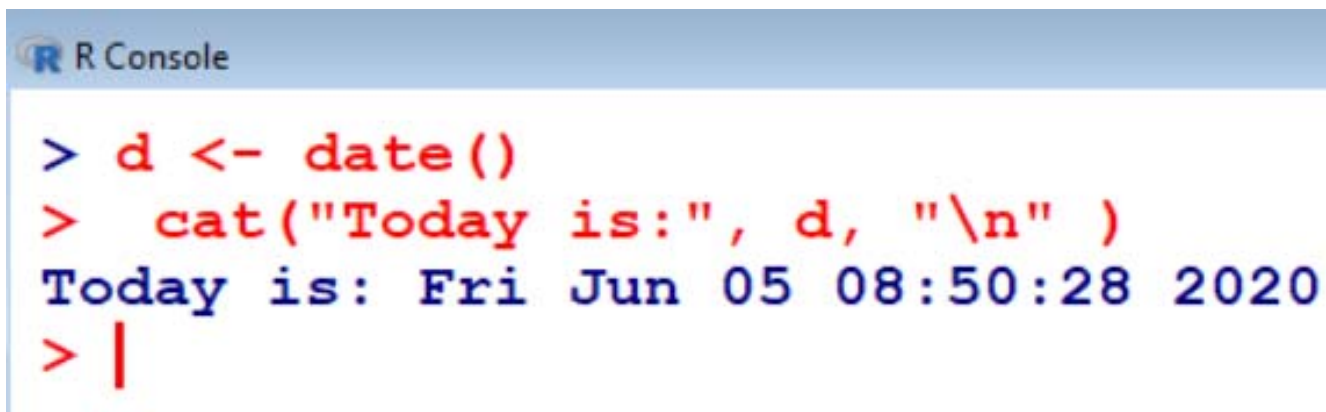
R Console

```
> print("The zero occurs at"); print(3*pi); print("radians")
[1] "The zero occurs at"
[1] 9.424778
[1] "radians"
>
> cat("The zero occurs at", 3*pi, "radians.", "\n")
The zero occurs at 9.424778 radians.
>
```

Formatting and Display of Strings

```
> d <- date()  
> cat("Today is:", d, "\n" )
```

```
Today is: Fri Jun 05 08:50:28 2020
```

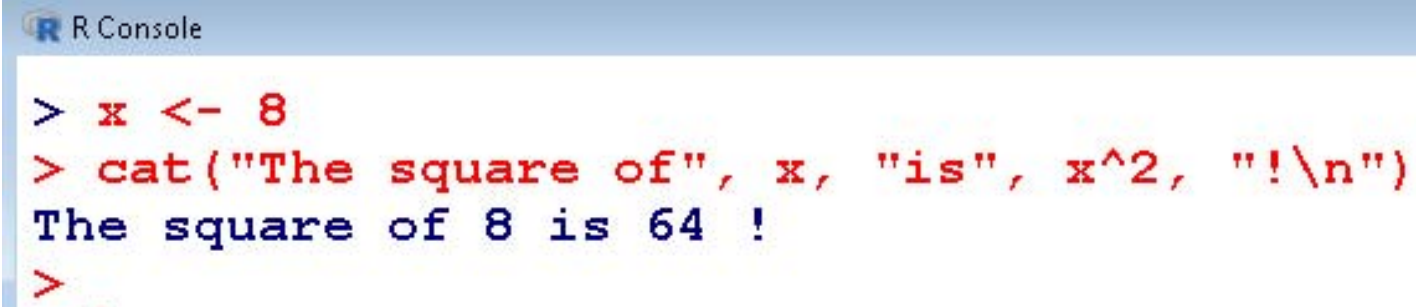


```
R Console  
  
> d <- date()  
> cat("Today is:", d, "\n" )  
Today is: Fri Jun 05 08:50:28 2020  
> |
```

Formatting and Display of Strings

```
> x <- 8  
> cat("The square of", x, "is", x^2, "!\n")
```

The square of 8 is 64 !

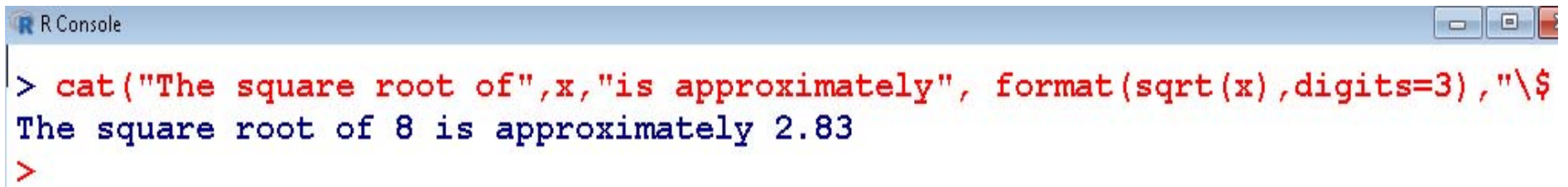


```
R Console  
> x <- 8  
> cat("The square of", x, "is", x^2, "!\n")  
The square of 8 is 64 !  
>
```

Formatting and Display of Strings

```
> cat("The square root of",x,"is  
approximately", format(sqrt(x),digits=3),"\n")
```

The square root of 8 is approximately 2.65



```
R Console  
> cat("The square root of",x,"is approximately", format(sqrt(x),digits=3),"\n")  
The square root of 8 is approximately 2.83  
>
```

Formatting and Display of Strings

The `cat` function can also print simple vectors

```
> evenno <- c(2,4,6,8,10,12)
```

```
> evenno
```

```
[1]  2  4  6  8 10
```

```
> cat("The first few even numbers are:",  
evenno, "... \n")
```

```
The first few even numbers are: 2 4 6 8 10 ...
```

Formatting and Display of Strings

R Console

```
> evenno <- c(2,4,6,8,10,12)
>
> evenno
[1]  2  4  6  8 10 12
>
> cat("The first few even numbers are:", evenno, "...\\n")
The first few even numbers are: 2 4 6 8 10 12 ...
.
```