NOTE: The contents of this document was adapted from <https://www.datamentor.io/r-programming/precedence-associativity/>

**R Operator Precedence and Associativity in R**

**Standard numeric operators, i.e. + - \* / ^**

R recognizes the "order of operations" of the standard mathematical operators, (+ - \* / ^). Many students are familiar with the acronym "PEMDAS" order of operations for the standard arithmetic operators (+ - \* / ^). This stands for:

First: Parentheses  
Second: Exponents  
Third: Multiplication and Division (in the order that they appear)  
Fourth: Addition and Subtraction (in the order that they appear)

> 2 + 6 \* 5 # 6\*5 is done first   
[1] 32  
  
> (2 + 6) \* 5 # 2+6 is done first  
[1] 40

**Additional operators in R**

In addition to the standard mathematical operators, R recognizes numerous other "operators". For example:  
 82 %/% 10 is 8 as the %/% operator performs integer division and   
 82 %% 10 is 2 as the %% operator results in the "remainder" of a division.   
As you learn R you will become familiar with many additional operators.

R defines an "order of operations" that goes beyond PEMDAS. R's complete order of operations includes all of R's operators. This "order of operations" is summarized in the table on the next page. Operators that appear higher in the table have a "higher precedence" (i.e. they are done before operators that appear lower in the table).

For example, you can see that the ^ operator appears before the other arithmetic operators. Similarly, \* and / appear above + and -.  
However, note that %/% and %% appear above \* and /. This would explain why code below produces the output that it does (read the code and the #comments)

> 100 / 25 %% 2 # 25 %% 2 is done first (remember %% is remainder - so 25 %% 2 is 1 and 100/1 is 100)  
[1] 100  
  
> (100 / 25) %% 2 # 100 / 25 is done first (remember %% is remainder - so 100/25 is 4 and 4 %% 2 is 0)  
 [1] 0

**Operator Associativity**

It is possible to have multiple operators of same precedence in an expression. In this case the order of execution is determined through associativity. The associativity of operators is given in the table below. We can see that most of them have left to right associativity.

**Example 2: Operator Associativity in R**

In the above below, 6 / 3 / 2 is evaluated as (6 / 3) / 2 due to left to right associativity of the / operator.   
However, 2 ^ 1 ^ 3 is evaluated as 2 ^ (1 ^ 3) due to right to left associativity of the ^ operator.

> 6 / 3 / 2 # / is left to right associative  
[1] 1  
  
> 2 ^ 1 ^ 3 # ^ is right to left associative  
[1] 2

However, these defaults can be changed by using parentheses ().

> 6 / (3 / 2)  
[1] 4  
  
> (2 ^ 1) ^ 3  
[1] 8

**Precedence and Associativity of different operators in R from highest to lowest**

The list of operators below contains the basic operators but is not 100% complete. For the full list, see the official documentation  
by typing ?Syntax in RStudio   
or at this webpage: <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/Syntax>

|  |  |  |
| --- | --- | --- |
| **Operator Precedence in R** | | |
| **Operator** | **Description** | **Associativity** |
| ^ | Exponent | Right to Left |
| -x, +x | Unary minus, Unary plus | Left to Right |
| %%, %/%, (and any other %% operator) | Modulus (AKA Remainder),  Integer Division | Left to Right |
| \*, / | Multiplication, Division | Left to Right |
| +, – | Addition, Subtraction | Left to Right |
| <, >, <=, >=, ==, != | Comparisons | Left to Right |
| ! | Logical NOT | Left to Right |
| &, && | Logical AND | Left to Right |
| |, || | Logical OR | Left to Right |
| ->, ->> | Rightward assignment | Left to Right |
| <-, <<- | Leftward assignment | Right to Left |
| = | Leftward assignment | Right to Left |