Select Data with Logic

Just as it is for selecting single values, there are multiple ways to subset and filter data from our larger data frames. As with most programming languages, we use a combination of operators and logical statements to tell R what data to filter. Thankfully, most operators are the same between R and Python, as shown below:

Category	R Operator	Description	Python Equivalent
Arithmetical	+	Addition operator	+
	-	Subtraction operator	-
	*	Multiplication operator	*
	/	Division operator	/
	^ or **	Exponent operator	**
	%%	Modulus operator (finds the remainder of the first element divided by the second)	%

Relational	<	Each element in the first data structure is less than each element in the second data structure.	<
	<=	Each element in the first data structure is less than or equal to each element in the second data structure.	<=
	>	Each element in the first data structure is greater than each element in the second data structure.	>
	>=	Each element in the first data structure is greater than or equal to each element in the second data structure.	>=
	==	Each element in the first data structure is equal to each element in the second data structure.	numpy.equal()
	!=	Each element in the first data structure is unequal to each element in the second data structure.	numpy.not_equal()

Logical	x|y	Element-wise OR operator—each element of *x* and *y* structures are combined and returns TRUE if either element is TRUE.	numpy.array(x) | numpy.array(y)
	x&y	Element-wise AND operator—each element of *x* and *y* structures are combined and returns TRUE if both elements are TRUE.	numpy.array(x) & numpy.array(y)
	x||y	Logical OR operator—the first element of *x* and *y* structures are combined and returns TRUE if either element is TRUE.	x[0] or y[0]
	x&&y	Logical AND operator—the first element of *x* and *y* structures are combined and returns TRUE if either element is TRUE.	x[0] and y[0]
Miscellany	isTRUE(x)	Checks if the logic *x* is TRUE, otherwise FALSE.	if x:
	x %in% y	Checks if *x* is contained within *y*.	x in y
	x:y	Creates a range of integer values from *x* to *y*.	range(x,y)