



Marine Data Science

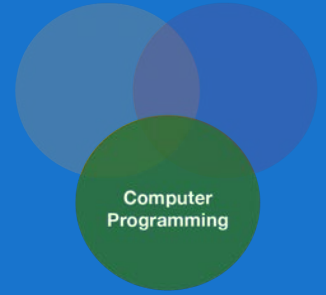


Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Data Analysis with R

4 - Operators

Saskia A. Otto
Postdoctoral Researcher



Operators

Overview of operators

Arithmetic Operators

+	-	*	/	^	%%	%/%
---	---	---	---	---	----	-----

Relational Operators

<	>	==	<=	=>	!=
---	---	----	----	----	----

Logical Operators

!	&		&&	
---	---	--	----	--

Assignment Operators

=	<-	->
---	----	----

Miscellaneous Operators

:	%in%	%*%	%>%
---	------	-----	-----

Relational operators



- find out **relation between two operands**
- **six** relational operations are supported in R
- **output** is **logical** (TRUE or FALSE) for all of these operators
- work **element-wise**

OPERATOR	USAGE	DESCRIPTION
<	$a < b$	a is LESS than b
>	$a > b$	a is GREATER than b
==	$a == b$	a is EQUAL to b
<=	$a <= b$	a is LESS than or EQUAL to b
>=	$a >= b$	a is GREATER than or EQUAL to b
!=	$a != b$	a is NOT EQUAL to b

Example of relational operators



```
# Example for numbers  
a <- 10  
b <- 5  
print(a < b) # less  
print(a >= b) # greater or equal  
print(a != b) # not equal
```

Example of relational operators



```
# Example for numbers
```

```
a <- 10
```

```
b <- 5
```

```
print(a < b) # less
```

```
## [1] FALSE
```

```
print(a >= b) # greater or equal
```

```
## [1] TRUE
```

```
print(a != b) # not equal
```

```
## [1] TRUE
```

Example of relational operators

<	>	==	<=	=>	!=
---	---	----	----	----	----

```
# Example for numbers
```

```
a <- 10
```

```
b <- 5
```

```
print(a < b) # less
```

```
## [1] FALSE
```

```
print(a >= b) # greater or equal
```

```
## [1] TRUE
```

```
print(a != b) # not equal
```

```
## [1] TRUE
```

```
# Example for vectors
```

```
a <- c(7.5, 3, 5)
```

```
b <- c(2, 7, 5)
```

```
print ( a <= b ) # less or equal
```

```
print ( a != b ) # not equal
```


Example of relational operators



```
# Example for numbers
```

```
a <- 10
```

```
b <- 5
```

```
print(a < b) # less
```

```
## [1] FALSE
```

```
print(a >= b) # greater or equal
```

```
## [1] TRUE
```

```
print(a != b) # not equal
```

```
## [1] TRUE
```

```
# Example for vectors
```

```
a <- c(7.5, 3, 5)
```

```
b <- c(2, 7, 5)
```

```
print ( a <= b ) # less or equal
```

```
## [1] FALSE TRUE TRUE
```

```
print ( a != b ) # not equal
```

```
## [1] TRUE TRUE FALSE
```

Logical (boolean) operators

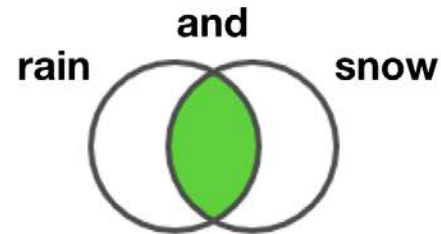


- work only for the **basic data types** (e.g. logical, numeric) and **atomic vectors** in R.

OPERATOR	USAGE	DESCRIPTION
&	a & b	Element -wise logical AND operation.
	a b	Element -wise logical OR operation.
!	!a	Element -wise logical NOT operation.
&&	a && b	Operand -wise logical AND operation.
	a b	Operand -wise logical OR operation.

Umbrellalogic

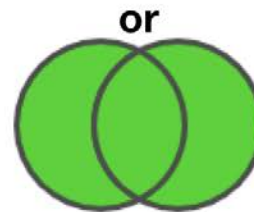
I carry an umbrella if it both rains **and** snows on the same day.



Boolean expression
can be connected:

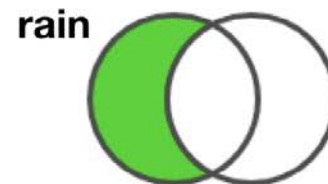
rain & snow

I carry an umbrella whenever it rains **or** snows.



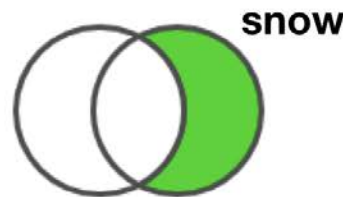
rain | snow

I carry an umbrella for rain but **never for snow**.



rain & ! snow

I **never** carry an umbrella **for rain**, only for snow.



! rain & snow

Example of combined boolean operators

! & | && ||

```
x <- 1:5  
x[ x < 4 & x >= 2]
```

```
## [1] 2 3
```

Example of combined boolean operators



```
x <- 1:5  
x[ x < 4 & x >= 2]
```

```
## [1] 2 3
```

STEP	USAGE	1	2	3	4	5
1	$x < 4$	TRUE	TRUE	TRUE	FALSE	FALSE
2						
3						

Example of combined boolean operators



```
x <- 1:5  
x[ x < 4 & x >= 2]
```

```
## [1] 2 3
```

STEP	USAGE	1	2	3	4	5
1	$x < 4$	TRUE	TRUE	TRUE	FALSE	FALSE
2	$x \geq 2$	FALSE	TRUE	TRUE	TRUE	TRUE
3						

Example of combined boolean operators



```
x <- 1:5  
x[ x < 4 & x >= 2]
```

```
## [1] 2 3
```

STEP	USAGE	1	2	3	4	5
1	$x < 4$	TRUE	TRUE	TRUE	FALSE	FALSE
2	$x \geq 2$	FALSE	TRUE	TRUE	TRUE	TRUE
3	$x < 4 \ \& \ x \geq 2$	FALSE	TRUE	TRUE	FALSE	FALSE

Element- vs. operand-wise operation



```
a <- c(TRUE, TRUE, FALSE, FALSE)
b <- c(TRUE, FALSE, TRUE, FALSE)
```

```
print(a | b)
```

```
## [1] TRUE TRUE TRUE FALSE
```

```
print(a || b)
```

```
## [1] TRUE
```


Other miscellaneous operators

:	%in%	%*%	%>%
---	------	-----	-----

- are similarly important for manipulating data.

OPERATOR	USAGE	DESCRIPTION
:	a:b	Creates series of numbers from left operand to right operand
%in%	a %in% b	Identifies if an element(a) belongs to a vector(b)
%*%	A %*% t(A)	Performs multiplication of a vector with its transpose

Other miscellaneous operators

: %in% %*% %>%

- are similarly important for manipulating data.

OPERATOR	USAGE	DESCRIPTION
:	a:b	Creates series of numbers from left operand to right operand
%in%	a %in% b	Identifies if an element(a) belongs to a vector(b)
%*%	A %*% t(A)	Performs multiplication of a vector with its transpose

Example for %in%

```
a <- c(25, 27, 76)
b <- 27
print(b %in% a)
```

```
## [1] TRUE
```

```
print(a %in% b)
```

```
## [1] FALSE TRUE FALSE
```

Your turn...

Quiz 1: Relational operators

What does the following operation return (try to find the answer without using R):

```
a <- c(6, 80, 107, 164, 208, 53, 216, 268, 65, 283)
a < 60
```

- ☐ NA
- ☐ a numerical vector containing 6 and 53
- ☐ TRUE
- ☐ FALSE
- ☐ a logical vector with TRUEs and FALSEs

Submit

Show Hint

Show Answer

Clear

Quiz 2: Relational operators

How many **TRUE**s would you get from the following operation (try to find the answer without using R):

```
a <- c(6, 80, 107, 164, 208, 53, 216, 268, 65, 283)
a <= 80
```

- ☐ 1
- ☐ 6
- ☐ 4
- ☐ 3

[Submit](#)[Show Hint](#)[Show Answer](#)[Clear](#)

Quiz 3: Relational operators

How many **TRUE**s would you get from the following operation (try to find the answer without using R):

```
a <- c(16, 47, 207)
b <- c(0, 49, 410)
a <= b
```

- ☐ 1
- ☐ 2
- ☐ 5

[Submit](#)[Show Hint](#)[Show Answer](#)[Clear](#)

Quiz 4: Relational operators

What do the following operations on these vectors return:

```
a <- c(4, 5, 1, 8, 8, 10)
b <- c(0, 0, 3, 6, 7, 9); c <- 3
```

1. `a[a < b]`

2. `b[b == c]`

3. `sum(c >= b)`

Submit

Show Hint

Show Answer

Clear

Logical operators

For 6 days it was measured whether it was sunny (sunny = TRUE) and whether it was hot (hot = TRUE). Now we want to check for several conditions (try to find the answer without using R):

```
sunny <- c(TRUE, TRUE, TRUE, FALSE, FALSE, FALSE)
hot <- c(FALSE, TRUE, FALSE, TRUE, FALSE, TRUE)
```


Quiz 5: Logical operators

What does the following return?

```
sunny <- c(TRUE, TRUE, TRUE, FALSE, FALSE, FALSE)
hot <- c(FALSE, TRUE, FALSE, TRUE, FALSE, TRUE)
sunny & hot
```

- ☐ a vector of length 12 (with 6 TRUEs and 6 FALSEs)
- ☐ a vector of length 6 (with 1 TRUE and 5 FALSEs)
- ☐ a vector of length 6 (with 3 TRUEs and 3 FALSEs)

Submit

Show Hint

Show Answer

Clear

Quiz 6: Logical operators

What does the following return?

```
sunny <- c(TRUE, TRUE, TRUE, FALSE, FALSE, FALSE)
hot <- c(FALSE, TRUE, FALSE, TRUE, FALSE, TRUE)
sunny | hot
```

- ☐ a vector with 6 TRUEs
- ☐ a vector with 5 TRUEs and 1 FALSE
- ☐ a vector with 1 TRUE and 5 FALSEs

Submit

Show Hint

Show Answer

Clear

Quiz 7: Logical operators

What does the following return?

```
sunny <- c(TRUE, TRUE, TRUE, FALSE, FALSE, FALSE)
hot <- c(FALSE, TRUE, FALSE, TRUE, FALSE, TRUE)
sunny || hot
```

- ☐ FALSE
- ☐ TRUE

Submit

Show Hint

Show Answer

Clear

Quiz 8: Combining operators

Which values do you get from the following vector:

```
a <- c(6, 80, 107, 164, 208, 53, 216, 268, 65, 283)
```

1. `a[a > 50 & a < 60]`

2. `a[a > a[5] & a < a[8]]`

3. `sum(a > 250 | a < 100)`

4. `sum(a[a %in% 1:60])`

Submit

Show Hint

Show Answer

Clear



Quiz 9 - Challenge: Using operators for subsetting

```
df <- data.frame(  
  sample = letters[1:10],  
  group = c(rep(1, 5), rep(2, 5)),  
  value = c(6, 80, 107, 164, 208, 53, 216, 268, 65, 283)  
)
```

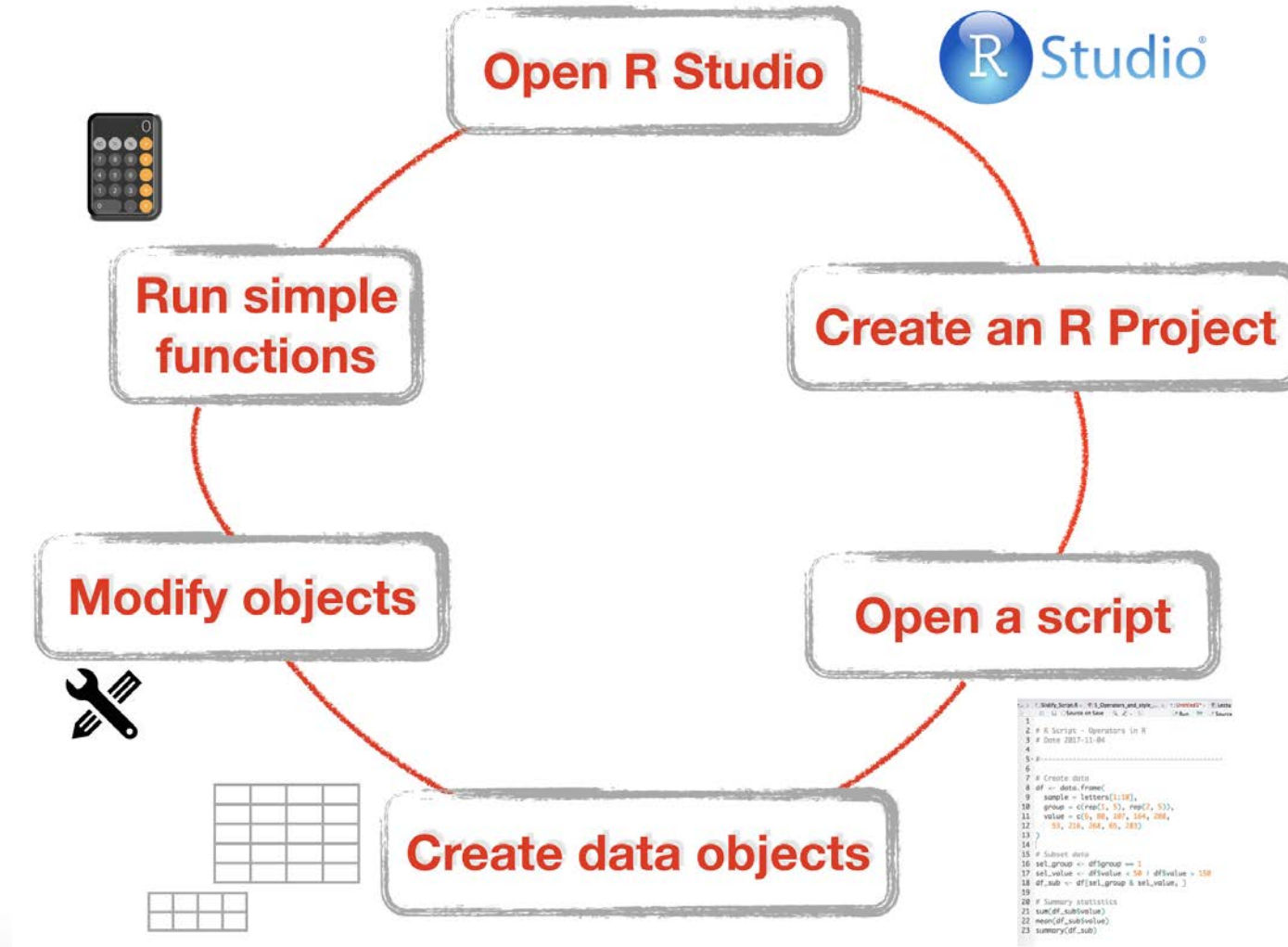
Subset this data frame using the operators you just learned:

1. Extract all observations from group 2
2. Extract all observations where values are greater than 150.
3. Extract all observations from group 1 where values are less than 50 or greater than 250.
4. Extract all observations that have the letters "a", "c", "g", or "j"

(for a hint press p and for a solution code see last slide)

Well Done!!!! You mastered the most important R basics that are fundamental for all that follows.

You are now able to ...



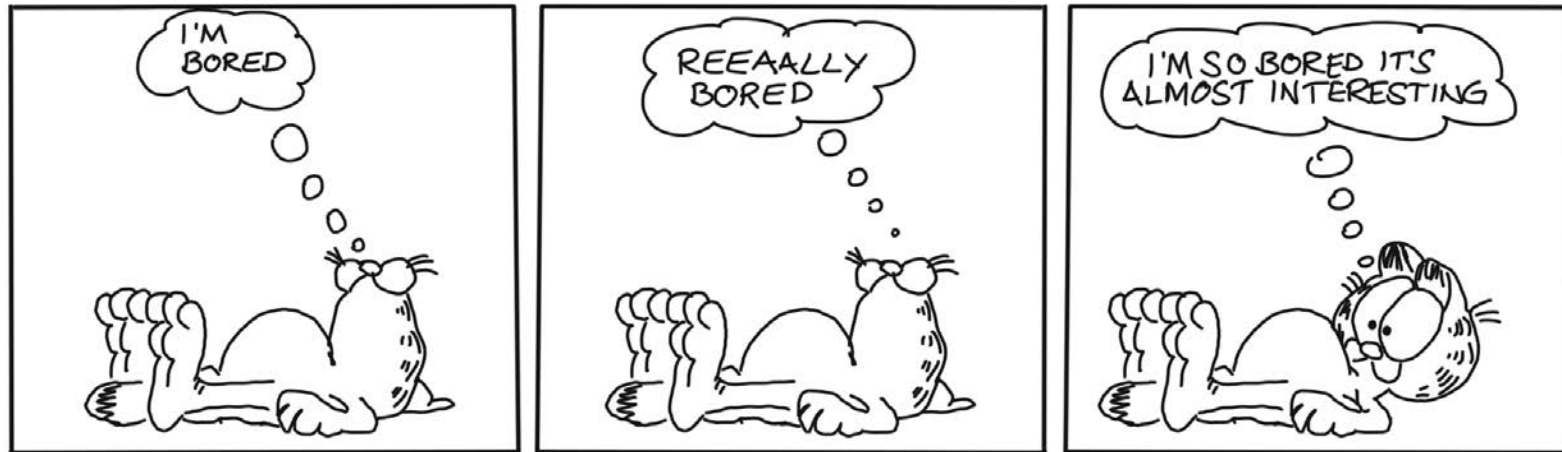
How do you feel now.....?

Totally confused?



Read up on R operators in this very nice tutorial provided on the tutorialcart website:
<https://www.tutorialkart.com/r-tutorial/r-operators/>.

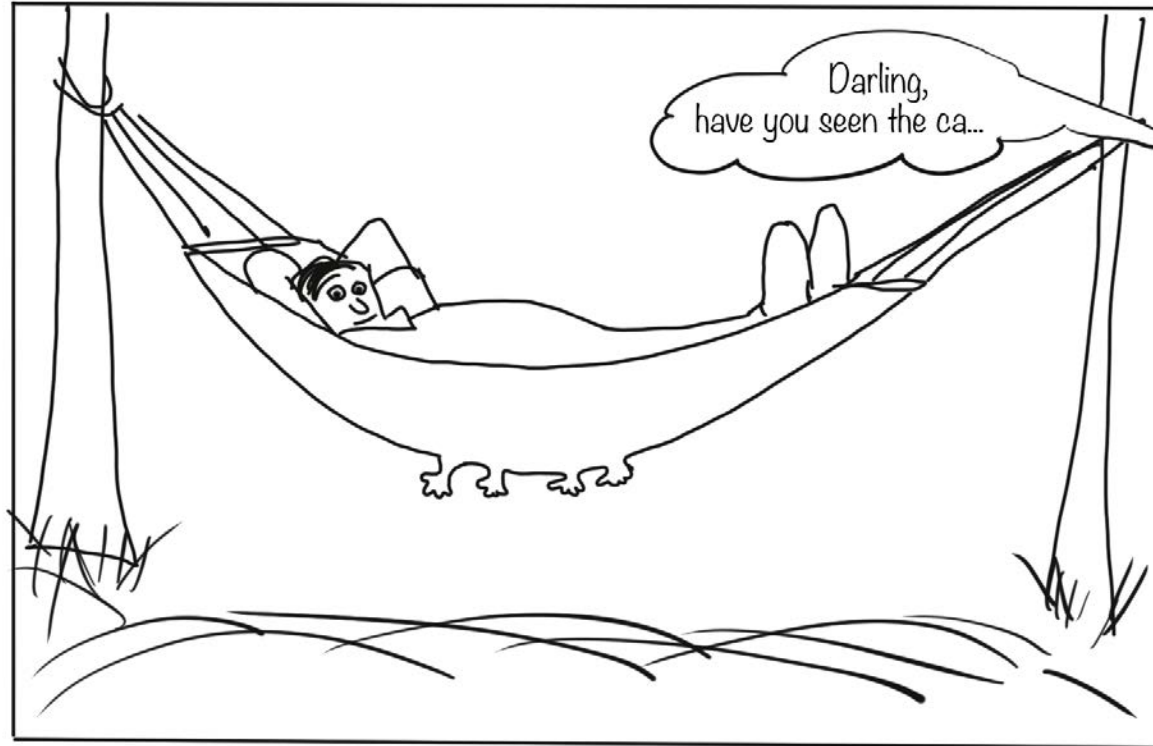
Totally bored?



Don't worry! Soon you won't be bored anymore!!

Totally content?

Then go grab a coffee, lean back and enjoy the rest of the day...!





Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Thank You

For more information contact me: saskia.otto@uni-hamburg.de

http://www.researchgate.net/profile/Saskia_Otto

<http://www.github.com/saskiaotto>



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/) except for the borrowed and mentioned with proper *source*: statements.

Image on title and end slide: Section of an infrared satallite image showing the Larsen C ice shelf on the Antarctic Peninsula - USGS/NASA Landsat: [A Crack of Light in the Polar Dark](#), Landsat 8 - TIRS, June 17, 2017 (under CC0 license)

Solution - Quiz 9

1.Extract all observations from group 2.

```
sel_group <- df$group == 2 # returns a logical vector  
df[sel_group, ] # column index is empty as we want all columns
```

2.Extract all observations where values are greater than 150.

```
sel_value <- df$value > 150  
df[sel_value, ]
```

3.Extract all obs. from group 1 where values < 50 or > 150.

```
sel_group <- df$group == 1  
sel_value <- df$value < 50 | df$value > 150  
df[sel_group & sel_value, ]
```

4.Extract all observations that have the letters "a", "c", "g", or "j".

```
sel_sample <- df$sample %in% c("a", "c", "g", "j")  
df[sel_sample, ]
```