INTERMEDIATE R

Relational Operators

Equality ==

```
> TRUE == TRUE
[1] TRUE

> TRUE == FALSE
[1] FALSE
```

```
> "hello" == "goodbye"
[1] FALSE
```

```
> 3 == 2
[1] FALSE
```

Inequality !=

```
> TRUE != TRUE
[1] FALSE

> TRUE != FALSE
[1] TRUE
```

```
> "hello" != "goodbye"
[1] TRUE
```

```
> 3 != 2
[1] TRUE
```

< and >

```
> 3 < 5
[1] TRUE

> 3 > 5
[1] FALSE

> "Hello" > "Goodbye"
[1] TRUE

> TRUE < FALSE
[1] FALSE
```

Alphabetical Order!

TRUE coerces to 1
FALSE coerces to 0

<= and >=

```
> 5 >= 3
[1] TRUE

> 3 >= 3
[1] TRUE
```

Relational Operators & Vectors

```
> linkedin <- c(16, 9, 13, 5, 2, 17, 14)

> linkedin
[1] 16  9 13  5  2 17 14

> linkedin > 10
[1]  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE

> facebook <- c(17, 7, 5, 16, 8, 13, 14)

> facebook
[1] 17  7  5 16  8 13 14

> facebook <= linkedin
[1] FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
```

INTERMEDIATE R

Let’s practice!

INTERMEDIATE R

Logical Operators

Logical Operators

- AND operator &
- OR operator |
- NOT operator !

AND operator &

```
> TRUE & TRUE
[1] TRUE

> FALSE & TRUE
[1] FALSE

> TRUE & FALSE
[1] FALSE

> FALSE & FALSE
[1] FALSE
```

Only TRUE if both are TRUE

FALSE otherwise

AND operator &

```
> x <- 12

   TRUE     TRUE
> x > 5 & x < 15
[1] TRUE

> x <- 17

   TRUE    FALSE
> x > 5 & x < 15
[1] FALSE
```

OR operator |

```
> TRUE | TRUE
[1] TRUE

> TRUE | FALSE
[1] TRUE

> FALSE | TRUE
[1] TRUE

> FALSE | FALSE
[1] FALSE
```

TRUE if at least one is TRUE

Only FALSE if both FALSE

OR operator |

```
> y <- 4

   TRUE    FALSE
> y < 5 | y > 15
[1] TRUE

> y <- 14

  FALSE    FALSE
> y < 5 | y > 15
[1] FALSE
```

NOT operator !

```
> !TRUE
[1] FALSE

> !FALSE
[1] TRUE

> !(x < 5)
> x >= 5
```

```
> is.numeric(5)
[1] TRUE

> !is.numeric(5)
[1] FALSE

> is.numeric("hello")
[1] FALSE

> !is.numeric("hello")
[1] TRUE
```

Logical Operators & Vectors

```
> c(TRUE, TRUE, FALSE) & c(TRUE, FALSE, FALSE)
[1]  TRUE FALSE FALSE

> c(TRUE, TRUE, FALSE) | c(TRUE, FALSE, FALSE)
[1]  TRUE  TRUE FALSE

> !c(TRUE, TRUE, FALSE)
[1] FALSE FALSE  TRUE
```

& vs &&, | vs ||

```
> c(TRUE, TRUE, FALSE) & c(TRUE, FALSE, FALSE)
[1]  TRUE FALSE FALSE

> c(TRUE, TRUE, FALSE) && c(TRUE, FALSE, FALSE)
[1] TRUE
```

A double ampersand only examines the first element of each vector.

```
> c(TRUE, TRUE, FALSE) | c(TRUE, FALSE, FALSE)
[1]  TRUE  TRUE FALSE

> c(TRUE, TRUE, FALSE) || c(TRUE, FALSE, FALSE)
[1] TRUE
```

vertical bar

INTERMEDIATE R

Use sum() to calculate the number of TRUEs in a vector.

Let’s practice!

INTERMEDIATE R

Conditional Statements

if statement

curly brackets

```
if(condition) {
  expr
}
```

```
> x <- -3

> if(x < 0) {
    print("x is a negative number")
  }
[1] "x is a negative number"
```

if statement

```
if(condition) {
  expr
}
```

```
> x <- 5
      FALSE
> if(x < 0) {
    print("x is a negative number") Not executed
  }
  No printout!
```

else statement

```
if(condition) {
  expr1
} else {
  expr2
}
```

else statement

```
if(condition) {
  expr1
} else {
  expr2
}
```

```
> x <- -3
     TRUE
> if(x < 0) {
    print("x is a negative number")
  } else {
    print("x is either a positive number or zero")
  }
[1] "x is a negative number"
```

else statement

```
if(condition) {
  expr1
} else {
  expr2
}
```

```
> x <- 5
     FALSE
> if(x < 0) {
    print("x is a negative number")
  } else {
    print("x is either a positive number or zero")
  }
[1] "x is either a positive number or zero"
```

else if statement

```
if(condition1) {
  expr1
} else if(condition2) {
  expr2
} else {
  expr3
}
```

else if statement

```
> x <- -3
      TRUE
> if(x < 0) {
    print("x is a negative number")
  } else if(x == 0) {
    print("x is zero")
  } else {
    print("x is a positive number")
  }
[1] "x is a negative number"
```

```
if(condition1) {
  expr1
} else if(condition2) {
  expr2
} else {
  expr3
}
```

else if statement

```
> x <- 0
      FALSE
> if(x < 0) {
    print("x is a negative number")
  } else if(x == 0) { TRUE
    print("x is zero")
  } else {
    print("x is a positive number")
  }
[1] "x is zero"
```

```
if(condition1) {
  expr1
} else if(condition2) {
  expr2
} else {
  expr3
}
```

else statement

```
> x <- 5
      FALSE
> if(x < 0) {
    print("x is a negative number")
  } else if(x == 0) { FALSE
    print("x is zero")
  } else {
    print("x is a positive number")
  }
[1] "x is a positive number"
```

```
if(condition1) {
  expr1
} else if(condition2) {
  expr2
} else {
  expr3
}
```

if, else if, else

```
> x <- 6
          TRUE
> if(x %% 2 == 0) {
    print("divisible by 2")
X } else if(x %% 3 == 0) {
X   print("divisible by 3")
X } else {
X   print("not divisible by 2 nor by 3...")
X }
[1] "divisible by 2"
```

Remember that as soon as R stumbles upon a condition that evaluates to TRUE, R executes the corresponding code and then ignores the rest of the control structure. This becomes important if the conditions you list are not mutually exclusive.

INTERMEDIATE R

Let’s practice!