Relational operators

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

datacamp

A relational example

```
today <- 54.33
yesterday <- 55.24
today < yesterday
```

```
TRUE
```

Relational operators

Format	Description
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	Not equal to

Relational operators

Format	Description
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	**Equal to**
!=	Not equal to

Relational operators

Format	Description
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	Not equal to

Relational operators

Format	Description
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	**Not equal to**

Relational operators

Format	Description
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	Not equal to

Relational operator examples

```
one <- 1
one == TRUE
```

```
TRUE
```

```
apple <- c(120.00, 120.08, 119.97, 121.88)
apple < 121
```

```
TRUE  TRUE  TRUE FALSE
```

Let's practice!

INTERMEDIATE R FOR FINANCE

Logical operators

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

datacamp

Logical operators

Operator	Description	Math representation
&	AND	Conjunction (∧)
|	OR	Disjunction (∨)
!	NOT	Negation (¬)

Logical operators

Operator	Description	Math representation
&	**AND**	Conjunction (∧)
|	OR	Disjunction (∨)
!	NOT	Negation (¬)

Logical operators

Operator	Description	Math representation
&	AND	Conjunction (∧)
|	**OR**	Disjunction (∨)
!	NOT	Negation (¬)

Logical operators

Operator	Description	Math representation
&	AND	Conjunction (∧)
|	OR	Disjunction (∨)
!	**NOT**	Negation (¬)

Logical operators

Operator	Description	Math representation
&	AND	Conjunction (∧)
|	OR	Disjunction (∨)
!	NOT	Negation (¬)

Logical operator examples

```
datacamp <- 64.69

# And
(datacamp > 64) & (datacamp < 65)
```

```
TRUE
```

```
apple <- 124

# Or
(datacamp > 63) | (apple > 126)
```

```
TRUE
```

Logical operators + subset()

```
cash <- data.frame(
  company   = c("A", "A", "A", "B", "B", "B", "B"),
  cash_flow = c(1000, 4000, 550, 1500, 1100, 750, 6000),
  year      = c(1, 3, 4, 1, 2, 4, 5))
```

```
# Filter for rows with low cash flow
subset(cash, cash_flow < 1000)
```

```
  company cash_flow year
3       A       550    4
6       B       750    4
```

```
# Filter for company A's low cash flow
subset(cash, cash_flow < 1000 & company == "A")
```

```
  company cash_flow year
3       A       550    4
```

Let's practice!

INTERMEDIATE R FOR FINANCE

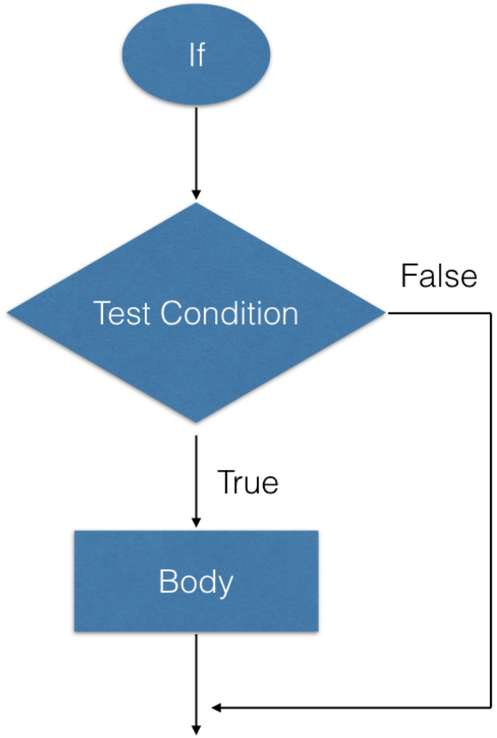
If statements

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

If statements




```
if(condition) {
    code
}
```

Default if...

```
default_chance <- runif(1)
if(default_chance > .5) {
  print("Take action! Likely to default!")
}
```

```
"Take action! Likely to default!"
```

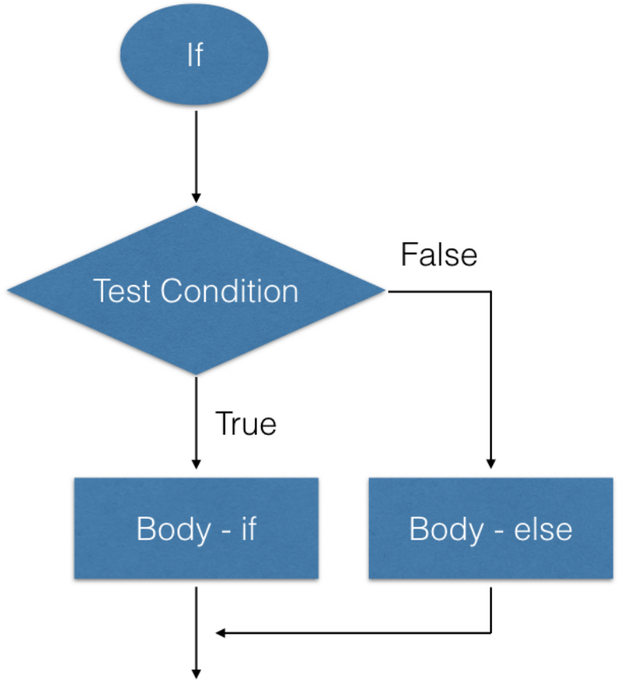
```
default_chance
```

```
0.9484406
```

```
default_chance <- runif(1)
if(default_chance > .5) {
  print("Take action! Likely to default!")
}
default_chance
```

```
0.3954069
```

If-else




```
if(condition) {
  code if true
} else {
    code if false
}
```

Default if...else...

```
default_chance <- .4
```

```
if(default_chance > .5) {
  print("Take action! Likely to default!")
} else {
  print("No problems here!")
}
```

```
"No problems here!"
```

If-else if-else

```
if(condition1) {
  code if condition1 is true
} else if(condition2) {
    code if condition2 is true
} else {
    code if both are false
```

Default if...else if...else

```
default_chance <- .4

if(default_chance > .5) {
  print("Take action! Likely to default!")
} else if(default_chance > .3 & default_chance <= .5) {
  print("Warning! Starting to get risky.")
} else {
  print("No problems here!")
}
```

```
"Warning! Starting to get risky."
```

Let's practice!

INTERMEDIATE R FOR FINANCE