R Syntax 2: Conditions

Pilsung Kang
School of Industrial Management Engineering
Korea University

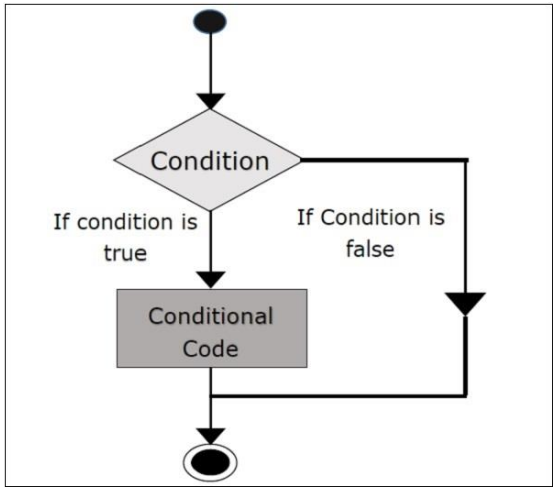
Conditions and Loops

- Understanding conditions and loops are necessary for efficient data analysis
  - ✓ Example of conditions
    - Want to remove instances whose value is greater than 3 standard deviations
    - Want to remove variables with zero variance
    - Want to replace NULL with a constant value
  - ✓ Example of loops
    - Want to make a histogram for each variable in a dataframe
    - Want to compare various machine learning algorithms for the same dataset

Conditions

- if-else condition

```
if (condition) {
statement 1
} else {
statement 2
}
```



✓ condition can be a simple logical comparison to a complex function

✓ statement 1: run if the condition is met

✓ statement 2: run if the condition is not met

Conditions

- Condition example 1

```
# Conditions
r <- 1
if (r==4) {
    print("The value of r is 4”)
} else {
    print(“The value of r is not 4")
}
```

✓ Condition: a simple comparison (ask whether r is 4 or not)

✓ Output: a simple statement (print a sentence)

Conditions

- Condition example 1: Caution!

```
# Caution!
r <- 4
if (r==4) {
    print("The value of r is 4”)
}
else {
    print(“The value of r is not 4")
}
```

✓ must be stated after the right curly bracket in same line

✓ The above code return the error message

```
> # Caution!
> r <- 4
> if (r==4) {
+   print("The valus of r is 4")
+ }
[1] "The valus of r is 4"
> else {
Error: unexpected 'else' in "else"
>   print("The valus of r is not 4")
```

Conditions

- Condition example 2

```
# Computations are possible in the statements
r <- 3
if (r < 5) {
    cat("The value of squared r is", r^2)
} else {
    cat("The value of squared root of r is", sqrt(r))
}
```

✓ Condition: a simple comparison (ask whether r is smaller than 5)

✓ Output: computation result

- If the condition is met (r is smaller than 5), return the square value of r
- If the condition is not met, return the squared root of r

Conditions

- Condition example 3

```
# the results of functions can be a condition
carbon <- c(10, 12, 15, 19, 20)
mean(carbon)
median(carbon)

if (mean(carbon) > median(carbon)) {
    print ("Mean > Median")
} else {
    print ("Median <= Mean")
}
```

✓ Condition can be a result of function

✓ In this example, mean of carbon (15.2) is greater than the median of carbon (15)

✓ Hence, the first statement will be printed

Conditions

- Condition example 4: Simple Form

```
# Simple form
x <- 1
if(x > 0) print("Non-negative number") else print("Negative number")
```

✓ If the statements are simple, the if conditions can be written in one line without curly brackets

Conditions

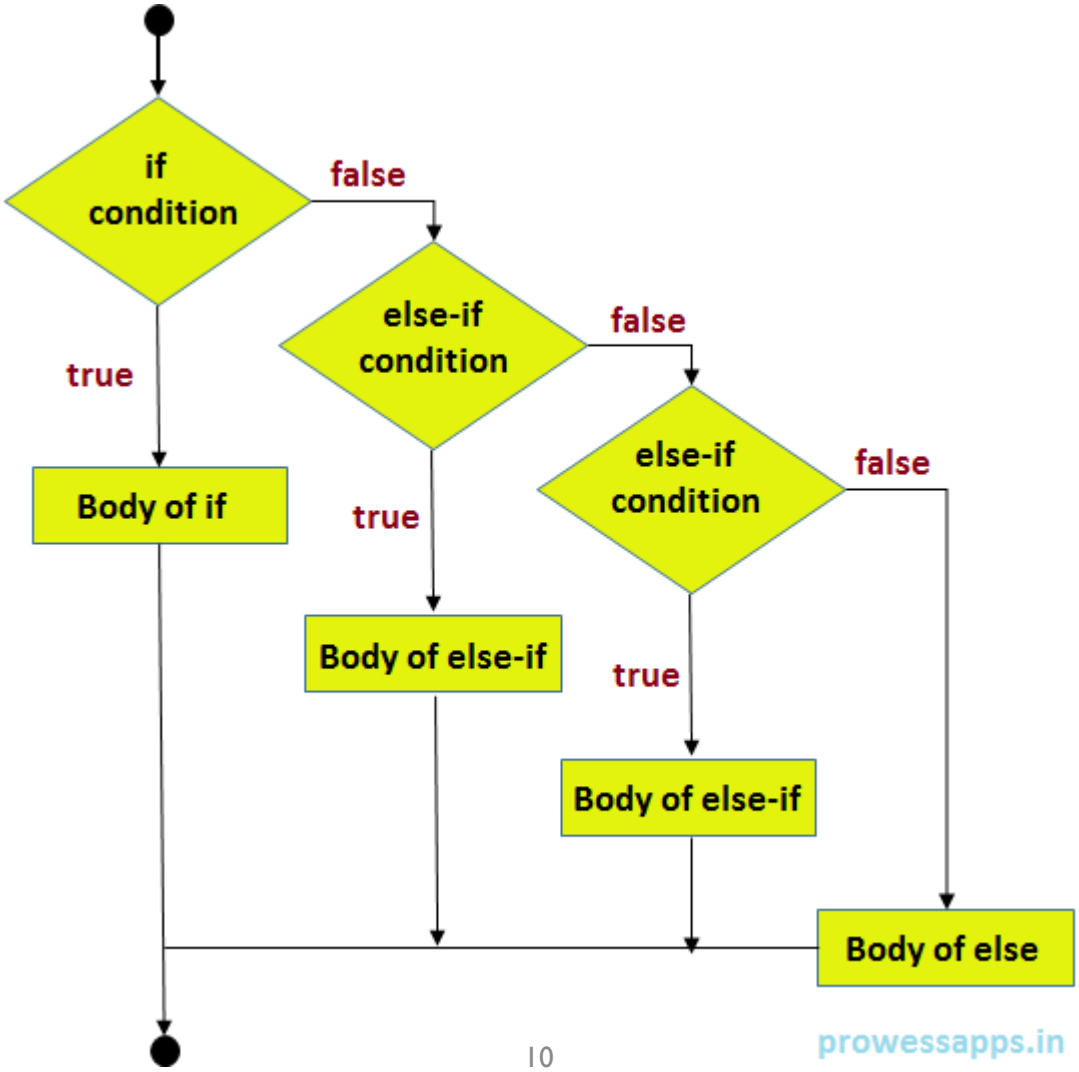
- Condition example 5

```
# variable initialization with if statement
x <- -2
y <- if(x > 0) 1 else -1
y
```

✓ The value of y is initialized by if condition

Conditions

- Condition example 6: if-else ladder



Conditions

- Condition example 6: if-else ladder

```r
# if-else ladder
x <- 0
if (x < 0) {
    print("Negative number")
} else if (x > 0) {
    print("Positive number")
} else print("Zero")
```

Conditions

- Condition example 7: Price calculation
  - ✓ Assume that the tax ratio is different according to the product category

Categories	Products	VAT
A	Book, magazine, newspaper, etc..	8%
B	Vegetable, meat, beverage, etc..	10%
C	Tee-shirt, jean, pant, etc..	20%

```
# Product price calculator w.r.t different category
category <- 'A'
price <- 10 if (category =='A'){
    cat('A vat rate of 8% is applied.','The total price is', price*1.08)
} else if (category =='B'){
    cat('A vat rate of 10% is applied.','The total price is', price*1.10)
} else {
    cat('A vat rate of 20% is applied.','The total price is', price*1.20)
}
```

Conditions

- ifelse: a vectorized condition

  ifelse (condition, statement 1, statement 2)

  ✓ condition: Boolean vector

  ✓ statement 1: run if the condition is met

  ✓ statement 2: run if the condition is not met

```
> x <- 1:10
> y <- ifelse(x%%2 == 0, "even", "odd")
> y
 [1] "odd"  "even" "odd"  "even" "odd"  "even" "odd"  "even" "odd"  "even"
```

Conditions

- Condition example 8: if-else statement

  ifelse (condition, statement 1, statement 2)

  ✓ condition: Boolean vector

  ✓ statement 1: run if the condition is met

  ✓ statement 2: run if the condition is not met

```
# ifelse example
x <- 1:10
y <- ifelse(x%%2 == 0, "even", "odd")
y
```


ANY
questions?