

Eric Pitman Summer Workshop in Computational Science



4. Writing Functions

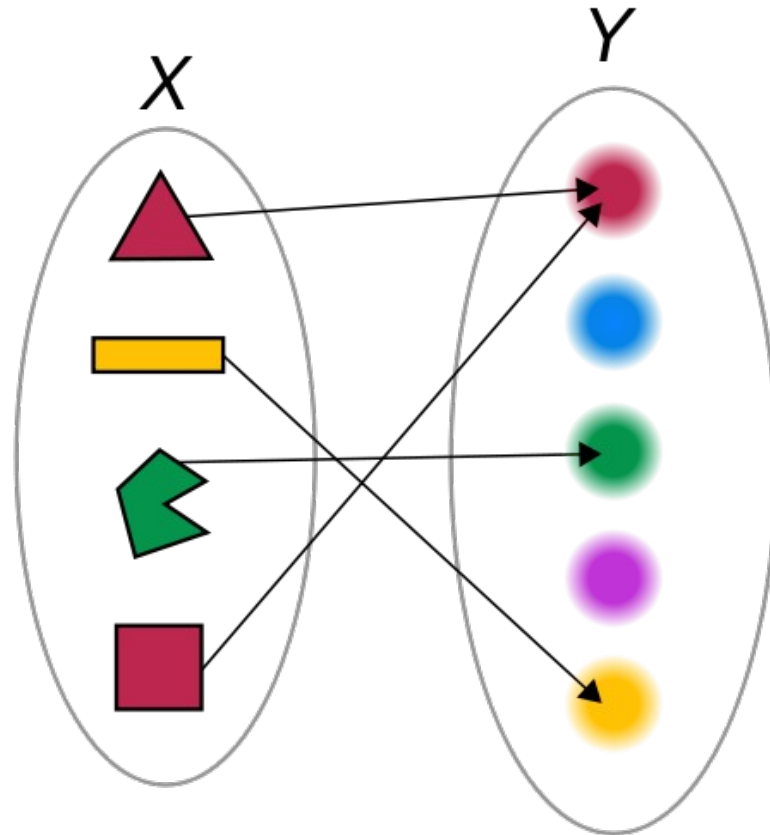


CENTER FOR **COMPUTATIONAL RESEARCH**

UB **University at Buffalo**
The State University of New York

Functions

A function generates an output (Y), given an input (X).



Control Structures: if/else

- Make a logical test
- Perform operations based on the outcome

```
if (condition is true)  
{  
    # do something  
}
```

Control Structures: if/else

```
age = 21;

if (age >= 17) {

    print("You can drive!");

} else if (age >= 16) {

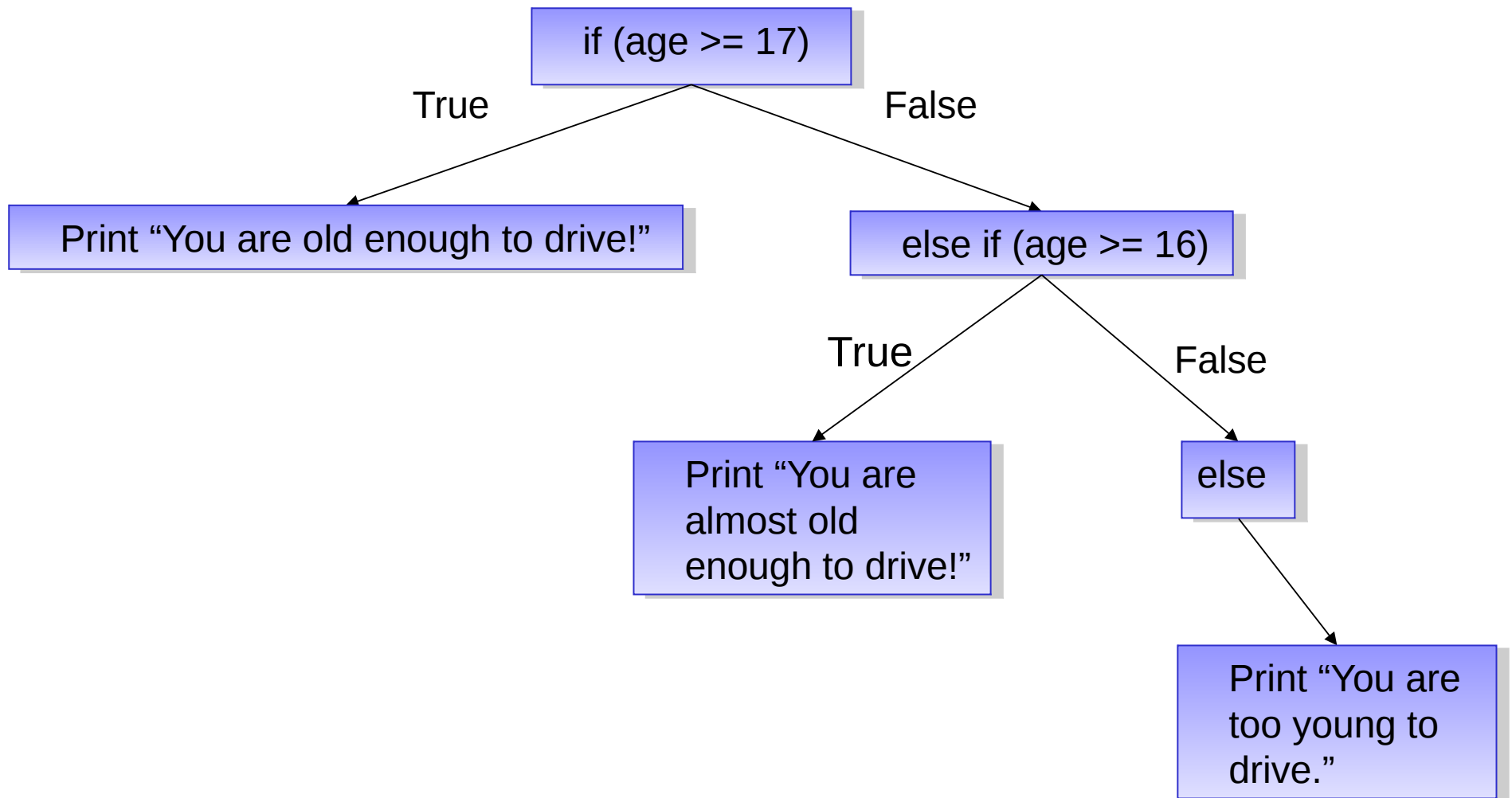
    print("You are almost old enough to drive!");

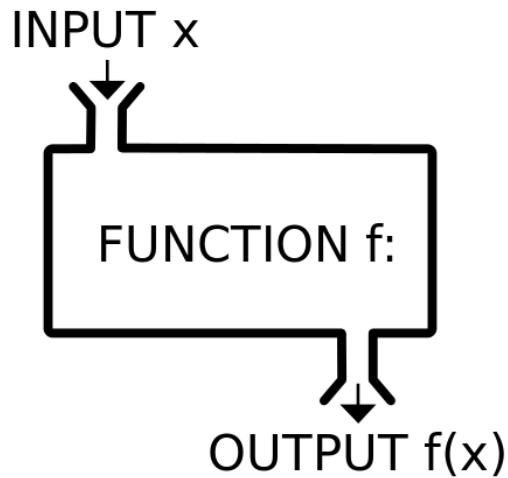
} else {

    print("You are not old enough to drive.");

}
```

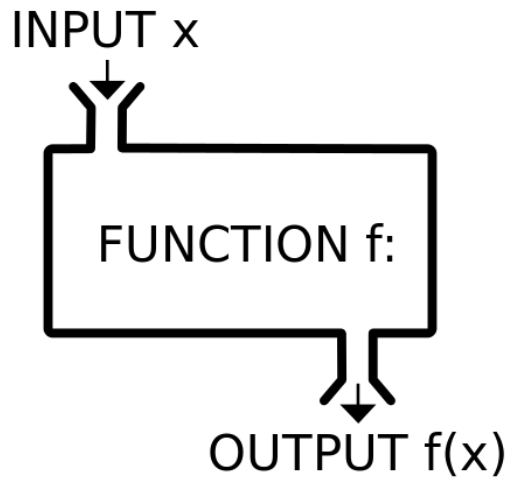
if/else Flowchart





Functions

- A function f takes an input, x , and returns an output $f(x)$.
- It's like a machine that converts an input into an output.



Functions

Function: a piece of code that can be called again and again

To call it, specify:

- Function name
- Input values

It may return an output value

Functions in R

Name of function

Input parameter(s)

```
functionName = function(inputs) {  
  # do something  
  # return the result  
}
```

Declaration
(start of function)

End of function

The diagram illustrates the syntax of an R function. It shows a function definition: 'functionName = function(inputs) { # do something # return the result }'. Annotations include: 'Name of function' pointing to 'functionName', 'Input parameter(s)' pointing to 'inputs', 'Declaration (start of function)' pointing to the opening curly brace '{', and 'End of function' pointing to the closing curly brace '}'.

Functions in R

Name of function

Input parameter(s)

```
toFahrenheit = function(celsius) {  
  f = (9/5) * celsius + 32; # do something  
  return(f); # return the result  
}
```

Declaration
(start of function)

Output value

End of function

Functions in R

```
toFahrenheit = function(celsius) {  
  f = (9/5) * celsius + 32; # do something  
  return(f); # return the result  
}
```

Functions in R

```
celsius = c(20:25); # define input temperatures
```

```
toFahrenheit = function(celsius) {
```

```
  f = (9/5) * celsius + 32; # perform the conversion
```

```
  return(f);
```

```
}
```

```
# call the function to convert temperatures to Fahrenheit:
```

```
toFahrenheit(celsius);
```

```
[1] 68.0 69.8 71.6 73.4 75.2 77.0
```

Control Structures for Iteration

- In other languages we write loops
- But R is a (functional, vector) language
- We can operate on multiple data subsets with one line of code!

```
apply()  
by()
```

Control Structures: apply() Family

- What if we want to call a function over and over?
- We can do this with a single line of R code!
- Use it on native R functions, or functions you wrote yourself.

```
sapply(vector, function)
```

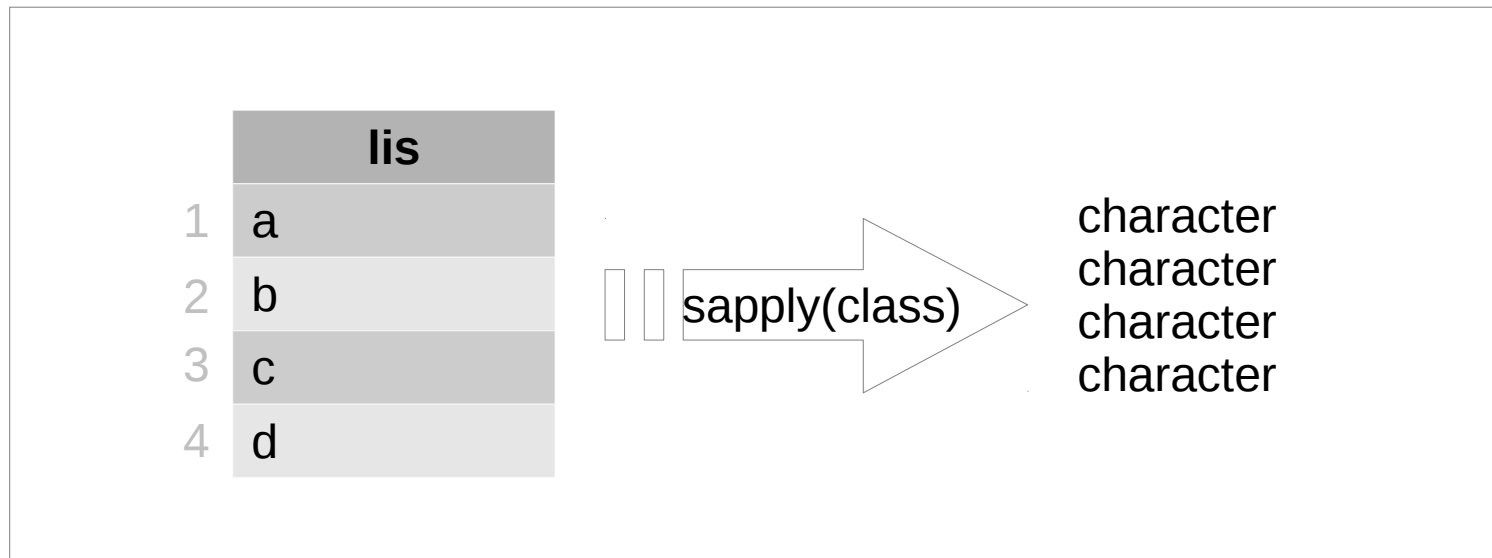
Control Structures: sapply()

```
> lis = c("a", "b", "c", "d")
```

```
> sapply(lis, class)
```

a b c d

```
"character" "character" "character" "character"
```



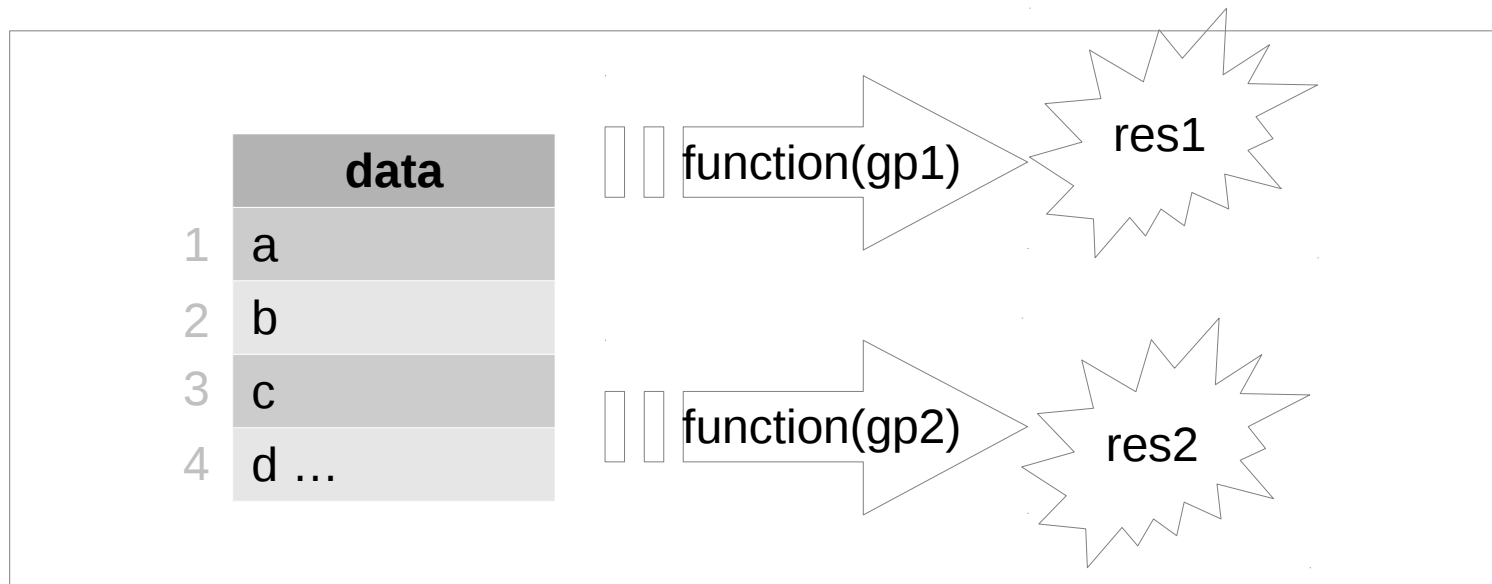
Control Structures: `by()`

- What if we want to call a function several times, on several *groups* of data?
- We can use a single line of R code:

```
by(data, group, function)
```

Control Structures: by()

by(data-to-operate-on,
data-to-group-by,
function)



iris and by()



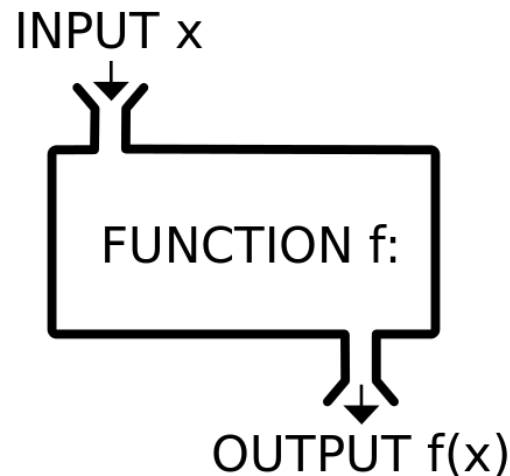
| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |

Compute summaries and means of data, grouping by Species:

```
<workshop>/examples/by-example.R
```

Tips: Writing Functions

- Use an editor window (not the command line) to compose functions
- Try out one line at a time, and test!
- Start with the simplest case and build.
- Comment your function to indicate:
 - input
 - output
 - purpose



Student Dataset Example



Remember our own dataset:

`firstInitial`, `lastInitial`, `school`, `height`, `htUnit`, `age`,
`handed`, `gender`

Let's write functions that:

- Convert heights to a uniform unit
- List initials of students that are old enough to drive

Interlude

Complete function exercises.



Open in the RStudio source editor:

`<workshop>/exercises/exercises-functions.R`