

Supplement: for() loops and alternatives

BIOSTAT 213: Introduction to Computing in the R Software Environment

Yea-Hung Chen, PhD, MS

for() loops

for() loops

Supplement:
for() loops and
alternatives

Yea-Hung Chen,
PhD, MS

for() loops

- ▶ for() loops provide a way to **repeat code**, with changing values

for() loops

- ▶ the **general form** is as follows:

```
for(arbitrary.name in some.vector){  
  # code here may use arbitrary.name  
}
```

- ▶ the loop **cycles through each element of some.vector**
- ▶ **arbitrary.name**: an arbitrary name for each cycle's value of some.vector
- ▶ **some.vector**: a vector of values to cycle through
- ▶ the **curly brackets**: a space for the code to repeat for each iteration

Example: squares

```
squares<-NULL
for(number in seq(0,12,3)){
  squares<-c(squares,number^2)
}
```

- ▶ the first line defines an empty object
- ▶ the loop cycles through each number in `seq(0,12,3)`
- ▶ `number`: an arbitrary name for each value in `seq(0,12,3)`
- ▶ `seq(0,12,3)`: a vector of values to cycle through
- ▶ in each iteration, the loop adds `number^2` to `squares`

Example: squares

```
> squares
```

```
[1] 0 9 36 81 144
```

- ▶ this was a just a [learning example](#)
- ▶ it would have been easier to simply square `seq(0,12,3)`:

```
> seq(0,12,3)^2
```

```
[1] 0 9 36 81 144
```

Example: cycling through variables

```
number.missing<-NULL
for(variable.name in names(ii)[1:10]){
  current<-sum(is.na(ii[,variable.name]))
  names(current)<-variable.name
  number.missing<-c(number.missing,current)
}
```

- ▶ the first line defines an empty object
- ▶ the loop cycles through each variable name in `names(ii)[1:10]`
- ▶ `variable.name`: an arbitrary name for each value of `names(ii)[1:10]`
- ▶ in each iteration, the loop adds the number of missing values to `number.missing`

Example: cycling through variables

Supplement:
for() loops and
alternatives

Yea-Hung Chen,
PhD, MS

for() loops

```
> number.missing
      seqn sddsrvyr ridstatr riagendr ridageyr ridagemn
         0         0         0         0         0      4209
ridreth1 ridreth3 ridexmon ridexagm
         0         0         0      3981
```

When and when not to use `for()` loops

Supplement:
`for()` loops and
alternatives

Yea-Hung Chen,
PhD, MS

`for()` loops

- ▶ `for()` loops are somewhat controversial
- ▶ one critique is that they can be slow

Example: cycling through variables

- ▶ the apply functions can often be used in place of for():

```
sapply(names(ii)[1:10],FUN=function(variable.name){  
  sum(is.na(ii[,variable.name]))  
})
```

seqn	sddsrvyr	ridstatr	riagendr	ridageyr	ridagemn
0	0	0	0	0	4209
ridreth1	ridreth3	ridexmon	ridexagm		
0	0	0	3981		