

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

Alternatives

- ▶ 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` is an arbitrary name for each cycle's value of `some.vector`
- ▶ the code within the curly brackets may use `arbitrary.name`

Example: squares

Supplement:
for() loops and
alternatives

Yea-Hung Chen,
PhD, MS

for() loops

Alternatives

```
> squares<-NULL
> for(number in seq(0,12,3)){
+   squares<-c(squares,number^2)
+ }
> squares
[1]    0    9   36   81  144
```

- ▶ the first line defines an empty object
- ▶ the loop cycles through each number in seq(0,12,3)
- ▶ in each iteration, the loop adds number^2 to squares

Example: cycling through variables

Supplement:
for() loops and
alternatives

Yea-Hung Chen,
PhD, MS

```
> 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)
+ }
```

for() loops

Alternatives

- ▶ the loop cycles through each variable name in `names(ii)[1:10]`
- ▶ in each iteration, the loop adds the number of missing values to `number.missing`
- ▶ `current` is an arbitrary name referring to the number of missing values that the current variable (`variable.name`) contains
- ▶ `names()` provides `current` with a name

Alternatives

Alternatives

Supplement:
for() loops and
alternatives

Yea-Hung Chen,
PhD, MS

for() loops

Alternatives

- ▶ for() loops are **sometimes** discouraged because they can be computationally inefficient
- ▶ they can also be difficult to read

Example: squares

- ▶ for() loops can often be avoided by taking advantage of the fact that **most** operators and functions in R work on vectors:

```
> squares<-seq(0,12,3)^2  
> squares  
[1] 0 9 36 81 144
```

Example: cycling through variables

Supplement:
for() loops and
alternatives

Yea-Hung Chen,
PhD, MS

for() loops

Alternatives

- ▶ the apply functions are also useful alternatives to 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      5769  
ridreth1 ridreth3 ridexmon ridexagm  
         0         0        181      5769
```