

Tutorial 10 - Really programming in R

Parentheses, Square Brackets, Curly Braces, oh my!

() using functions

[] subsetting

{ } control flow: for loops, if-else statements, defining functions

Cheatsheet for R

On today's Sakai page

or directly from the source:

<https://rstudio.com/wp-content/uploads/2016/10/r-cheat-sheet-3.pdf>

Includes most of what we have covered, but some extra stuff you can ignore.

Indexing Review

- ▶ To index a vector or matrix, you need square brackets. For example...

```
myVector[2]
```

```
myMatrix[3,]
```

- ▶ To index a data frame, you need square brackets and/or a dollar sign. For example...

```
myDataFrame[3,2]
```

or

```
myDataFrame$column2name[3]
```

Indexing Review

Given a data frame called `data` that has 4 rows and 3 columns (with column names A, B, C), what code, without using any loops, would:

- ▶ `print the 2nd column`
- ▶ `print each row, one at a time`
- ▶ `print each element of the 3rd column, one at a time`

for loops in R

In Bash, we looped through files in directories, with code like

```
for file in $@
```

In R, we commonly load full files into R and need to combine sets of integers with subsetting to loop through these data structures.

```
for(line in 1:10){  
    x <- myDF[line,]  
    print(x)  
}
```

for loop Review

Given a data frame called `data` that has 4 rows and 3 columns (with column names A, B, C), what code, USING LOOPS, would:

- ▶ print each row, one at a time
- ▶ print each element of the 3rd column, one at a time

Tips for working with loops

- ▶ when writing a loop, work on a single case, get it to work, and then generalize for each row
- ▶ think of the index variable as placeholder for each integer in the set you'll loop through
- ▶ use print statements to figure out which parts are working and which are not

if-else

This is a useful way to let your code make decisions for you

Given the outcome of a logic test, one or more behaviors can occur

```
if(x > 0){  
    print("x is positive")  
}else if(x < 0){  
    print("x is negative")  
}else{  
    print("x is equal to zero")  
}
```

if-else in a loop

You can even use if-else statements in loops. Actually this is where they are most useful!

Challenge:

Use a `for` loop and `if-else` statement to calculate the sum of wages for males and females in `wages.csv`.

Challenge:

- ▶ Use a for loop to calculate the average sepal length for each species in the iris data set. Don't cheat and use the `mean()` function!
- ▶ Use a for loop and if-else statement to find the minimum petal width of Setosa iris in the iris data set. Don't cheat and use the `min()` function!

Exercise 8

Due next Friday, 10/23

Fork from TA's Github repository

Remember R scripts are just text files and so they can be version controlled with `Git` just like any other file.

You'll need to use `linux` to interact with `Git`