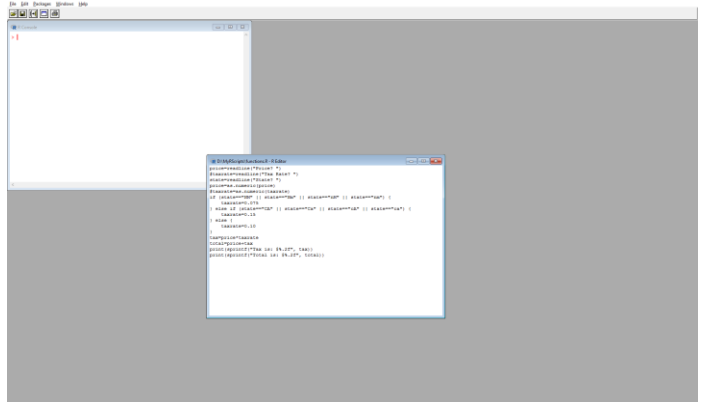
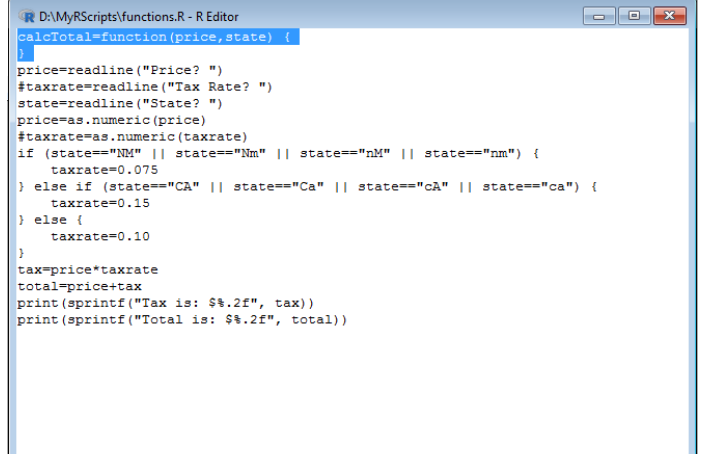
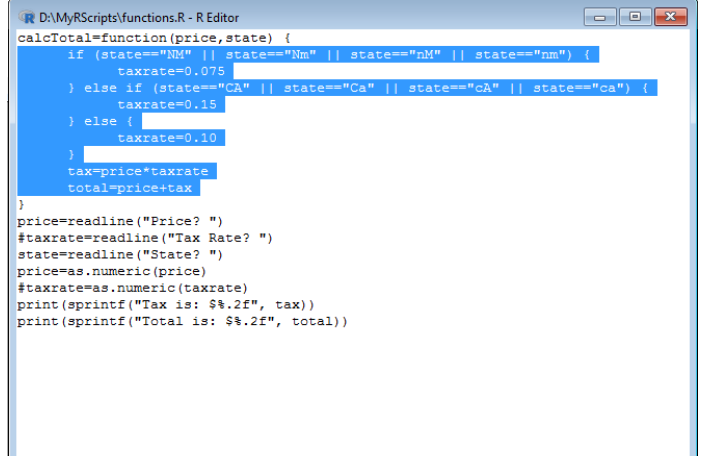


R Programming Fundamentals for Business Students— Functions

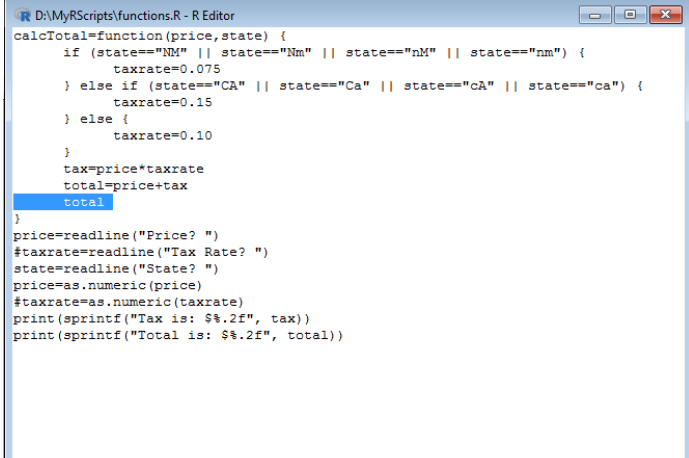
Nick V. Flor, University of New Mexico (nickflor@unm.edu)

ACTION	REACTION
<ul style="list-style-type: none"> Startup R (not shown) File > Change dir... to your workspace from the previous tutorial (not shown) File > Open script... (or ctrl-O) salescalc2.R File > Save as... functions.R <p>In the previous tutorial we explored loops. Next, we'll look at functions by modifying salescalc2.R</p>	
VARIABLE AS FUNCTION NAME, FUNCTION KEYWORD, PARAMETERS, BODY	
<ul style="list-style-type: none"> Enter the following code at the top of the file: <pre>calcTotal=function(price, state) { }</pre> <p>Like functions in most languages, you have some a keyword <i>function</i> to denote that what follows is a function, and the function has parameters (<i>price</i> and <i>state</i>, in this example. Of course these labels are up to you and you could have used p & s).</p> <p>Unlike functions in many languages, you must assign the function to a variable (in this example <i>calcTotal</i>)</p>	
<ul style="list-style-type: none"> Cut the lines of code starting with the if..., up to and including total=..., and paste this code inside the function (between the braces) <p>Everything between the braces is known as the body of the function, and it's usually processing code.</p> <p>Note: The reason why I named the function parameters price and state, in the previous step, is so I wouldn't have to rename them after I pasted the code. If I labelled them p and s, after pasting I would have to rename all instances of price and state to p and s, respectively</p>	

RETURN VALUE:

- Add the variable **total** as the last line in the body of the function

Unlike programming languages like javascript, C#, or java, where you return an object using **return** *objectname*, simply typing the variable name of the object returns its value in R



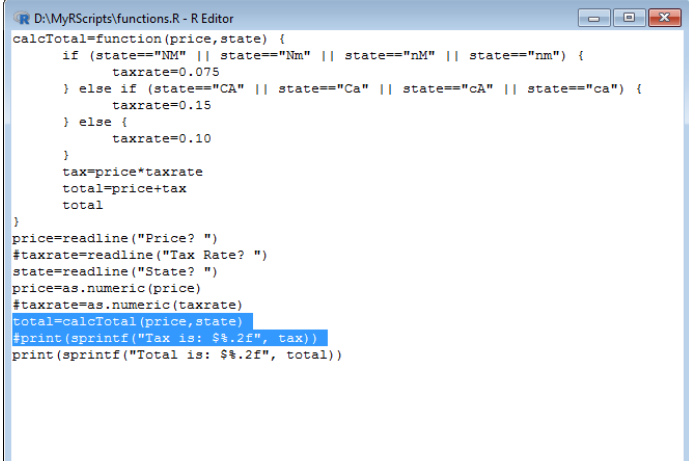
```
DA\MyRScripts\functions.R - R Editor
calcTotal=function(price,state) {
  if (state=="NM" || state=="Nm" || state=="nM" || state=="nm") {
    taxrate=0.075
  } else if (state=="CA" || state=="Ca" || state=="cA" || state=="ca") {
    taxrate=0.15
  } else {
    taxrate=0.10
  }
  tax=price*taxrate
  total=price+tax
  total
}
price=readline("Price? ")
#taxrate=readline("Tax Rate? ")
state=readline("State? ")
price=as.numeric(price)
#taxrate=as.numeric(taxrate)
print(sprintf("Tax is: $%.2f", tax))
print(sprintf("Total is: $%.2f", total))
```

CALLING THE FUNCTION

- Call the function with the user entered *price* and *state* as parameters, assigning it to *total*
- Comment out the printing of tax

Note 1: The function call goes where we cut the code, which was after we retrieved all user input, and before we print any output.

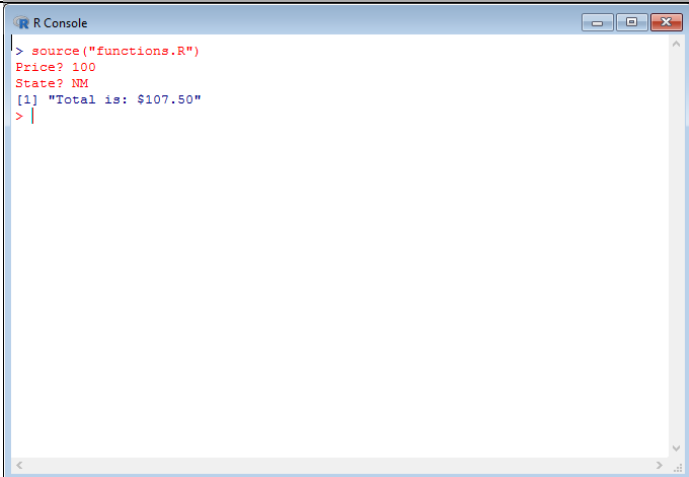
Note 2: The reason we commented out the printing of tax, is because our function currently only returns total. We'll fix this after we test that total is correct.



```
DA\MyRScripts\functions.R - R Editor
calcTotal=function(price,state) {
  if (state=="NM" || state=="Nm" || state=="nM" || state=="nm") {
    taxrate=0.075
  } else if (state=="CA" || state=="Ca" || state=="cA" || state=="ca") {
    taxrate=0.15
  } else {
    taxrate=0.10
  }
  tax=price*taxrate
  total=price+tax
  total
}
price=readline("Price? ")
#taxrate=readline("Tax Rate? ")
state=readline("State? ")
price=as.numeric(price)
#taxrate=as.numeric(taxrate)
total=calcTotal(price,state)
#print(sprintf("Tax is: $%.2f", tax))
print(sprintf("Total is: $%.2f", total))
```

- File > Save
- Enter: source("functions.R") in the R Console
- Enter 100 as the price and NM as the state

The total is returned correctly. But what about tax? How do we return two values from a single function call? The answer is we can combine the total and tax into a vector using the built-in function **c**. The **c** function will combine any number of values N, into a vector, which you can index using 1..N



```
R Console
> source("functions.R")
Price? 100
State? NM
[1] "Total is: $107.50"
> |
```

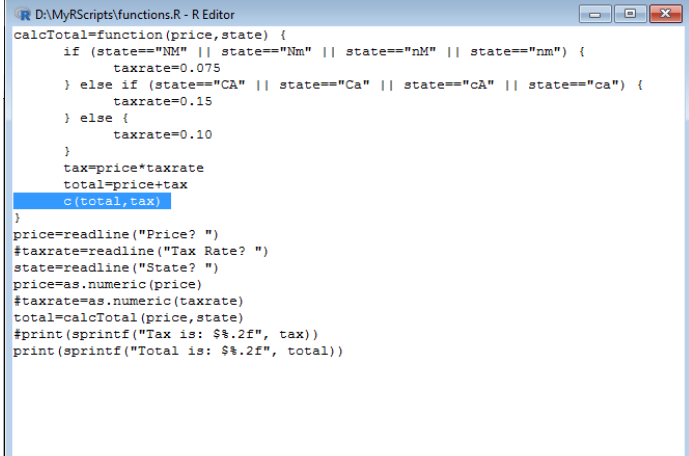
RETURNING MULTIPLE VALUES USING C() AND ACCESSING SPECIFIC VALUES

- Modify the function return value as follows

```
...  
  c(total, tax)  
}
```

As stated, the built-in function `c` combines its parameters into a vector. Total is in the first location, and value is in the second location of the vector.

You can access total using `vectorname[1]` and tax using `vectorname[2]`, as follows:



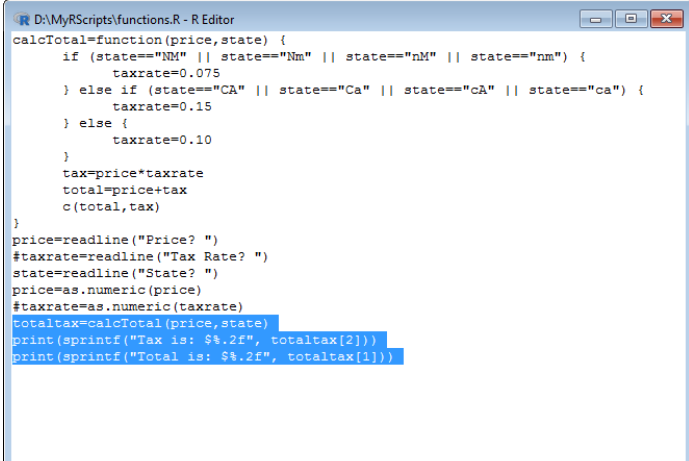
```
D:\MyRScripts\functions.R - R Editor  
calcTotal=function(price,state) {  
  if (state=="NM" || state=="Nm" || state=="nM" || state=="nm") {  
    taxrate=0.075  
  } else if (state=="CA" || state=="Ca" || state=="cA" || state=="ca") {  
    taxrate=0.15  
  } else {  
    taxrate=0.10  
  }  
  tax=price*taxrate  
  total=price+tax  
  c(total, tax)  
}  
price=readline("Price? ")  
#taxrate=readline("Tax Rate? ")  
state=readline("State? ")  
price=as.numeric(price)  
#taxrate=as.numeric(taxrate)  
total=calcTotal(price,state)  
#print(sprintf("Tax is: $%.2f", tax))  
print(sprintf("Total is: $%.2f", total))
```

- Rename `total=...` to `totaltax=...`
- Uncomment the print statement for tax
- Replace `tax` with `totaltax[2]`
- Replace `total` with `totaltax[1]`

Note 1: `totaltax` is the vectorname

Note 2: `totaltax[1]` contains the total computed in the `calcTotal` function

Note 3: `totaltax[2]` contains the tax computed in the `calcTotal` function

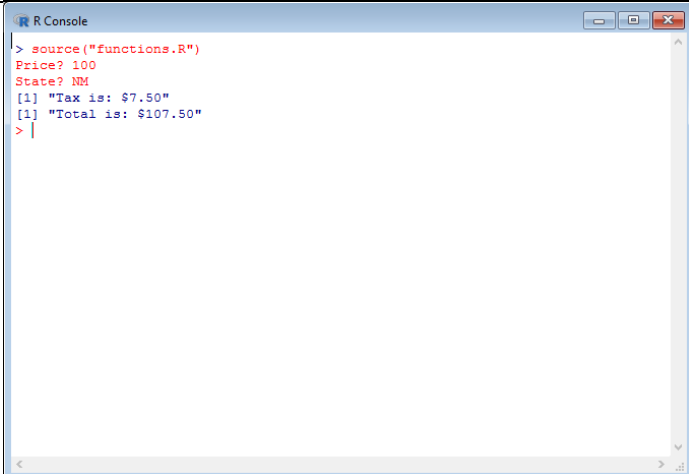


```
D:\MyRScripts\functions.R - R Editor  
calcTotal=function(price,state) {  
  if (state=="NM" || state=="Nm" || state=="nM" || state=="nm") {  
    taxrate=0.075  
  } else if (state=="CA" || state=="Ca" || state=="cA" || state=="ca") {  
    taxrate=0.15  
  } else {  
    taxrate=0.10  
  }  
  tax=price*taxrate  
  total=price+tax  
  c(total, tax)  
}  
price=readline("Price? ")  
#taxrate=readline("Tax Rate? ")  
state=readline("State? ")  
price=as.numeric(price)  
#taxrate=as.numeric(taxrate)  
totaltax=calcTotal(price,state)  
print(sprintf("Tax is: $%.2f", totaltax[2]))  
print(sprintf("Total is: $%.2f", totaltax[1]))
```

- File > Save (or Ctrl-S)
- Enter: `source("functions.R")` in the R Console
- Enter 100 for the price and NM for the state

R calculates the correct Tax and Total!

So that's how functions work in a very brief nutshell. I hope that's enough to leverage your understanding of other functions, if not, read through the R manuals and try creating your own functions.



```
R Console  
> source("functions.R")  
Price? 100  
State? NM  
[1] "Tax is: $7.50"  
[1] "Total is: $107.50"  
> |
```