Creating R Functions

Heide Jackson
heidej@umd.edu

University of Maryland Population Research Center

August 2019

High Level Things to Know

- Custom functions can streamline R coding and group together related tasks.
- Functions are available in base R, via packages added, and can be readily created.

The Structure of a Function

- All functions can be defined with a common structure

```
#function name
myfirstfunction<-
#function denotes we are defining a function
function
#parantheses contain objects defined by function
(){
#within brackets give what the function does
print("Hello")
}
```

Viewing and Calling the Function

- To view what is in the user created function, we can just type myfirstfunction.
- Running the function is similar type myfirstfunction()

Useful Functions within a Function

- paste()–combines text or objects together.
- assign()–assigns function, text, or something else to an object.
- if(), else(), and print–if/else statements in combination with print can show warning messages or document function functioning.

An Example of a Function in Action

Adding a Prefix in a Data Frame

```
nnames<-paste(prefix,tnames,sep="")
colnames(data)<-nnames
print(paste("All Variables Renamed with Prefix", prefix,
sep=" "))
return(data)
library("gapminder")
addprefix(gapminder, "n")
```

Other Useful Things to Know

- Objects defined for a function exist only in the context of that function.
- When calling a function, R matches objects to function names based on position.
- Source function can be used to load in functions stored in other scripts.