

7.4 Functions

Writing functions in R is very easy. Whenever you want to run certain code over and over again, but don't want to source or paste things into the console repeatedly, you can and should create your own function with the `function()`. Functions take objects as arguments and return another object as output. Consider the following function:

```
1 > MyFunction <- function(Object){
  Object + Object
}

2 > MyFunction
function(Object){
  Object + Object
}

3 > MyFunction(Object = 17)
[1] 34
```

You can wrap all sorts of functions, loops, and statements inside of a function to simplify repetitive tasks. You can specify as many parameters as you like. Let's make a function that makes a basic scatterplot. Note: the `require()` function will automatically load the `ggplot` package if needed.

```
1 > Scatterplot <- function(X, Y, Title, X.Axis, Y.Axis) {
  require(ggplot2)
  temp <- data.frame(X, Y)
  limits.X <- c(min(X) - 0.25*sd(X), max(X) + 0.25*sd(X))
  limits.Y <- c(min(Y) - 0.25*sd(Y), max(Y) + 0.25*sd(Y))
  ggplot(data = temp) +
    geom_point(aes(x = X, y = Y), alpha = 0.3) +
    labs(title = Title, x = X.Axis, y = Y.Axis) +
    coord_cartesian(xlim = limits.X , ylim = limits.Y)
}
>
```

Let's take our function for a test drive. We can use the fuel economy data.

```
1 > Var1 <- FE2013$FEhighway
2 > Var2 <- FE2013$FEcity

3 > Scatterplot(X = Var1, Y = Var2, Title="Fuel Economy", X.
  Axis = "Highway", Y.Axis = "City")
>
```

