7.2 For Loops

Another control flow operator is the `for()` operator. For loops are the workhorses of R programming. They are extremely intuitive and straightforward to write and wrap your head around. Unfortunately, they also tend to be very inefficient. With modern computing power this may not be a problem for simple tasks. But you can bog your computer down with these if you are not careful.[3] The easiest way to understand how to use the `for()` operator is by example:

```
> for(i in 1:10) {print(i)}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

In the above example we asked R that for each i in the sequence from 1 and 10 we want that element printed to the screen. Or put differently, for each i in some object do the thing inside the curly braces.

Try this:

```
> X <- 1000000:1
> for(i in X) {print(i)}
>
```

To stop this silly loop hit the `Esc` button or press `Ctrl+C`. Now how about this:

Warning: This example may induce SEIZURES!

```
> X <- 10000:1
> for(i in X) {
    temp <- rnorm(n = 100, mean = 0, sd = 2)
    plot <- ggplot() + geom_density(aes(temp), fill = i) +
      labs(title = paste("i =", i))
    print(plot)
  }
>
```

[3] The often unintuitive alternative to for loops is vectorization via the `apply()` and `lapply()` functions. Feel free to google it.

7.2.1 Applications with Data

Let's actually use for loops to do something useful. We start by loading the Fuel Economy dataset and write a simple loop to start with.

```
1 > FE2013 <- read.csv("http://peterhaschke.com/Teaching/R-
    Course/FE2013.csv")
2 > levels(FE2013$Manufacturer) {
  [1] "Audi"           "Bentley"
  [3] "BMW"            "Bugatti"
  [5] "Chrysler"       "Ferrari"
  [7] "Ford"           "General Motors"
  [9] "Honda"          "Hyundai"
 [11] "Jaguar"         "Kia"
 [13] "Lamborghini"    "Land Rover"
 [15] "Lotus"          "Maserati"
 [17] "MAZDA"          "Mercedes-Benz"
 [19] "Mitsubishi"     "Nissan"
 [21] "Porsche"        "Rolls-Royce"
 [23] "Roush"          "Subaru"
 [25] "Suzuki"         "Toyota"
 [27] "Volkswagen"     "Volvo"

3 > for(i in levels(FE2013$Manufacturer)){{
      print(i)
    }
 [1] "Audi"
 [1] "Bentley"
 [1] "BMW"
 [1] "Bugatti"
 [1] "Chrysler"
 [1] "Ferrari"
 [1] "Ford"
 [1] "General Motors"
 [1] "Honda"
 [1] "Hyundai"
 [1] "Jaguar"
 [1] "Kia"
 [1] "Lamborghini"
 [1] "Land Rover"
 [1] "Lotus"
 [1] "Maserati"
 [1] "MAZDA"
 [1] "Mercedes-Benz"
 [1] "Mitsubishi"
 [1] "Nissan"
 [1] "Porsche"
 [1] "Rolls-Royce"
```

```
[1] "Roush"
[1] "Subaru"
[1] "Suzuki"
[1] "Toyota"
[1] "Volkswagen"
[1] "Volvo"
```

Nice. With the above loop we were able to print out each level of the variable manufacturer. Knowing this we can start adding some useful features to the loop.

```
> for(i in levels(FE2013$Manufacturer)){
    temp <- subset(FE2013, FE2013$Manufacturer==i)
    mean.mpg <- round(mean(temp$FEcombined))
    cat(mean.mpg, "mpg for", i,"\n")
  }
23 mpg for Audi
15 mpg for Bentley
23 mpg for BMW
10 mpg for Bugatti
22 mpg for Chrysler
14 mpg for Ferrari
22 mpg for Ford
20 mpg for General Motors
26 mpg for Honda
26 mpg for Hyundai
19 mpg for Jaguar
26 mpg for Kia
15 mpg for Lamborghini
16 mpg for Land Rover
21 mpg for Lotus
15 mpg for Maserati
26 mpg for MAZDA
20 mpg for Mercedes-Benz
24 mpg for Mitsubishi
22 mpg for Nissan
21 mpg for Porsche
14 mpg for Rolls-Royce
17 mpg for Roush
24 mpg for Subaru
25 mpg for Suzuki
23 mpg for Toyota
27 mpg for Volkswagen
21 mpg for Volvo
```

In the for loop above we did the following things for each i in `levels(FE2013$Manufacturer)` (i.e. for each car manufacturer):

1. For each manufacturer i, we subset our dataset such that it only contains observations for i. For each i we saved this subset of the dataset to the object `temp`.
2. After the subsetting, we compute the rounded mean of the combined fuel economy for the subset and store it in the object called `mean.mpg`.
3. After each loop we tell **R** to concatenate (`cat()`) the mean.mpg to the i^{th} manufacturer.

Suppose we actually wanted to save the output the loop generated instead of just printing it to the screen. The easiest way to do this is to create a matrix populated by `NA`'s which we can then populate it with the data the loop generates.

```
> Data <- matrix(NA, nrow = length(levels(FE2013$Manufacturer
   )), ncol = 4)
> rownames(Data) <- as.character(levels(FE2013$Manufacturer))
> colnames(Data) <- c("meanFE", "sdFE", "medianRating", "N")
> Data
                MeanFE   sdFE   medianRating      N
Audi                NA     NA             NA     NA
Bentley             NA     NA             NA     NA
BMW                 NA     NA             NA     NA
Bugatti             NA     NA             NA     NA
Chrysler            NA     NA             NA     NA
Ferrari             NA     NA             NA     NA
Ford                NA     NA             NA     NA
General Motors      NA     NA             NA     NA
Honda               NA     NA             NA     NA
Hyundai             NA     NA             NA     NA
Jaguar              NA     NA             NA     NA
Kia                 NA     NA             NA     NA
Lamborghini         NA     NA             NA     NA
Land Rover          NA     NA             NA     NA
Lotus               NA     NA             NA     NA
Maserati            NA     NA             NA     NA
MAZDA               NA     NA             NA     NA
Mercedes-Benz       NA     NA             NA     NA
Mitsubishi          NA     NA             NA     NA
Nissan              NA     NA             NA     NA
Porsche             NA     NA             NA     NA
Rolls-Royce         NA     NA             NA     NA
Subaru              NA     NA             NA     NA
Suzuki              NA     NA             NA     NA
Toyota              NA     NA             NA     NA
Volkswagen          NA     NA             NA     NA
Volvo               NA     NA             NA     NA
```

Now we can use a for loop to populate the matrix.

```
> for(i in levels(FE2013$Manufacturer)){
    temp <- subset(FE2013, FE2013$Manufacturer==i)
    Data[i,1] <- round(mean(temp$FEcombined), digits = 2)
    Data[i,2] <- round(sd(temp$FEcombined), digits = 2)
    Data[i,3] <- median(temp$FErating)
    Data[i,4] <- nrow(temp)
  }
> Data
                MeanFE     sdFE    medianRating      N
Audi             22.66     3.15              6.0     43
Bentley          14.85     1.77              2.0      9
BMW              23.37     4.84              6.0    139
Bugatti          10.44       NA              1.0      1
Chrysler         21.55     4.50              5.0     82
Ferrari          14.36     1.02              2.5      8
Ford             21.90     6.96              5.0     94
General Motors   19.78     5.25              4.0    177
Honda            25.83     6.17              6.0     31
Hyundai          25.98     4.39              6.0     34
Jaguar           18.65     1.23              4.0     13
Kia              25.90     3.06              7.0     34
Lamborghini      14.67     1.09              3.0      5
Land Rover       16.24     4.21              2.5      4
Lotus            21.46     1.03              5.0      4
Maserati         15.03     0.80              3.0      3
MAZDA            26.07     4.14              6.0     24
Mercedes-Benz    20.41     4.36              5.0     77
Mitsubishi       24.29     3.09              6.0     17
Nissan           21.80     5.47              5.0     57
Porsche          20.83     2.24              5.0     41
Rolls-Royce      14.34     0.74              2.0      6
Roush            17.42     1.07              4.0      2
Subaru           24.02     3.80              6.0     22
Suzuki           24.64     2.01              6.0     16
Toyota           23.15     7.62              5.0     77
Volkswagen       27.00     5.18              6.5     46
Volvo            21.27     1.95              5.0     16
```

7.2.2 Putting the Pieces Together

Although the code below may look complicated, most of it should be straightforward to interpret. Nothing you haven't seen before:

```
> library(ggplot2)
> FE2013$Gears <- as.factor(FE2013$Gears)
> MAKE<-as.character(levels(FE2013$Manufacturer))
> LIST <- as.list(rep(NA, length(MAKE)))
> names(LIST) <- MAKE

> for(i in levels(FE2013$Manufacturer)){
    temp <- subset(FE2013 , FE2013$Manufacturer==i)
    LIST[[i]] <- ggplot(data = temp, aes(x = FEcity, y =
       FEhighway)) +
    geom_point(aes(color = Gears)) +
    labs(title = paste("Manufacturer:",i), x = "Fuel Economy:
        City", y = "Fuel Economy: Highway ") +
    facet_wrap(~ Division) +
    if(nrow(temp) > 2 & nrow(temp) < 50) {
      geom_smooth(method = "lm")} else {
        if(nrow(temp) >= 50) {
          geom_smooth(method = "loess", span = 2 )}
    }
    pdf(file = paste("z:/", i, ".pdf", sep = ""), width=6,
       height=5)
    print(LIST[[i]])
    dev.off()
}
```

7.3 Other Loops

There are a few other types of loops and control flow operators. The `repeat` operator, simply repeats everything after it until you tell it to stop. It will loop until the lights go out. Like so:

```
> Number <- 1
> repeat{Number <- Number + 1; print(Number)}
>
```