# Homework #1: Looping Galore

*PPOL 670*

Data is often times stored in smaller chunks in order to ease searchability and discoverability of information, often doing so at the smallest time, geographic, or reporting unit possible. Weather radar, for example, will produce volumetric scans of the atmosphere and store the data in five-minute intervals that correspond to the amount of time it takes to produce a scan. Economic data is often times reported on a monthly basis as it takes time to collect the data. Newly enacted laws in the United States are named and stored based on the meeting of the Congress (e.g. the 104th Congress) and the number of laws that were approved in that Congress (e.g. the Patriot Act is 107-56).

But what if the data needed to be strung together to be able to tell a greater story?

In this homework assignment, you work with US Census Bureau's Monthly Building Permit Survey by Metropolitan Area. Using what you've learned thus far in class, you assemble a dataset of monthly data that spans from January 2000 to December of 2015 – 204 periods/files. Data is stored on the Census Bureau's FTP – http://www2.census.gov/econ/bps/Metro/. On the FTP, there are multiple data series. Use only data series that match the following naming convention: `maYYDDc.txt`. For example, the data for January 2008 would be `ma0801c.txt` and July 2011 would be `ma1107c.txt`. Do not use the data that ends in `y.txt` as that is an annual cumulative estimate.

## Tasks

1. Write a 'padding' function that accepts a numeric vector and returns a vector of characters in which each input number can occupy a max of two characters and single digit numbers have a left-padded zero value . For example, passing the vector `c(1, 3, 10)` will be returned as `c("01", "03", "10")`.
2. Using `seq()`, `expand.grid()`, `paste0()` and your new padding function, create a string vector that contains all dates from Jan. 2000 to Dec. 2015. The vector should be in the form `YYMM`. For example, Jan. 2013 is represented as `"1301"` and Dec. 2007 is `"0712"`.
3. Using a for-loop, download all monthly files from Jan. 2000 to Dec. 2015 from the Census Monthly Building Permit survey FTP, keeping only columns of interest, namely 1, 5 and 6. Append each monthly file to a master file. An example link for Jan. 2000 looks like the following: http://www2.census.gov/econ/bps/Metro/ma0001c.txt. A combination of the following functions will be useful: `for()`, `rbind()`, `download.file()`, and `tempfile()`. For `download.file()`, use `mode = "wb"`.
4. Drop any row with a `NA` value. Calculate the mean monthly volume of housing permits for each MSA. Name the MSA ID column as "msa" and the mean monthly volume as "mean". Save this table as a CSV, following this naming convention "firstname-lastname-hmwk1.csv". For example: Dan Hammer would be "dan-hammer-hmwk1.csv".

Turn in your code as a functional `.R` file and the CSV of your results.

## How you will be graded

- 40 pts. Adherence to the Google Script Style Guide, elegance and functionality of the code (e.g. do not manually copy and pasting names of files from browser, code needs to be readable)
- 60 pts. Accuracy relative to the master file as calculated by professors.

## Answer Key

1. Write padding function. Two versions are presented. One with support up to two characters, and one with any number of characters

```r
#Create code to pad zeroes
  lead0 <- function(m, pad_val){
      # A function to pad values with a specified characters
      # Only designed to handle up to two characters
      #
      # Args:
      #       m: a vector
      #       pad_val: the padding character (e.g. "0")
      #
      # Returns:
      #       A vector where all elements have two characters
      #
    m[nchar(m)==1] <- paste0(pad_val,m[nchar(m)==1])
    return(m)
  }

#An alternative version for any length
   lead0a <- function(m,pad_val, pad_len){
      # A function to pad values with a specified characters
      #
      # Args:
      #       m: a vector
      #       pad_val: the padding character (e.g. "0")
      #       pad_len: number of characters
      #
      # Returns:
      #       A vector with all elements have the specified number of characters
      #       where elements with less than specified length are padded
      m <- as.character(m)
      for(k in 1:(pad_len-1)){
        m[nchar(m)==k] <- paste0(paste(rep(pad_val,pad_len - k), collapse = ""), m[nchar(m)==k])
      }
      return(m)
    }
```

2. Create vector of periods using `lead0` function. Use `expand.grid` to get all combinations of months and years.

```r
#Prep time codes to be looped
  month <- lead0(as.character(seq(1,12,1)),"0")
  year <- lead0(as.character(seq(0,15,1)),"0")
  comb <- expand.grid(year,month)
  period <- paste0(comb[,1],comb[,2])
  period <- sort(period)
```

3. Write function to download all periods

```r
#Write download function
  permits <- function(vec){
      # A function to retrieve MSA housing permits for periods specified in YYMM form
      #
      # Args:
      #       vec: a vector of dates in YYMM form
      #
      # Returns:
      #       A data frame with all housing permit records for dates in vec
      #
    master <- data.frame()

    for(k in vec){
      #Create a new temp file
      temp <- tempfile()

      #Download file to temp
      download.file(paste0("http://www2.census.gov/econ/bps/Metro/ma",k,"c.txt"),
                    temp, mode="wb")

      #Read into data frame
      df <- read.csv(temp, skip=1)

      # Note that while in the homework cols 1, 5 and 6 were requested, 3 is saved here for grading pur
      df <-  df[,c(1,3,5,6)]

      # Rename columns
      colnames(df)<-c("date","msa.code","msa.name","bldgs")

      #Append to master
      master <- rbind(master,df)
    }
    return(master)
  }

# Run the function of the period vector
  df <- permits(period)
  head(df, 3)
```

```
##     date msa.code         msa.name bldgs
## 1     NA       NA                     NA
## 2 200001     9999 Abilene  TX MSA     16
## 3 200001     9999  Albany  GA MSA     42
```

4. Calculate answer key

```
# Keep non-blank records
  short <- df[!is.na(df),]

# Aggregate data
  result <- aggregate(short$bldgs,
                      by = list(msa.name = short$msa.name, msa.code = short$msa.code),
                      FUN = mean)

#Rename columns
  colnames(result) <- c("msa.name","msa.code","mean")

#Save data as CSV
  write.csv(result, "answer_key.csv", row.names=F)
```