

R, FRED, and the 2016 Texas Primary: Part 2

Luke Smith

April 26, 2016

Overview: Picking up from last time

I have all the metadata and the data itself. However, all that data is floating around in data frames inside a list. I need a way to clean this up to make operations on the data frames more manageable.

Introduction to `cat.functions.R`

The file `cat.functions.R` contains a functions that we will use to tidy up the FRED data:

- `cat.info()` = creates a master/top-level table
 - `obs.catcher()` = uses 'category' information from `cat.info()` to group data frames in `fred.obs` by 'category'
 - `cat.tabler()` = creates tables each 'category' list element returned by `obs.catcher()`
-

Why create and use `cat.functions.R`?

1. The first function, `cat.info()`, will create a handy table with information about each category
 2. We can use this table to check the frequency and range of each category
 3. The second function, `obs.catcher()`, will categorize the data in `fred.obs`
 4. The last function in the list, `cat.tabler()`, will create a table for each category (I initially thought this would be useful but since have reconsidered, but I am including it anyway)
-

Function overview: `cat.info()`

This will create a master/top-level table from which to draw certain metadata which will be used in other functions.

```
cat.info <- function(series.table){  
  
### load dependencies  
  require(plyr, quietly = T) ## I have a habit of loading plyr BEFORE dplyr  
  require(dplyr, quietly = T)
```

```
### make an aggregate table based on $Category
summarized.series <- fred.series %>% select(2,3,6,7,8,9) %>%
  aggregate(list(series.table$Category), unique) %>%
  select(3,2,4:7) %>% arrange(Release, Category)
}
```

Function overview: obs.catcher()

This will categorize the data frames in the fred.obs list. With this categorized list, we will be able to aggregate data as needed for our analysis (i.e. summarizing monthly data into annual data, etc).

```
obs.catcher <- function(series.table, obs.list, cat){

### subset by category
cat.series <- series.table[series.table$Category == cat, ]

### Find all SeriesID matching the subset's SeriesID
cat.obs <- list()
cat.obs <- obs.list[cat.series$SeriesID]

}
```

Function overview: cat.tabler()

This will create a table from each category in the fred.obs list. The first column is Date and has a further 254 columns (one for each county). The 254 columns have column headers that correspond to the SeriesID.

```
cat.tabler <- function(obs.list){
  x <- 1
  while(x <= length(obs.list)){
    if(x == 1){
      main.frame <- data.frame() # initialize the data frame
      main.frame <- obs.list[[x]] # add the first list object
      names(main.frame)[2] <- names(obs.list)[1] # give that object a name
      main.frame
    } else {
      main.frame <- merge(main.frame, obs.list[[x]], by = "Date")
      names(main.frame)[x + 1] <- names(obs.list)[x]
      main.frame
    }
    x = x + 1
  }
  main.frame
}
```

```
}  
  main.frame  
}
```

Running the functions: Plan of attack

Here is our plan of attack for using these functions to tidy our data:

1. Create a top-level/master table using `cat.info()`
 2. Use the result table from `cat.info()` to group the data frames in `fred.obs` by category: we will use `obs.catcher()` to do this
 3. Create separate tables of each category: each table will correspond to one of the categories in `cat.info()`, and each table will have 255 columns (one called Date and the others will have names corresponding to the SeriesID of each of the 254 counties in Texas)
-

Running the functions: Loading the dependencies

```
library(plyr, quietly = T)  
library(dplyr, quietly = T)  
library(rvest, quietly = T)  
library(lubridate, quietly = T)  
source("cat.functions.R")  
load("Data/fred.series")  
load("Data/fred.obs")
```

Running the functions: `cat.info()`, `obs.catcher()`, and `cat.tabler()`

Create a top-level/master table using `cat.info()`

```
fred.master <- cat.info(fred.series)
```

Group the data frames in `fred.obs` by category using `obs.catcher()`

```
fred.cat.list <- lapply(seq_along(fred.master$category), function(x){
  obs.catcher(series.table = fred.series,
               obs.list    = fred.obs,
               cat         = fred.master$category[x])
})
names(fred.cat.list) <- fred.master$category
save(fred.cat.list, file = "Data/fred.cat.list.RData")
```

Create separate tables of each category using `cat.tabler()`

```
fred.tables <- lapply(seq_along(fred.cat.list), function(x){
  cat.tabler(fred.cat.list[[x]])
})
names(fred.tables) <- fred.master$category
# fred.tables <- fred.tables %>% select(-1) %>% as.character() %>% as.numeric()
save(fred.tables, file = "Data/fred.tables.RData")
```

Results

Let's take a look at what we have.

```
# `cat.info()`
head(fred.master)
```

```
##              category release  freq  start
## 1      Civilian Labor Force   116 Monthly 1990-01-01
## 2      Unemployment Rate     116 Monthly 1990-01-01
## 3      Resident Population   119  Annual 1970-01-01
## 4      Per Capita Personal Income 175  Annual 1969-01-01
## 5      Personal Income      175  Annual 1969-01-01
## 6 Bachelor's Degree or Higher (5-year estimate) 330 5-Years 2010-01-01
##          end          units
## 1 2016-01-01      Persons
## 2 2016-01-01      Percent
## 3 2014-01-01 Thousands of Persons
## 4 2013-01-01      Dollars
## 5 2013-01-01 Thousands of Dollars
## 6 2012-01-01      Percent
```

```
# `obs.catcher()`
tail(fred.cat.list$`Resident Population`$TXHARR1POP)
```

```
##           Date      Value
## 40 2009-01-01 4074.423
## 41 2010-01-01 4108.909
## 42 2011-01-01 4181.948
## 43 2012-01-01 4263.060
## 44 2013-01-01 4352.752
## 45 2014-01-01 4441.370
```

```
# `cat.tabler()`
head(fred.tables$`Civilian Labor Force`[1:5])
```

```
##           Date TXANDE1LFN TXANDR3LFN TXANGE5LFN TXARAN7LFN
## 1 1990-01-01      17854      6356      32084      7444
## 2 1990-02-01      17768      6219      32244      7394
## 3 1990-03-01      17844      6296      32451      7390
## 4 1990-04-01      17818      6230      32197      7373
## 5 1990-05-01      17999      6290      32306      7486
## 6 1990-06-01      18010      6211      32600      7629
```