dplyr Example 1 - BLS Wide to Tidy

Jim Porzak

2016-02-17

This example shows how to solve a common data prep problem - how to convert a *wide* table to a *tidy* narrow table. The wide data form is common in spread sheets - especially those representing time series.

The Bureau of Labor Statistics (the BLS) data page, has many data sets and many ways to access them. For a research project we needed the monthly employment figures by â€œmetro areas.â€ Using the *Multi-screen Data Search* tool for *Employment, Hours, and Earnings - State and Metro Area* we captured the non-farm employment from 1995 through 2015 as a tab seperated file. See the BLS *sm* file spec

There is a bit clean-up to do to make it readable with `read_tsv()` from Hadleyâ€™s readr package. See the function `BLS_CleanRawTextFile()` included in this package if you are interested in the details (which are not relevant for this dplyr example).

Read the BLS file into the data frame `employment` and take a glimpse of the first and last ten columns

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(readr)
library(dplyrExamples)
library(stringr)
library(ggplot2)
library(lubridate)
fn <- system.file("extdata", "BLS_NonFarmEmploymentInAreas_1995_2015.tsv",
                  package = "dplyrExamples")
file_out <- "file_out.tsv"
BLS_CleanRawTextFile(fn, file_out)  ## Clean up the BLS raw text file
employment <- read_tsv(file_out)
dim(employment)
```

```
## [1] 436 253
```

```
glimpse(employment[1:10])
```

```
## Observations: 436
## Variables: 10
## $ Series_ID (chr) "SMU01115000000000001", "SMU01122200000000001", "SMU...
## $ Jan_1995  (dbl) 47.5, 40.2, 466.4, 38.7, 52.7, 56.3, 56.6, 37.8, 162...
## $ Feb_1995  (dbl) 47.5, 40.8, 467.0, 39.1, 52.9, 57.0, 57.0, 37.9, 163...
## $ Mar_1995  (dbl) 48.0, 40.6, 470.8, 40.2, 53.6, 57.4, 57.6, 37.9, 164...
## $ Apr_1995  (dbl) 48.3, 41.0, 471.9, 41.8, 53.8, 57.7, 57.7, 38.1, 165...
## $ May_1995  (dbl) 48.1, 41.4, 472.7, 42.4, 53.6, 57.4, 58.2, 38.4, 166...
## $ Jun_1995  (dbl) 48.4, 41.0, 476.4, 44.0, 54.4, 57.9, 58.6, 38.4, 166...
## $ Jul_1995  (dbl) 48.1, 40.8, 474.4, 43.7, 53.4, 56.3, 58.0, 38.5, 163...
## $ Aug_1995  (dbl) 47.9, 40.7, 473.5, 43.3, 53.3, 57.6, 58.2, 38.3, 165...
## $ Sep_1995  (dbl) 48.1, 41.1, 477.2, 42.7, 53.9, 58.4, 58.7, 38.7, 167...
```

```
glimpse(employment[244:253])
```

```
## Observations: 436
## Variables: 10
## $ Mar_2015 (dbl) 46.3, 60.5, 516.2, 68.5, 53.8, 57.5, 55.2, 37.2, 217....
## $ Apr_2015 (dbl) 46.6, 60.9, 517.5, 69.3, 53.5, 57.7, 55.5, 37.4, 218....
## $ May_2015 (dbl) 46.5, 60.9, 520.2, 70.7, 54.0, 57.2, 55.5, 37.8, 220....
## $ Jun_2015 (dbl) 46.7, 60.7, 521.7, 72.5, 54.3, 57.4, 55.2, 38.0, 218....
## $ Jul_2015 (dbl) 46.3, 60.5, 518.6, 71.4, 53.7, 57.2, 54.8, 37.4, 218....
## $ Aug_2015 (dbl) 46.4, 61.2, 519.7, 70.4, 53.6, 57.1, 55.1, 37.4, 216....
## $ Sep_2015 (dbl) 46.0, 62.1, 519.1, 70.0, 53.9, 57.2, 55.4, 37.4, 219....
## $ Oct_2015 (dbl) 46.3, 61.7, 520.9, 69.3, 53.8, 57.5, 55.7, 37.6, 219....
## $ Nov_2015 (dbl) 46.5, 62.2, 525.4, 69.2, 54.3, 57.8, 56.3, 38.0, 221....
## $ Dec_2015 (dbl) 46.6, 61.8, 524.4, 69.0, 54.4, 58.0, 56.1, 37.7, 221....
```

There are two challenges with this data set:

1. The *Series_ID* needs to be decoded to pull out the state and area for the row.
2. It is *very* wide - 253 columns! We wish to tidy it up, in the Hadley sense, so each row will just a single numeric column, the number of non-farm employees in the area for the month.

The series_id decoder is in Section 5 of the BLS SM file spec and is repeated here:

```
0        1         2    <--
12345678901234567890    <-- character counter
SMU01266207072200001    <-- Sample Series ID

Positions Code                          Value

1-2       survey abbreviation   =       SM
```

```
3         seasonal (code)       =       U
4-5       state_code            =       01
6-10      area_code             =       26620
11-12     supersector_code      =       70
13-18     industry_code         =       70722000
19-20     data_type_code        =       01
```

We did not need NIACâ€™s code breakdown so positions 11-18 are all zeros in our extract.

The BLS has standard lookup files for the state and area codes which we load now.

Read State & Area Code Tables and take a glimpse

```
fac <- system.file("extdata", "BLS_AreaCodes.tsv", package = "dplyrExamples")
area_codes <- read_tsv(fac)
fsc <- system.file("extdata", "BLS_StateCodes.tsv", package = "dplyrExamples")
state_codes <- read_tsv(fsc)
glimpse(area_codes)
```

```
## Observations: 444
## Variables: 2
## $ area_code (chr) "00000", "10180", "10380", "10420", "10500", "10540"...
## $ area_name (chr) "Statewide", "Abilene, TX", "Aguadilla-Isabela, PR",...
```

```
glimpse(state_codes)
```

```
## Observations: 53
## Variables: 2
## $ state_code (chr) "01", "02", "04", "05", "06", "08", "09", "10", "11...
## $ state_name (chr) "Alabama", "Alaska", "Arizona", "Arkansas", "Califo...
```

Now we have everything we need to tidy up our data.

Tidy Up Area Employment by Month

This is the dplyr sequence you would use in production. The next section breaks down the process step-by-step.

```
Employment_By_Area_1995_2015 <- employment %>%
  gather(mmm_yyyy, NonFarm_000, -Series_ID) %>%
  mutate(Month_Of = as.Date(paste0("01_", mmm_yyyy), format = "%d_%b_%Y"),
         state_code = str_sub(Series_ID, 4, 5),
         area_code = str_sub(Series_ID, 6, 10)) %>%
  left_join(state_codes) %>%
  left_join(area_codes) %>%
  mutate_each(funs(factor), ends_with("name")) %>%
  rename(State = state_name, Area = area_name) %>%
  select(State, Area, Month_Of, NonFarm_000) %>%
  arrange(State, Area, Month_Of)
```

```
## Joining by: "state_code"
```

```
## Joining by: "area_code"
```

```
glimpse(Employment_By_Area_1995_2015)
```

```
## Observations: 109,872
## Variables: 4
## $ State       (fctr) Alabama, Alabama, Alabama, Alabama, Alabama, Alab...
## $ Area        (fctr) Anniston-Oxford-Jacksonville, AL, Anniston-Oxford...
## $ Month_Of    (date) 1995-01-01, 1995-02-01, 1995-03-01, 1995-04-01, 1...
## $ NonFarm_000 (dbl) 47.5, 47.5, 48.0, 48.3, 48.1, 48.4, 48.1, 47.9, 48...
```

Step-by-step dplyr

Convert from wide to narrow using `tidyr::gather()`.

```
eba <- employment %>%
  gather(mmm_yyyy, NonFarm_000, -Series_ID)
glimpse(eba)
```

```
## Observations: 109,872
## Variables: 3
## $ Series_ID   (chr) "SMU01115000000000001", "SMU01122200000000001", "S...
## $ mmm_yyyy    (chr) "Jan_1995", "Jan_1995", "Jan_1995", "Jan_1995", "J...
## $ NonFarm_000 (dbl) 47.5, 40.2, 466.4, 38.7, 52.7, 56.3, 56.6, 37.8, 1...
```

Convert character `mmm_yyyy` to Date and pull out state & area codes from Series_ID.

```
eba <- eba %>%
mutate(Month_Of = as.Date(paste0("01_", mmm_yyyy), format = "%d_%b_%Y"),
       state_code = str_sub(Series_ID, 4, 5),
       area_code = str_sub(Series_ID, 6, 10))
glimpse(eba)
```

```
## Observations: 109,872
## Variables: 6
## $ Series_ID   (chr) "SMU01115000000000001", "SMU01122200000000001", "S...
## $ mmm_yyyy    (chr) "Jan_1995", "Jan_1995", "Jan_1995", "Jan_1995", "J...
## $ NonFarm_000 (dbl) 47.5, 40.2, 466.4, 38.7, 52.7, 56.3, 56.6, 37.8, 1...
## $ Month_Of    (date) 1995-01-01, 1995-01-01, 1995-01-01, 1995-01-01, 1...
## $ state_code  (chr) "01", "01", "01", "01", "01", "01", "01", "01", "0...
## $ area_code   (chr) "11500", "12220", "13820", "19300", "19460", "2002...
```

Look-up the state and area names from the codes data frames.

```
eba <- eba %>%
  left_join(state_codes) %>%
  left_join(area_codes)
```

```
## Joining by: "state_code"
```

```
## Joining by: "area_code"
```

```
glimpse(eba)
```

```
## Observations: 109,872
## Variables: 8
## $ Series_ID   (chr) "SMU01115000000000001", "SMU01122200000000001", "S...
## $ mmm_yyyy    (chr) "Jan_1995", "Jan_1995", "Jan_1995", "Jan_1995", "J...
## $ NonFarm_000 (dbl) 47.5, 40.2, 466.4, 38.7, 52.7, 56.3, 56.6, 37.8, 1...
## $ Month_Of    (date) 1995-01-01, 1995-01-01, 1995-01-01, 1995-01-01, 1...
## $ state_code  (chr) "01", "01", "01", "01", "01", "01", "01", "01", "0...
## $ area_code   (chr) "11500", "12220", "13820", "19300", "19460", "2002...
## $ state_name  (chr) "Alabama", "Alabama", "Alabama", "Alabama", "Alaba...
## $ area_name   (chr) "Anniston-Oxford-Jacksonville, AL", "Auburn-Opelik...
```

Convert the names to factors and rename them to user friendly names.

```
eba <- eba %>%
  mutate_each(funs(factor), ends_with("name")) %>%
  rename(State = state_name, Area = area_name)
glimpse(eba)
```

```
## Observations: 109,872
## Variables: 8
## $ Series_ID   (chr) "SMU01115000000000001", "SMU01122200000000001", "S...
## $ mmm_yyyy    (chr) "Jan_1995", "Jan_1995", "Jan_1995", "Jan_1995", "J...
## $ NonFarm_000 (dbl) 47.5, 40.2, 466.4, 38.7, 52.7, 56.3, 56.6, 37.8, 1...
## $ Month_Of    (date) 1995-01-01, 1995-01-01, 1995-01-01, 1995-01-01, 1...
## $ state_code  (chr) "01", "01", "01", "01", "01", "01", "01", "01", "0...
## $ area_code   (chr) "11500", "12220", "13820", "19300", "19460", "2002...
## $ State       (fctr) Alabama, Alabama, Alabama, Alabama, Alabama, Alab...
## $ Area        (fctr) Anniston-Oxford-Jacksonville, AL, Auburn-Opelika,...
```

Keep just final columns and sort by month within area within state.

```
eba <- eba %>%
  select(State, Area, Month_Of, NonFarm_000) %>%
  arrange(State, Area, Month_Of)
glimpse(eba)
```

```
## Observations: 109,872
## Variables: 4
## $ State       (fctr) Alabama, Alabama, Alabama, Alabama, Alabama, Alab...
## $ Area        (fctr) Anniston-Oxford-Jacksonville, AL, Anniston-Oxford...
## $ Month_Of    (date) 1995-01-01, 1995-02-01, 1995-03-01, 1995-04-01, 1...
## $ NonFarm_000 (dbl) 47.5, 47.5, 48.0, 48.3, 48.1, 48.4, 48.1, 47.9, 48...
```

EDA is simple now that we have tidy data!

Here are a cople of examplesâ€¦

1. Overall summary.
2. Plot New Mexico monthly employment by year from 2004 through 2015.

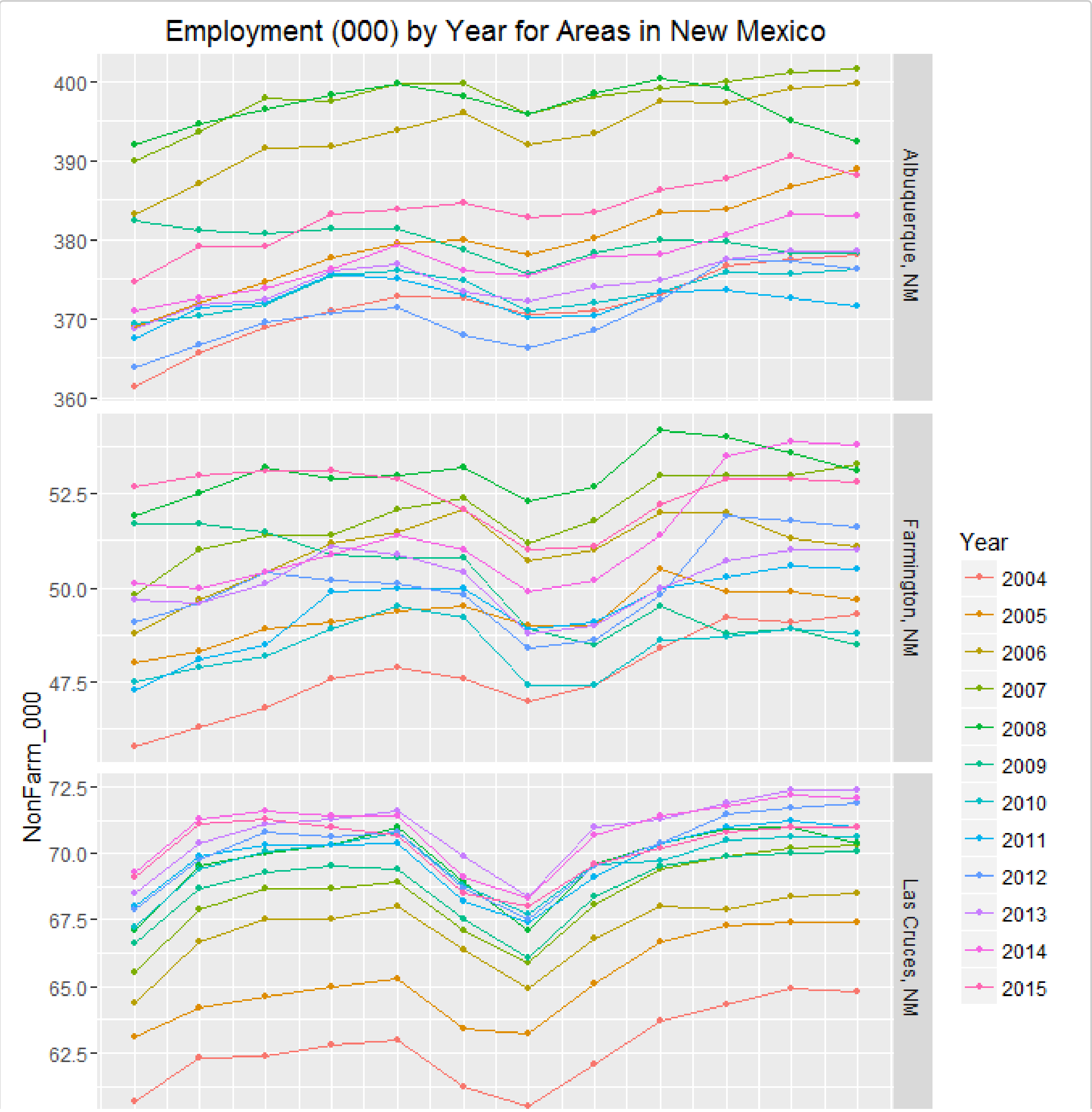
```
summary(Employment_By_Area_1995_2015)
```

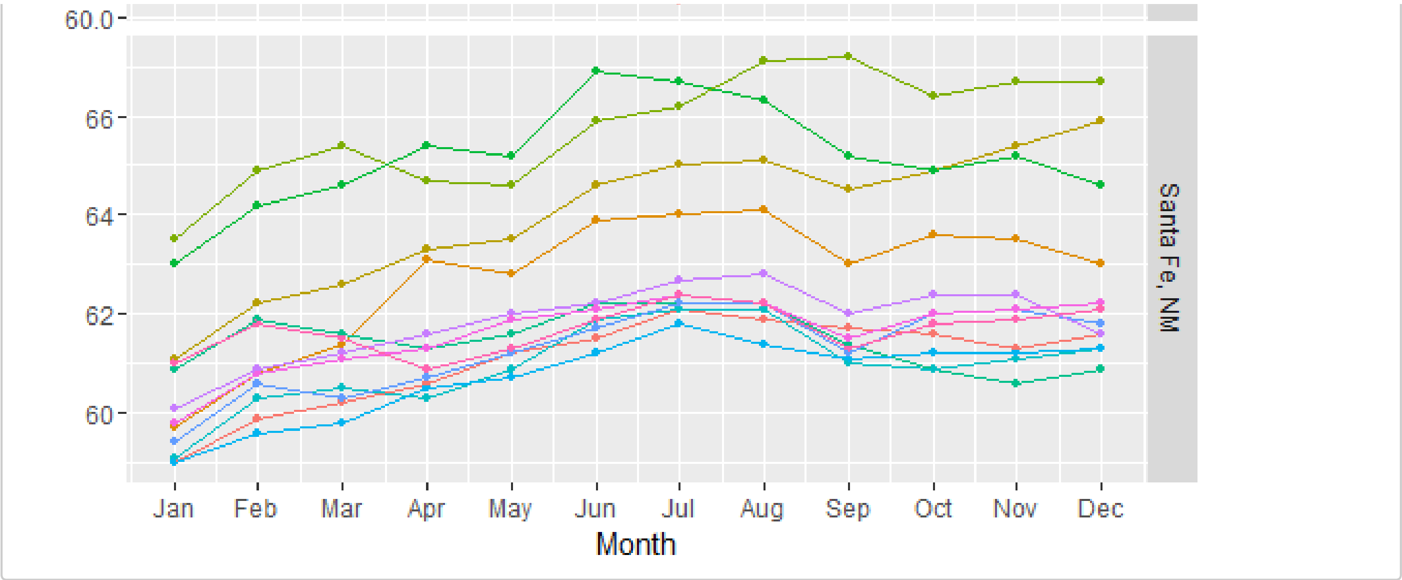
```
##            State                                Area
##  California   : 7812   Abilene, TX                 :   252
##  Texas        : 6804   Akron, OH                   :   252
##  Florida      : 6300   Albany-Schenectady-Troy, NY:   252
##  Pennsylvania : 5544   Albany, GA                  :   252
##  New York     : 4284   Albany, OR                  :   252
##  Massachusetts: 4032   Albuquerque, NM             :   252
##  (Other)      :75096   (Other)                     :108360
##     Month_Of            NonFarm_000
##  Min.   :1995-01-01   Min.   :   7.6
##  1st Qu.:2000-03-24   1st Qu.:  57.1
##  Median :2005-06-16   Median : 108.8
##  Mean   :2005-06-16   Mean   : 369.7
##  3rd Qu.:2010-09-08   3rd Qu.: 293.2
##  Max.   :2015-12-01   Max.   :9469.2
##                       NA's   :60
```

```
# pull out New Mexico data
AreasInNM<- Employment_By_Area_1995_2015 %>%
  filter(State == "New Mexico",
         Month_Of >= as.Date("2004-01-01")) %>%
  mutate(Year = factor(year(Month_Of)),
         Month = month(Month_Of)) %>%
  droplevels()

# get month labels for plot
months <- month(seq(as.Date("2000/1/1"), by = "month", length.out = 12),
                label = TRUE, abbr = TRUE)

ggplot(AreasInNM, aes(Month, NonFarm_000, color = Year)) +
  geom_point(size = I(1)) + geom_line() +
  scale_x_continuous(breaks = 1:12, labels = months) +
  ggtitle("Employment (000) by Year for Areas in New Mexico") +
  facet_grid(Area ~ ., scales = "free_y" )
```





Learning More

The place to start, of course, is Hadleyâ€™s vignettes in the dplyr and tidy packages. Especially Introduction to dplyr and Tidy Data.

Now that Hadley is with RStudio, search their blog for dplyr and tidyr; get the Data Wrangling Cheat Sheet; watch Data Wrangling with R & RStudio. To understand Hadleyâ€™s current thinking about data analysis watch Pipelines for Data Analysis in R and The Grammar and Graphics of Data Science - the latter with Winston Chang.

Lastly, see Garrett & Hadleyâ€™s chapter on data transform in their upcoming R for Data Science

Conclusion

We hope you have found this example of using dplyr and tidyr useful. Please send comments and suggestions to Jim at DS4CI.org or leave an issue or pull request at my github.

Thanks! Jim

P.S. Donâ€™t forget to clean house.

```
file.remove(file_out)
```

```
## [1] TRUE
```

END