U.S. Births & Unemployment Rate 2007 - 2012

Daniel Dittenhafer

Monday, December 8, 2014

Preface

This study was performed as part of the Data Acquisition and Management (IS607) course requirements for the Master of Science, Data Analytics program at City University of New York (CUNY).

Introduction

The U.S. Department of Health and Human Services (HHS) aggregates birth related data from around the United States and, among other things, makes it available on their website for public access (HHS, 2014). Using this data in combination with census data from the U.S. Census Bureau (U.S. Census Bureau, 2014) and unemployment data from the Bureau of Labor Statistics (U.S. Bureau of Labor Statistics, 2014), this study analyzes how birth rates have changed from 2007 (prior to the great recession) through to 2012 (the most recently available year for natality data from HHS).

Data Set Profile

United States Natality 2007 - 2012 The following table describes the HHS Natality data at a high level.

Data Set Characteristics	Number of Observations	Area	Attribute Characteristics	Number of Attributes	Missing Values?
Multivariate	41,184	Health	Categorical, Integer	10	See Note

- The Natality data is partitioned by year, month, state and county.
- For counties with populations less than 100,000, births are binned together into an Unidentified Counties entry.

United States Census 2010 The following table describes the U.S. Census data at a high level.

Data Set Characteristics	Number of Observations	Area	Attribute Characteristics	Number of Attributes	Missing Values?
Multivariate	3,194	Civil	Categorical, Integer	40+	No

- The Census data is partitioned by state and county, and contains many more attributes than are used in this study.

United States Unemployment Rate 2007 - 2012 The following table describes the U.S. Unemployment data at a high level.

Data Set Characteristics	Number of Observations	Area	Attribute Characteristics	Number of Attributes	Missing Values?
Multivariate	72	Economic	Year/Month, Integer	2	No

- The Unemployment data is partitioned by year and month.

Methodology

At a high level, the methodology employed in this study follows Hadley Wickham's Grammar of Data Science (Wickham and Chang, 2014). Specifically, this includes acquiring data, tidying the data, analyzing the data, and communicating information regarding the data.

The R language was used to perform the data analysis presented in this study. As such, the following R packages are used throughout the code to enable various data loading, transformation and visualization aspects of the study.

```
require.package("ggplot2")
require.package("plyr")
require.package("reshape2")
require.package("ggmap")
require.package("sp")
require.package("maptools")
require.package("grid")
require.package("gridExtra")
```

The data sets used throughout this study are hosted in my DataAcqMgmt repository on GitHub to enable a more reproducible research experience. Since GitHub uses SSL, `setInternet2` will help Windows users if they choose to reproduce the results presented here in. If you choose to reproduce this study using the original R Markdown, please ensure you have a reliable high-speed internet connection before proceeding.

```
gitHubRoot <- "https://github.com/dwdii/DataAcqMgmt/raw/master/FinalProject/Data"
if(.Platform$OS.type == "windows") {
  setInternet2(TRUE)
}
```

HHS provides various data sets related to natality, though the most recent year available is 2012. Given our interest in the period including the 2008 recession, the 6 year period from 2007 - 2012 was downloaded and posted to GitHub for this study. The data is tab delimited and although many values are integer in nature (State Code and County Code for example), they are better loaded as character data to preserve '0' padding. HHS includes data set notes which result in `NA` values in the Births data column. These NA values were removed. Also the `Year.Code` and `Month.Code` are combined to give a single `Date` column for use in aggregations later.

```
#
# FUNCTION: loadBirthData
#
loadBirthData <- function()
{
  # Load the Natality data
  birthFile <- sprintf("%s/Natality, 2007-2012-StateCounty.txt", gitHubRoot)
  birthData <- read.table(birthFile,
```

```
                          header=TRUE,
                          sep="\t",
                          fill=TRUE,
                          stringsAsFactors=FALSE,
                          colClasses=c('character', # Notes
                                       'character', # Year
                                       'character', # Year.Code
                                       'character', # Month
                                       'character', # Month.Code
                                       'character', # State
                                       'character', # State.Code
                                       'character', # County
                                       'character', # County.Code
                                       'numeric'))  # Births

  # Eliminate rows with no birth data (some rows have comments only)
  birthDataWoNa <- subset(birthData, !is.na(birthData$Births))

  # Transform raw year/month columns into a Date column
  birthDataWoNa <- mutate(birthDataWoNa,
                          Date = lubridate::parse_date_time(sprintf("%s-%s-01",
                                                                    Year.Code,
                                                                    Month.Code),
                                                            orders="ymd"))
  return (birthDataWoNa)
}
```

The following function, `loadCensusData`, loads the census data from the GitHub repository. Three new columns are added to the data set. State and County codes are concatenated to form the same combined identifier `County.Code` found in the Natality data. Additionally, the population values are reduced by some orders of magnitude for use in the birth rate normalization performed for the geographic visualization later.

```
#
# FUNCTION: loadCensusData
#
loadCensusData <- function()
{
  # Load the Census data
  dataFile <- sprintf("%s/CO-EST2013-Alldata.txt", gitHubRoot)
  data <- read.table(dataFile,
                     header=TRUE,
                     sep=",",
                     quote="",
                     fill=TRUE,
                     stringsAsFactors=FALSE,
                     colClasses=c('character'))

  data <- mutate(data, County.Code=paste(STATE, COUNTY, sep=""))
  data <- mutate(data, CENSUS2010POPThousands=as.numeric(CENSUS2010POP) / 1000)
  data <- mutate(data, CENSUS2010POPHundreds=as.numeric(CENSUS2010POP) / 100)

  return (data)
}
```

Like the other data sets, the Unemployment Rate data set is downloaded via a helper function `loadUnemploymentData`, and prepared for analysis. The U.S. Bureau of Labor Statistics publishes the Unemployment rate in a wide format whereby rows are by year, and columns represent each month of the year with the Unemployment rate as the values of these cells. As part of the tidying activity, this wide format is melted to a long format more useful for our analysis. Again, the year and month columns are combined into a single `Date` column.

```
#
# FUNCTION: loadUnemploymentData
#
loadUnemploymentData <- function()
{
  # Load the Unemployment data
  dataFile <- sprintf("%s/USUnemploymentRates2007-2012.csv", gitHubRoot)
  data <- read.table(dataFile,
                     header=TRUE,
                     sep=",",
                     fill=TRUE,
                     stringsAsFactors=FALSE)

  # Melt the data to a long format
  data <- melt(data,
               id.vars=c("Year"),
               variable.name="Month",
               value.name="UnemploymentRate")

  # Transform raw year/month columns into a Date column, sorted
  data <- mutate(data,
                 Date = lubridate::parse_date_time(sprintf("%s-%s-01",
                                                           Year,
                                                           Month),
                                                   orders="ybd"))
  data <- data[order(data$Date), ]

  return (data)
}
```

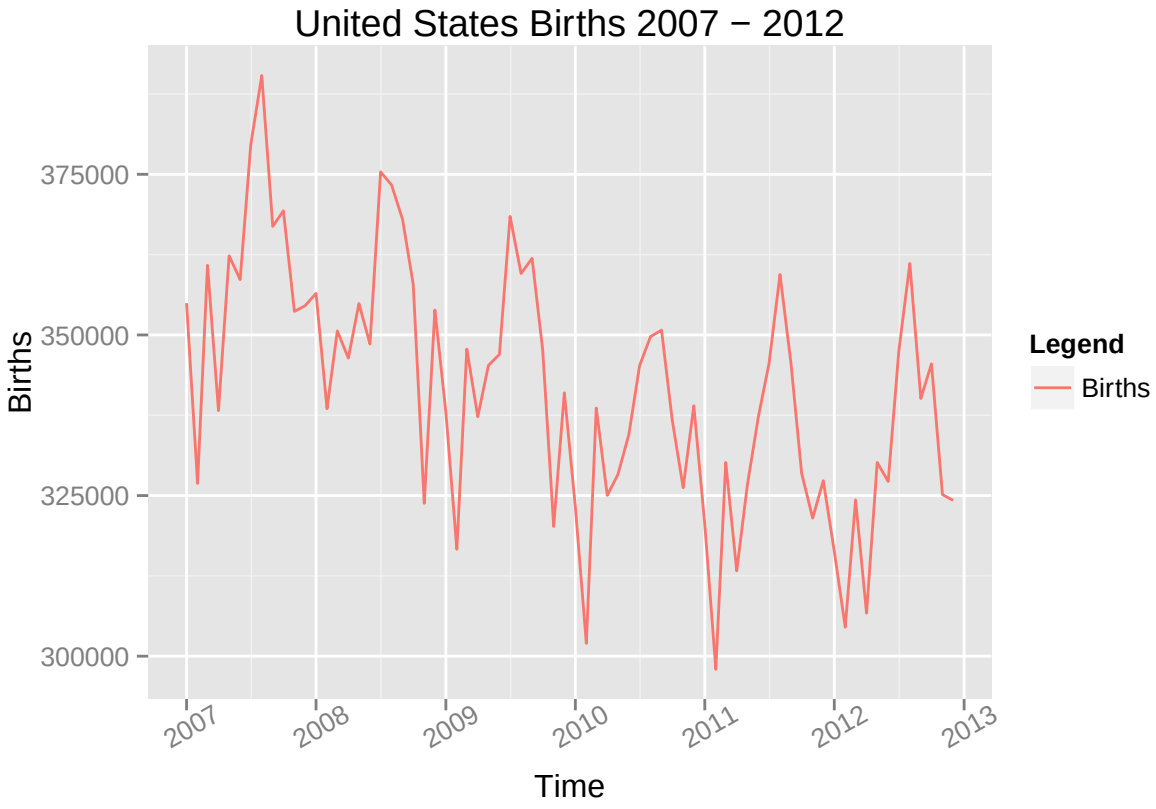
Data Analysis

First, using the functions defined above, the natality and census data are loaded into our session.

```
birthDataWoNa <- loadBirthData()
censusData <- loadCensusData()
```

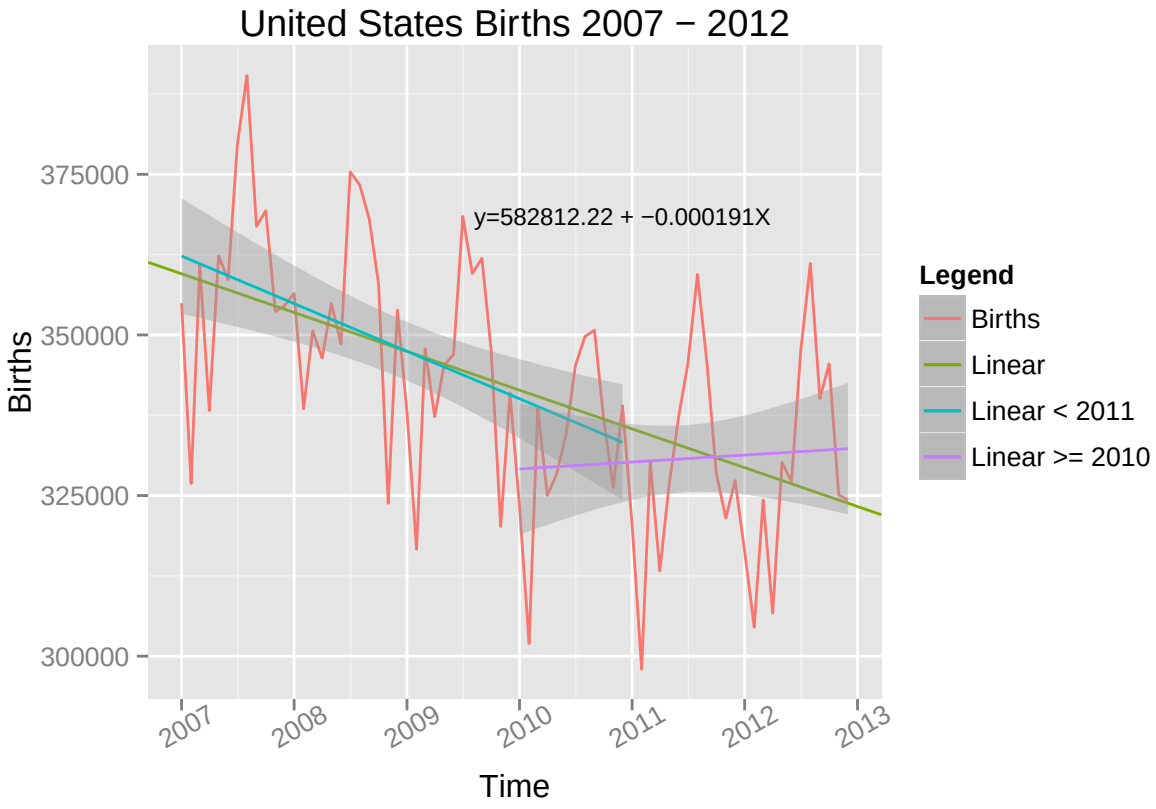
We can determine how many births occurred each month by aggregating the birth data by date. The `Date` column in this case is a monthly granularity as a result of the transformation applied in the `loadBirthData` function.

```
birthsPerYearMth <- aggregate(Births ~  Date, birthDataWoNa, sum)
birthsPerYearMth <- birthsPerYearMth[order(birthsPerYearMth$Date), ]
```



Besides the apparent seasonality to the birth data (with peaks in late summer/early fall), it looks like there is a downward trend and possibly a trough during 2010. Lets run a linear regression and overlay on the existing chart.

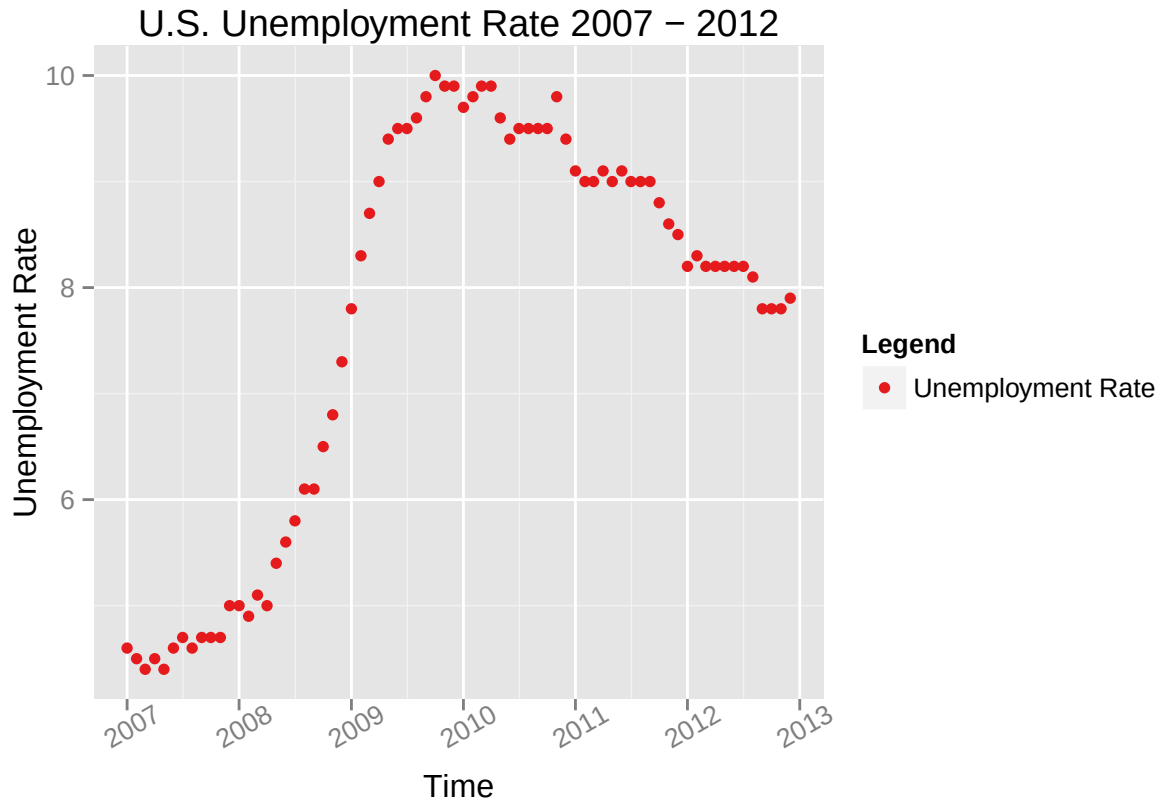
```
fit <- lm(Births ~ Date, data=birthsPerYearMth)
```



The "Linear" line is an ordinary least squares (OLS) linear regression on the entire data set, while the "Linear < 2011" line shows an OLS linear regression of only the data between 2007 - 2010 with a 95% confidence interval shown. Likewise, the "Linear >= 2010" line shows an OLS linear regression for the data between 2010 - 2012. As can be seen by the "Linear" line, there is definitely a negative trend overall, but not a drastic one... the slope is only -0.000191. Notice that when we differentiate the pre-2011 data from 2010 onward, the slope of the regression line flips. This is evidence of the suspected trough in 2010 mentioned earlier. After 2010, there is a definite rise in births.

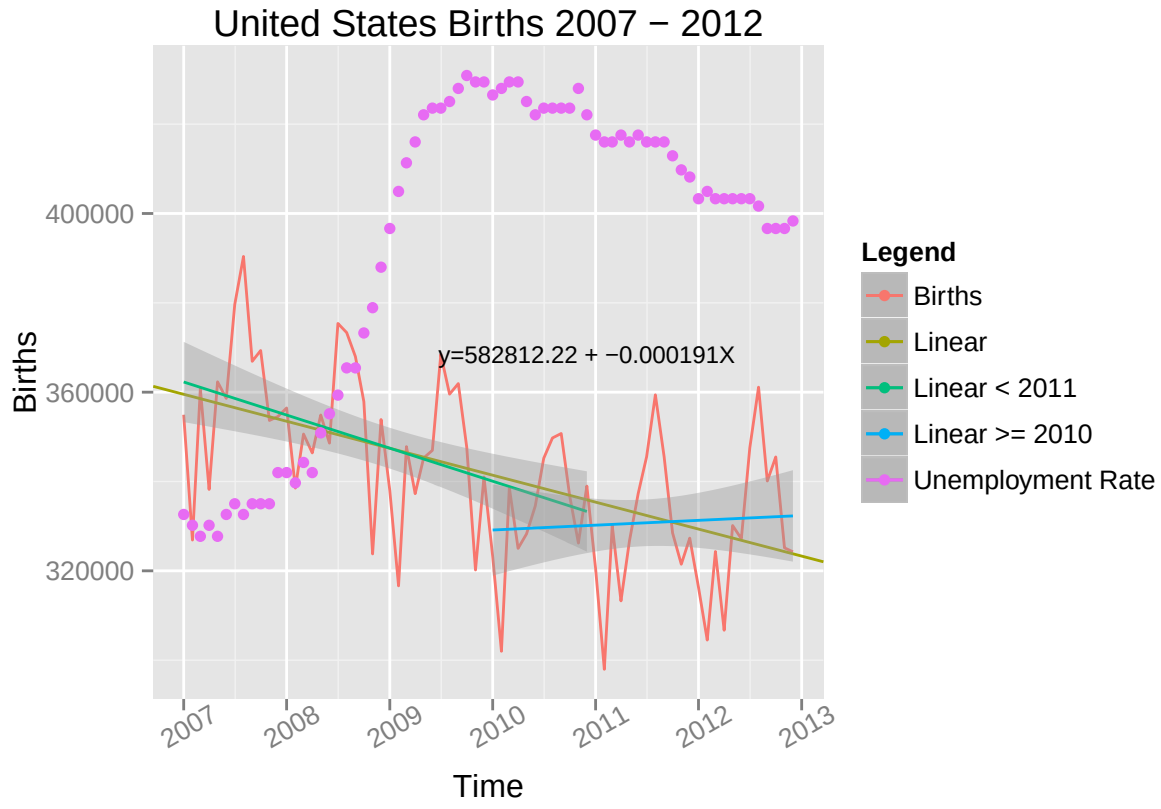
Ok, lets bring unemployment data for this same time frame in the picture.

```
unempData <- loadUnemploymentData()
```



As shown in the unemployment rate chart, above, unemployment increased significantly during the time of the great recession (2008 and 2009). Let's overlay unemployment (slightly transformed through multiplication and rooting) on our Births chart and see how things line up. The following `modelRoot` variable and `nth.root` function are used in the transformation as well as later in the proposed model code.

```
modelRoot <- 3
nth.root <- function(y, root)
{
  return (y ^ (1/root))
}
```



Admitedly, the chart is getting a bit busy, and its hard to judge if there is a good relationship between births and unemployment, but it seems like there might be.

After playing around with various functional forms, the following form, defined in LaTeX/Mathematics, provided a reasonable outcome as will be shown shortly (Wikibooks contributors, 2014).

$$y = B_0 + B_1 X_1 + B_2 \sqrt[3]{X_1} + B_3 X_1 \sqrt[3]{X_1}$$

Let's examine the results when the above functional form is applied to the unemployment and birth data:

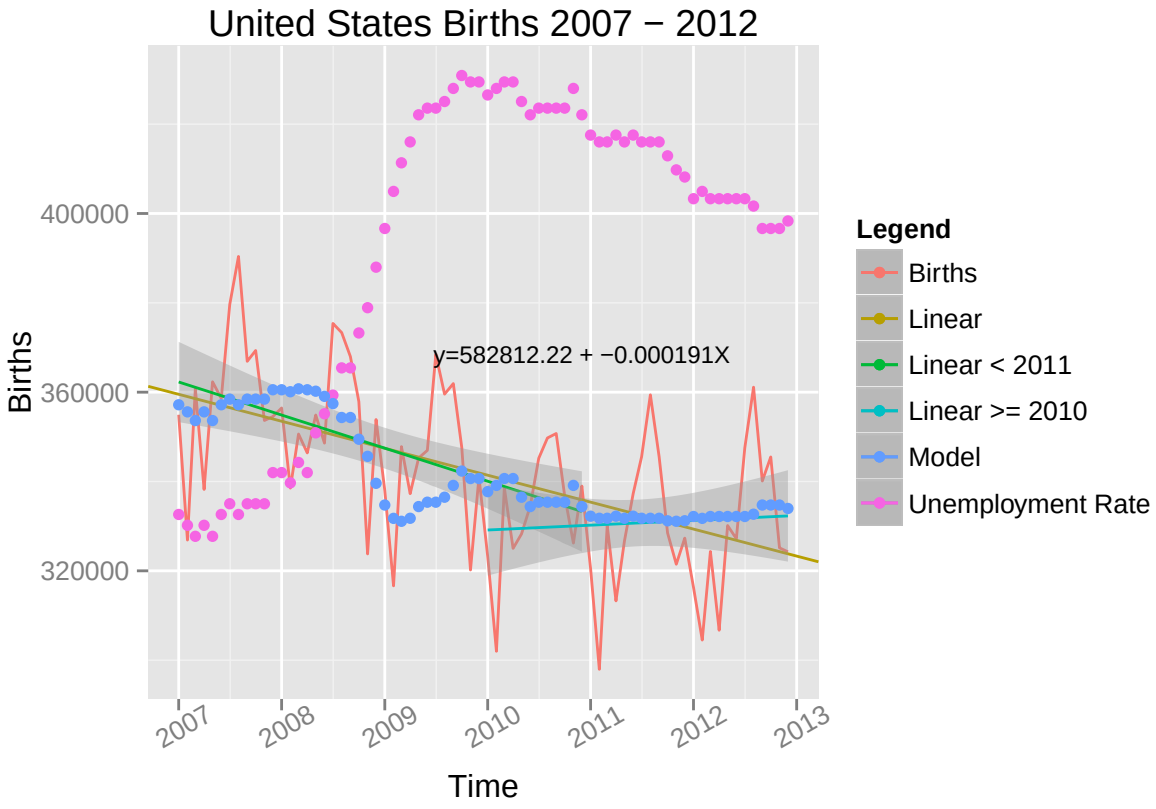
```
# Bind the two time series data together
tsData <- ts.intersect(B=as.ts(birthsPerYearMth), U=as.ts(unempData))
# Run the regression using the proposed functional form
fit2 <- lm(B.Births ~ U.UnemploymentRate*nth.root(U.UnemploymentRate, modelRoot),
           data=tsData)
model.summary <- summary(fit2)
print(model.summary)
```

```
##
## Call:
## lm(formula = B.Births ~ U.UnemploymentRate * nth.root(U.UnemploymentRate,
##     modelRoot), data = tsData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -37132 -10207   -734  11063  33201
```

```
##
## Coefficients:
##                                                                 Estimate
## (Intercept)                                                     -5255729
## U.UnemploymentRate                                              -1723086
## nth.root(U.UnemploymentRate, modelRoot)                          6049242
## U.UnemploymentRate:nth.root(U.UnemploymentRate, modelRoot)       454701
##                                                                 Std. Error
## (Intercept)                                                        2518906
## U.UnemploymentRate                                                  729921
## nth.root(U.UnemploymentRate, modelRoot)                            2657391
## U.UnemploymentRate:nth.root(U.UnemploymentRate, modelRoot)          190173
##                                                                 t value
## (Intercept)                                                       -2.09
## U.UnemploymentRate                                                -2.36
## nth.root(U.UnemploymentRate, modelRoot)                            2.28
## U.UnemploymentRate:nth.root(U.UnemploymentRate, modelRoot)         2.39
##                                                                 Pr(>|t|)
## (Intercept)                                                        0.041 *
## U.UnemploymentRate                                                 0.021 *
## nth.root(U.UnemploymentRate, modelRoot)                            0.026 *
## U.UnemploymentRate:nth.root(U.UnemploymentRate, modelRoot)         0.020 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16100 on 68 degrees of freedom
## Multiple R-squared:  0.326,  Adjusted R-squared:  0.296
## F-statistic: 10.9 on 3 and 68 DF,  p-value: 5.92e-06
```

From the R-squared included in the model summary shown above, the model explains approximately 32.6 % of the total variability in the Birth data. Graphically, we can see that the model follows the linear regressions fairly closely, though it doesn't predict any seasonality.

$$y = -5255729.04 + -1723085.49X_1 + 6049241.70\sqrt[3]{X_1} + 454700.50X_1\sqrt[3]{X_1}$$



Next Steps

A couple of questions come to mind related to validation and improvement. How does this base model predict historical U.S. births given unemployment for the same period? Likewise, how should seasonality be integrated into the model? These questions and others are out of scope for this study but are natural extensions which would add to the results presented herein.

Conclusions

Though a validation using pre-2007 (or post-2012) natality data was not performed, the model presented in this study provides a foundation for further research into the relationship between unemployment rate and births in the United States. At a minimum, this study appears to confirm that there is some relationship between level of employment and by extension, economic activity, in the United States and births in following months and years.

Source Code

The raw R markdown code used to produce this study can be found on GitHub, in my DataAcqMgmt repository.

Appendix

As part of the discovery process used to better understand the HHS Natality data, a geographic visualization was produced. Although the visualization didn't fit neatly into the birth/unemployment study, the

transformations, code and an example visualization are included herein for the benefit of readers and future researchers.

The ggmap package helps to visualize data geographically and the birth data comes complete with county identifiers. As such, shape data for United States counties was acquired from the U.S. Census Bureau (U.S. Census Bureau, 2013). The following function downloads this shape data and initializes it for use by ggmap (Kahle and Wickham, 2013).

```
#
# FUNCTION: loadShapeData
#
loadShapeData <- function(fileNoExt)
{
  # Download the shape files from the cloud.
  shpFile <- NA
  shapeFileExts <- c(".shp", ".shx", ".dbf")
  for(i in 1:length(shapeFileExts))
  {
    shpFileUrl <- sprintf("%s/%s%s", gitHubRoot, fileNoExt, shapeFileExts[i])
    tmpFile <- sprintf("%s\\%s%s", tempdir(), fileNoExt, shapeFileExts[i])

    download.file(shpFileUrl, tmpFile, method="internal", mode="wb")
    if(shapeFileExts[i] == ".shp") {
      shpFile <- tmpFile
    }
  }

  # Read data into R
  shapefile <- readShapeSpatial(shpFile,
                                proj4string = CRS("+proj=longlat +datum=WGS84"),
                                IDvar="GEOID")
  # Convert to a format ggmap likes
  shapeData <- fortify(shapefile)

  return (shapeData)
}
```

The geoVisual helper function was developed to integrate the shape data with the individual county birth rate data. Online resources from various sources were used to produce this function including posts by Kevin Johnson, and Wendi Wang (Johnson, 2014), (Wang, 2013),

```
#
# FUNCTION: geoVisual
#
#
geoVisual <- function(shapeData, data, title, filename, saveLocation)
{
  # integrate the birth data
  shapeData <- join(shapeData, data, by='id')
  shapeData <- dplyr::mutate(shapeData, BirthsNormalized = BirthsPer1000Pop)

  if(TRUE) {
  # Center the map over the United States
  map <- get_map(location=c(-97.279404, 39.828127), zoom=4)
```

```
  gmap <- ggmap(map)
  gmap <- gmap + scale_fill_gradientn(colours=rainbow(50, start=0.5),
                                      limits=c(.5, 2),name="Births/1000",
                                      guide=guide_colourbar(barwidth=0.5))

  gmap <- gmap + geom_polygon(aes(x = long, y = lat, group = group, fill=BirthsNormalized),
                              data = shapeData, #,
                              colour = 'white',
                              alpha = .5,
                              size = .1)
  gmap <- gmap + xlab("Births per 1000 people by County based on 2010 Census")
  gmap <- gmap + ylab("")
  gmap <- gmap + theme(axis.ticks = element_blank(),
                       axis.text = element_blank(),
                       axis.title = element_text(size=8),
                       plot.title = element_text(size=10),
                       legend.title = element_text(size=6))
  gmap <- gmap + ggtitle(title)

  sFooter <- "Created by Daniel Dittenhafer; Source: U.S. Health & Human Services"
  gmapft <- arrangeGrob(gmap,
                        sub=textGrob(sFooter,
                                     x = 0,
                                     hjust = -0.1,
                                     vjust=0.1,
                                     gp = gpar(fontface = "italic", fontsize = 6)))

  plot(gmap)

  if(!is.na(saveLocation)) {
      ggplot2::ggsave(sprintf("%s%s.png", saveLocation, filename),
                      plot=gmapft,
                      width=5, height=4)
    }
  }
}
```

In order to better visualize the Unidentified Counties mentioned in the data set profile, the `fillinCounties` function was developed to distribute the "Unidentified Counties" births for a given state proportionally across those counties (after being reidentified).

```
#
# FUNCTION: fillinCounties
#
fillinCounties <- function(r, reident)
{
  counties <- subset(reident, reident$STNAME == r$State)
  countyLen <- nrow(counties)

  dfCountiesYearMonth <- data.frame(
        Notes=rep(NA, countyLen),
        Year=rep(r$Year, countyLen),
        Year.Code=rep(r$Year.Code, countyLen),
```

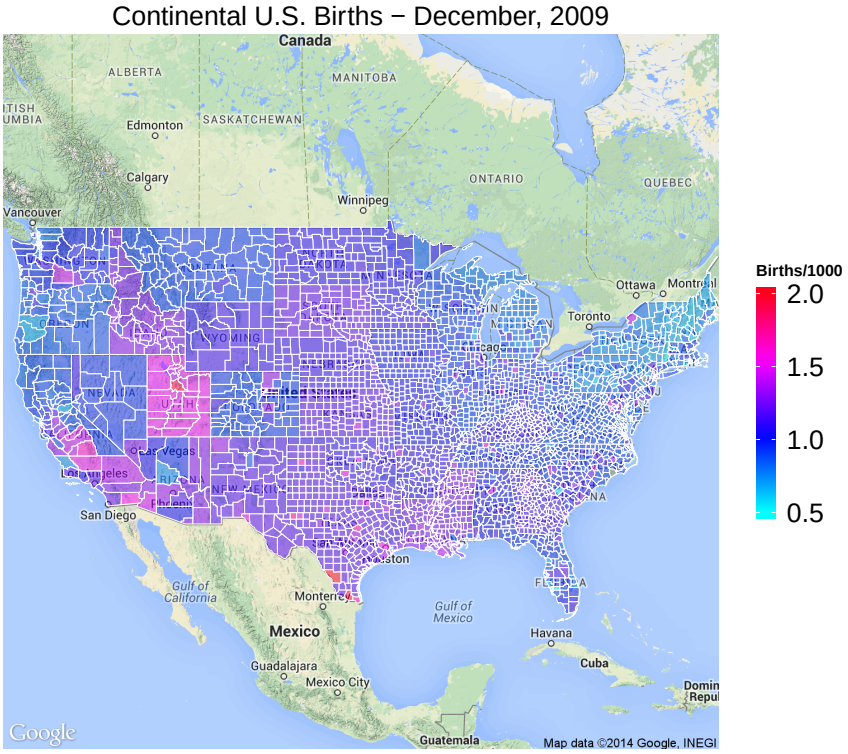
```
        Month=rep(r$Month, countyLen),
        Month.Code=rep(r$Month.Code, countyLen),
        State=rep(r$State, countyLen),
        State.Code=rep(r$State.Code, countyLen),
        County=paste(counties$CTYNAME, ", ", counties$STNAME, sep=""),
        County.Code=counties$County.Code,
        Births=r$Births * (as.numeric(counties$CENSUS2010POP) /
                             sum(as.numeric(counties$CENSUS2010POP))),
        Date=rep(lubridate::parse_date_time(sprintf("%s-%s-01",
                                                    r$Year.Code,
                                                    r$Year.Code),
                                            orders="ymd"), countyLen)
        )

  return (dfCountiesYearMonth)
}
```

The following code block is the driver code for the geographic visualization. It performs the following key operations:

- Extract county census data from the raw census data set (County "000" is the state level aggregate population).
- Select out the "Unidentified Counties" rows from the natality data.
- Identify the "Unidentified Counties" using a "not in" clause.
- Apply the `fillinCounties` function to the "Unidentified Counties" data subset using the newly identified counties.
- Bind the results from ddply to the origin birth data.
- Normalize all the birth data using the census data.
- Select just the data needed for the visualization and update the `County.Code` column name to match the fortified shape data column `id`.
- Finally, loop through the years and months to render the visualization as needed.

```
if(TRUE) {
  # Load the shape data
  shapes <- loadShapeData("cb_2013_us_county_20m")

  # Pull out just what we need of the census data
  censusCtyPop2010 <- subset(censusData,
                             censusData$COUNTY != "000",
                             select=c("County.Code",
                                      "STNAME",
                                      "CTYNAME",
                                      "CENSUS2010POP",
                                      "CENSUS2010POPHundreds",
                                      "CENSUS2010POPThousands"))

  # What are the unidentified counties?
  unidentCounties <- subset(birthDataWoNa,
                            stringr::str_detect(birthDataWoNa$County, "^Unidentified Counties,"))
  #print(head(unidentCounties))

  reidentCounties <- subset(censusCtyPop2010,
```

```
                            !(County.Code %in% birthDataWoNa$County.Code) )

  #print(head(reidentCounties, 100))

  # Attempt to divy up the Unidentified Counties across the rest of the state.
  unidentCountyAvg <- ddply(unidentCounties,
                            .variables=c("County.Code", "Year.Code", "Month.Code"),
                            .fun=fillinCounties, reidentCounties)
  birthDataWoNa <- rbind(birthDataWoNa, unidentCountyAvg)

  # Normalize to births per 1000 population
  #print(head(censusCtyPop2010))
  birthDataWoNa <- join(birthDataWoNa, censusCtyPop2010, by="County.Code")
  birthDataWoNa <- mutate(birthDataWoNa, BirthsPer100Pop=Births / CENSUS2010POPHundreds)
  birthDataWoNa <- mutate(birthDataWoNa, BirthsPer1000Pop=Births / CENSUS2010POPThousands)

  # select out the id and births columns
  subCountyBirth <- subset(birthDataWoNa, select=c('Year',
                                                   'Month',
                                                   'County.Code',
                                                   'BirthsPer100Pop',
                                                   'BirthsPer1000Pop'))
  subCountyBirth <- rename(subCountyBirth, c('County.Code'='id'))

  years <- c(2009) # 2007, 2008, 2009, 2010, 2011, 2012
  months <- c("January", "February", "March", "April", "May",
              "June", "July", "August", "September", "October",
              "November", "December")
  for(year in years) {
    for (m in seq(12, 12, by=1) ) {
      month = months[m]
      birthsYrMonth <- subset(subCountyBirth,
                              subCountyBirth$Year == year & subCountyBirth$Month == month)

      title <- sprintf("Continental U.S. Births - %s, %d", month, year)
      filename <- sprintf("%d_%00d_US_Births", year, m)

      geoVisual(shapes, birthsYrMonth, title, filename, NA)
    }
  }
}
```



Births per 1000 people by County based on 2010 Census

The final result is a map of the United States with the county-level birth data overlaid as shown above.

An animated version of the above visualization produced using the Gifmaker.me website can be found on GitHub, in my DataAcqMgmt repository (Gifmaker.me, 2014).

References

Gifmaker.me. Free Online Animated GIF Maker. 2014. URL: http://gifmaker.me/.

HHS. Natality public-use data 2007-2012 on CDC WONDER Online Database. Apr. 2014. URL: http://wonder.cdc.gov/natality-current.html.

Johnson, K. Making Maps in R. 2014. URL: http://www.kevjohnson.org/making-maps-in-r/.

Kahle, D. and H. Wickham. ggmap: Spatial Visualization with ggplot2. 2013. URL: http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf.

U.S. Bureau of Labor Statistics. Labor Force Statistics from the Current Population Survey. 2014. URL: http://data.bls.gov/timeseries/LNS14000000.

U.S. Census Bureau. Cartographic Boundary Shapefiles - Counties. 2013. URL: http://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html.

— County Totals: Vintage 2013. 2014. URL: http://www.census.gov/popest/data/counties/totals/2013/index.html.

Wang, W. Best way to add a footnote to a plot created with ggplot2. 2013. URL: http://bigdata-analyst.com/best-way-to-add-a-footnote-to-a-plot-created-with-ggplot2.html.

Wickham, H. and W. Chang. The Grammar and Graphics of Data Science. 2014. URL: http://pages.rstudio.net/Webinar-Series-Recording-Essential-Tools-for-R.html.

Wikibooks contributors. LaTeX/Mathematics. [Online; accessed 28-November-2014]. 2014. URL: http://en.wikibooks.org/w/index.php?title=LaTeX/Mathematics&oldid=2736437.