United States Natality Models 2003-2014

DATA 621: Business Analytics and Data Mining

Daniel Dittenhafer & Justin Hink

May 22, 2016

1 Abstract

The drivers of conception decisions have been studied for some time by economists and sociologists. Consultancies have been established to forecast birth activity, but a publicly available, unified model to forecast births in the United States would aid business and government with planning and resource allocation. In this study we utilize publicly accessible data sets from various United States federal agencies to research relationships and possible predictors of birth counts. The unemployment rate, women's earnings, gender ratio and census estimates of female population across several age ranges between 2003 and 2014 were considered in the model building process. Both multiple linear regression and generalized linear models were developed and analyzed. A multiple linear regression model was selected as the best performer, yielding a $R^2 = 0.52589$. We performed the analysis using the open source R runtime, and have shared our project materials to a GitHub repository to facilitate reproducible research.

2 Keywords

natality, births, demographic theory

3 Literature Review

As a starting point for this study, Daniel Dittenhafer has done prior work analyzing births and unemployment rate in the United States. Dittenhafer found a negative relationship between births and unemployment during the time period studied of 2007 - 2012 (Dittenhafer, 2014). Dittenhafer's single predictor linear model using unemployment rate alone yielded an adjusted R^2 of 0.296 with a p-value approaching 0. Although the unemployment-based model showed some interesting analysis, its usefulness is still to be determined. On the other hand, the negative relationship finding appears to be a relavent output of the study, and one which our current research supports.

Morgan and Taylor published a paper in The Annual Review of Sociology regarding recent fertility trends, and specifically a shift to lower birth rates as compared to the second half of the twentieth century (Morgan and Taylor, 2006). Our research did not relate to this directly, but further reductions in births were observed beginning in 2008. This may constitute another change point or may be a continuation of the trend studied by Morgan and Taylor.

We include women's earnings as a possible predictor in this study based on research by Aliaksandr Amialchuk regarding wage related effects on fertility (Amialchuk, 2013). Across all women, women's education, men's education, men's earnings and metro area were all found to be significant in age-specific fertility regression. We were not able to include age specific earnings, but rather a single earnings measure for women of child bearing age (Bureau of Labor Statistics, 2015).

4 Methodology

4.1 Technology

We used the R runtime via RStudio as our primary environment for all data steps including data preparation, exploration, analysis, model deveopment, validation and selection. A GitHub repository was setup to facilitate collaboration amoungst the team, as well as to share our work in the spirit of reproducible research.

4.2 Data Preparation

Data sets from the Census Bureau, Centers for Disease Control, and Bureau of Labor Statistics identified and downloaded to our project GitHub repository (Dittenhafer and Hink, 2016). These data sets were subsequently joined together in order to provide a unified data set for analysis and modeling.

4.2.1 Natality Data

The natality data including birth counts per month were acquired from the Centers for Disease Control and Prevention in two data sets. The first data set contained data for the years 2003 - 2006 (Centers for Disease Control and Prevention, 2009). The second data set contained data for the years 2007 - 2014 (Centers for Disease Control and Prevention, 2016). The data sets were merged together and augmented with additional census, earning and unemployment data as described in the following sections.

4.2.2 Census Data

For the period of May 2010 - Decemeber 2015, the Census Bureau's census data was available as monthly population estimates broken down by age and gender (Census Bureau, 2015). The age data was in whole year granulatity and we created 10 year buckets for the female population by age: 15-24, 25-34, and 35-44.

For the period of 2000 - April 2010, monthly population estimates were only available for the total population (Census Bureau, 2010). We used annual age and gender estimates from the Census Bureau's 2000 - 2010 time period (converted to ratios) to divide the monthly total population into age and gender bins as shown in the following expressions:

Gender Bins

For each year, 2003 - 2010:

$$G_{year} = \frac{F_{year}}{P_{year}}$$

$$F_{month} = P_{month} * G_{year}$$

$$M_{month} = P_{month} - F_{month}$$

Where:

- G Gender Ratio
- F Total females, TOT_FEMALE
- M Total males, TOT_MALE
- P Total population, TOT_POP

Age Bins

Again, for each year, 2003 - 2010:

$$F_{year\_x\_y} = \sum_{i=x}^{y-1} F_{year\_i}$$

$$A_{year\_x\_y} = \frac{F_{year\_x\_y}}{F_{year}}$$

$$F_{month\_x\_y} = F_{month} * A_{year\_x\_y}$$

Where:

x Lower age bound of bin

y Upper age bound of bin

A Age bin's ratio

4.2.3 Earnings Data

The earnings data was acquired from the Bureau of Labor Statistics and specifically covers women's weekly earnings from 2003 - 2015 (Bureau of Labor Statistics, 2015). The acquired data was at a quarter year granularity and was transformed to a monthly granularity for use in this study by simply assigning a quarter's weekly earnings to each of the related 3 months in the 12 month annual period.

4.2.4 Unemployment Data

Unemployment data (U3) was acquired from the Bureau of Labor Statistics. The data was at a monthly granularity with no transformations applied before use in the study (Bureau of Labor Statistics, 2015).

4.3 Data Exploration

We conducted exploratory data analysis to better understand the relationships in the data including correlations, feature distributions and basic summary statistics.

4.4 Model Development

Ten models were developed and examined for significance using a subset (80%) of the original full data set. Gaussian, Poisson and Negative Binomial linear models were fit using a variety of predictor variables and their significance, VIFs, adjusted R^2 and Akaike information criteria (AIC) were examined.

4.5 Model Validation

A validation data set (VS) was created from a subset of the full data set (20%). This VS data set was used to test how well our candidate models generalize to unseen data. The validation metric for the multiple linear regression models is the mean squared error from the validation set.

5 Results

5.1 Data Exploration

5.1.1 Correlations

The following table shows the correlation coefficients associated with each variable and the dependent variable, *Births*.

Table 1: Pearson's r Correlation Coefficients

Births	1.0000000
FEMALE_35_44	0.3880661
Month	0.3646307
GenderRatio	0.2862173
FEMALE_15_24	-0.2307949
TOT_MALE	-0.3214851

TOT_POP	-0.3219328
TOT_FEMALE	-0.3223760
Year	-0.3593053
Earnings	-0.3697992
UnemploymentRate	-0.3862666
FEMALE_25_34	-0.3879287

5.1.2 Seasonality

As one might expect, we saw seasonality in the birth data. As shown in the scatter plot, below, August is a very popular month for births. July and September are close behind. This suggests that many conceptions are occuring during the United States holiday season between Thanksgiving and New Years.

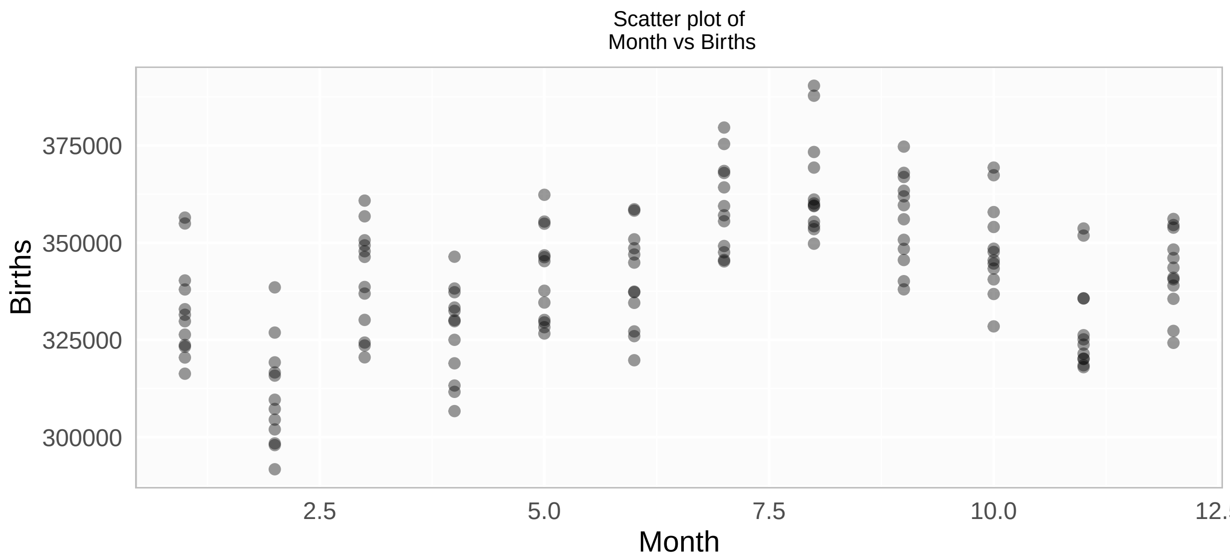


Figure 1: Month vs Births

5.1.3 Gender Ratio

The computed gender ratio which was used to enable the gender buckets for the period of 2003 - 2010 can be seen in the scatterplot below. For these years, the gender ratio is constant for all months of a given year while the birth counts fluctuate. Interestingly, the proportion of females has been dropping steadily, though only slightly during the time period being studied.

5.1.4 Earnings

Women's weekly earnings as a broad median value, as reported by the Current Population Survey via the Bureau of Labor Statistics, revealed a negative correlation with births, as previously shown.

As part of variance inflation factor (VIF) analysis, we found that the earnings measure we used was correlated with female population levels. As shown, in some cases this was a positive relationship (both 15-24 and 25-34), but for the 35-44 age range this was a negative relationship. In general, these relationship resulted in VIFs which significantly exceeded 10 when earnings and the female age values were included in a model. Further study on this relationship may be warranted inorder to better understand the drivers.

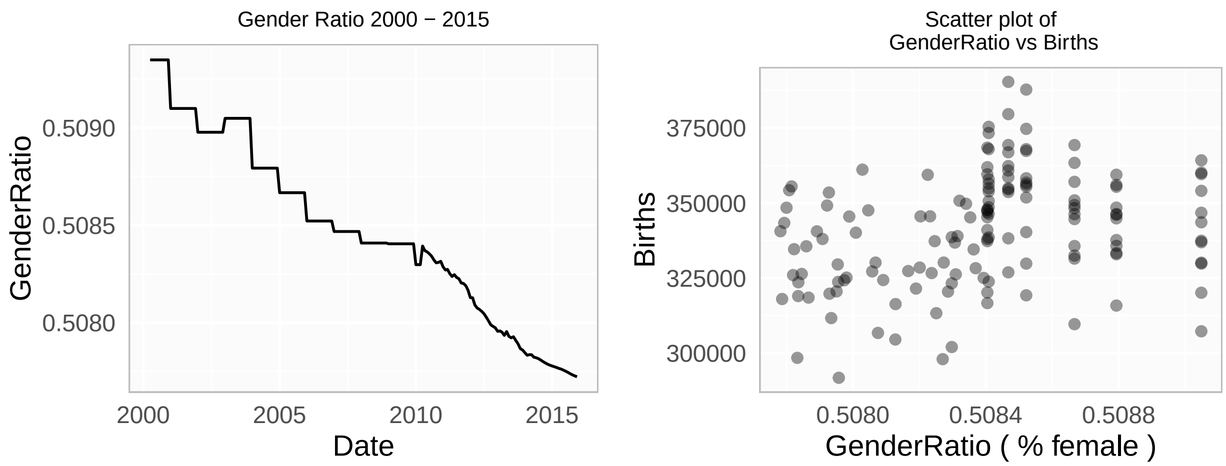


Figure 2: GenderRatio vs Date, GenderRatio vs Births

Figure 3: Earnings vs Births

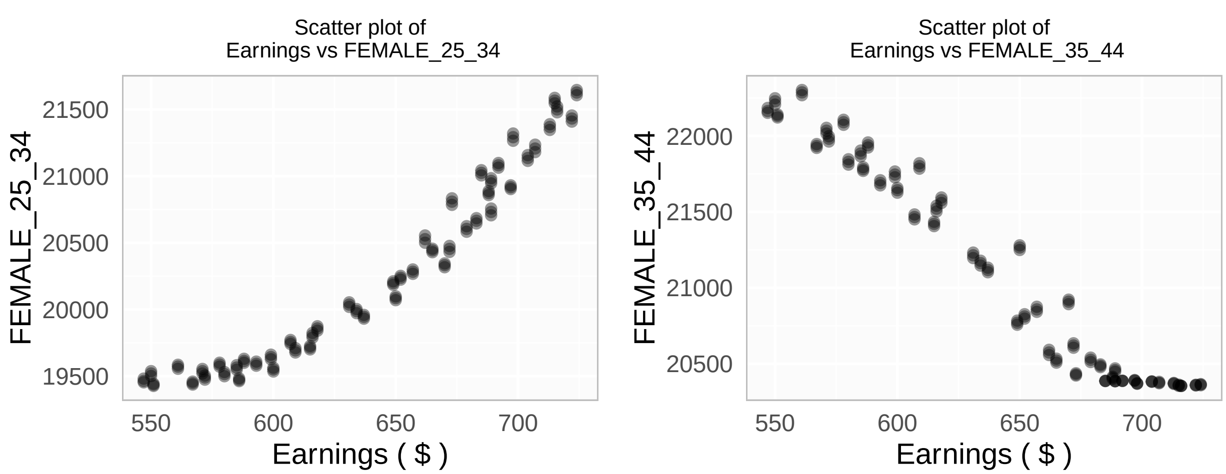


Figure 4: Earnings vs Female Population 25-34 and 35-44

5.2 Model Development

We started with all possible predictor variables from our data sets in a multiple linear regression model. This model yielded an adjusted R^2 of 0.61129, but had significant variance inflation factor issues (> 120,000,000).

Next, we took the significant predictors from the "All Variables" model and created a new, smaller model. Again, VIFs were large. Specifically, the population bins were highly related.

Table 2: Signficant Variables Linear Model VIFs

TOT_POP	97350400.65649
GenderRatio	47225.03676
TOT_FEMALE	93360042.80857
FEMALE_15_24	405.25307
FEMALE_25_34	481.02800
FEMALE_35_44	40.62177
Earnings	88.00442

A new model using guidance from the correlation analysis yielded our "High Correlation Variables" linear model. VIFs were better, but still well above our threshold of 10. Our fourth model was created using R's *step* function. The "Step" model had an impressive adjusted R^2 of 0.62001 for only 5 predictor variables (all significant at $\alpha < 0.05$). The variance inflation factors were improved, but unfortunately, still exceeded our threshold.

Next, we experimented with a revised "Significant Variables Minus" model which was based on the "Significant Variables" model but with three high VIF variables removed. Again, we saw high VIFs. Around this time, the relationship between the female age bins and earnings, and its impact on our data analysis became apparent.

A review of the variables and the introduction of a lag variable, $Month9Ago = Month - 9$, brought us to the next model, "Significant Limited". Again a multiple linear regression model, the "Significant Limited" model included 4 significant variables with an adjusted $R^2 = 0.52589$ and no VIF issues. This was promising, but we continued our investigation of possible models.

Table 3: Significant Variables Limited Linear Model Coefficient Estimates

	Estimate	Pr(>|t|)
Intercept *	468350.706460	0.0000000
Month *	2490.804603	0.0000000
Month9Ago *	2649.171132	0.0000000
FEMALE_25_34 *	-7.181586	0.0003757
UnemploymentRate *	-2190.844273	0.0030120

Table 4: Significant Variables Limited Linear Model VIFs

Month	1.062619
Month9Ago	1.028000
FEMALE_25_34	1.400205
UnemploymentRate	1.360376

A Poisson generalized linear model version of the "Significant Limited"" model was produced next. Like its sister model, the "Poisson Significant Ltd" looked good.

Table 5: Poisson Significant Limited Model Coefficient Estimates

	Estimate	Pr(>|z|)
Intercept *	13.1113316	0
Month *	0.0073299	0
Month9Ago *	0.0077043	0
FEMALE_25_34 *	-0.0000210	0
UnemploymentRate *	-0.0064324	0

A component we felt was missing was interaction between population and month. We again developed a model based on the "Signifcant Limited" model, but this time included an interaction term for *FEMALE_25_34* and *Month9Ago*. This did not perform as we anticipated and instead introduced VIF issues without any predictive benefit.

For our final two models we used R's *stepAIC* to produce Poission and Negative Binomial generalized linear models, respectively. The "Negative Binomial Step" looked promising also with 6 predictor variables, all but one significant, and no VIF issues.

5.3 Model Validation

As mentioned earlier, a validation data set was reserved for use confirming the performance of each of the developed models. We ran each of the ten previously described models through the validation data and computed the mean squared error (MSE) of the resulting model output. We also captured the Akaike Information Criterion (AIC) for each of the models for reference.

The results of the multiple linear regression model validation are shown below.

Table 6: Linear Model Validation Error Results

Model	VS Error	Adj R^2	AIC	Variables	VIF
All Variables	161072343	0.6112935	2504.690	11	BAD
Neg Binomial Step	161290241	NA	2506.956	10	BAD
Poisson Step	161316049	NA	40569.475	10	BAD
Step	172024296	0.6200088	2497.456	5	BAD
Poisson Signif Ltd	176055186	NA	51551.445	4	OK
Significant Limited	176416016	0.5258888	2522.176	4	OK
Signif Ltd w/ Interaction	177767094	0.5218164	2524.118	5	BAD
High Cor	212269346	0.4788873	2535.031	6	BAD
Significant	227028994	0.4771385	2536.351	7	BAD
Significant Minus	231634851	0.3600735	2557.915	5	BAD

The variance inflation factor issues limited us to the "Significant Limited" and "Poisson Significant Ltd" models for further investigation. They have the least complex models and were in the median range in terms of validation performance although the AIC value for the "Poisson Significant Ltd" was strangely high.

5.3.1 In Depth: Significant Limited Linear Model

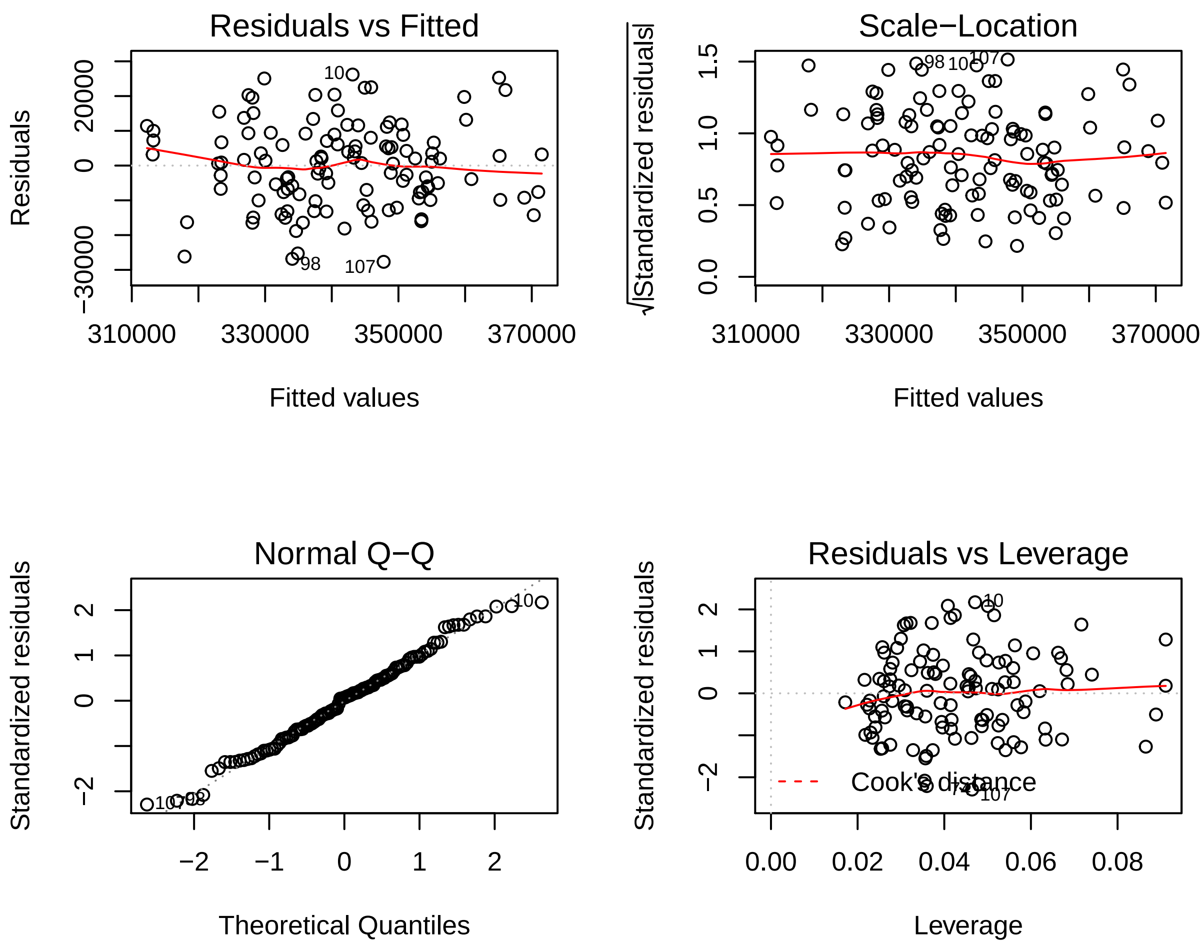
The Significant Limited model has an F-statistic of 32.89 and a mean squared error (MSE) of 146358133.79.

$$
\begin{aligned}
y_{births} = \quad 468350.7064601 \quad & +2490.8046026 x_{Month} \\
& +2649.1711324 x_{Month9Ago} \\
& -7.181586 x_{FEMALE\_25\_34} \\
& -2190.8442727 x_{UnemploymentRate}
\end{aligned}
$$

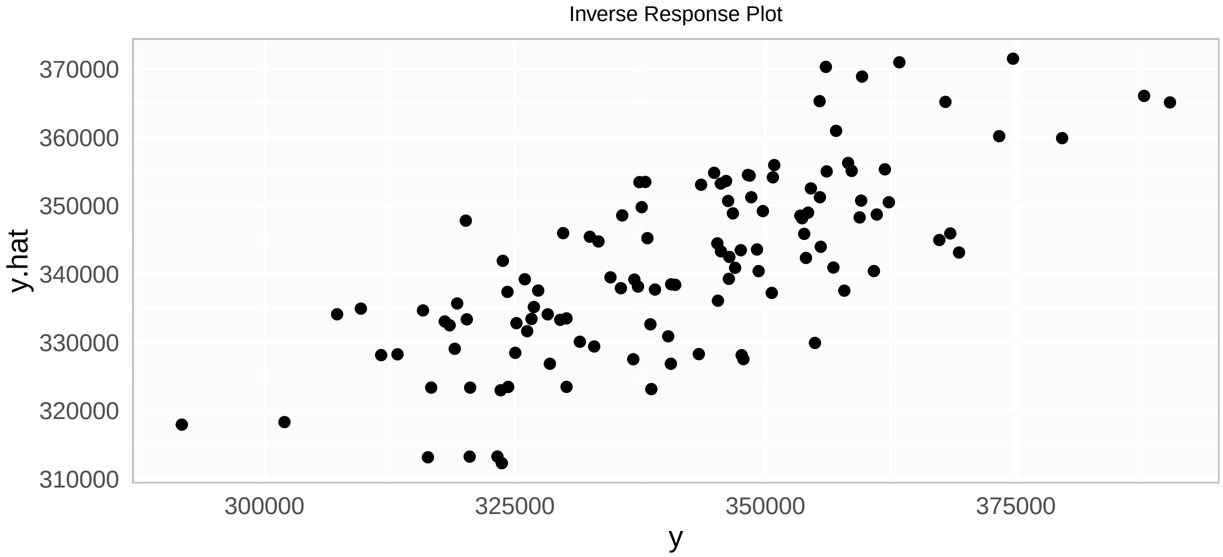
We can interpret the coefficients in the following manner. Holding all other predictors constant, for variable:

- *Month*, as the month of the year increased, a 2490.8 increase in births would occur.
- *Month9Ago*, as the 9 month lagged month of the year increased, a 2649.17 increase in births would occur.
- *FEMALE_25_34*, a unit increase in the population of females age 25-34 would yield a 7.18 decrease in births.
- *UnemploymentRate*, a unit increase in the *UnemploymentRate* related to a 2190.84 decrease in births.

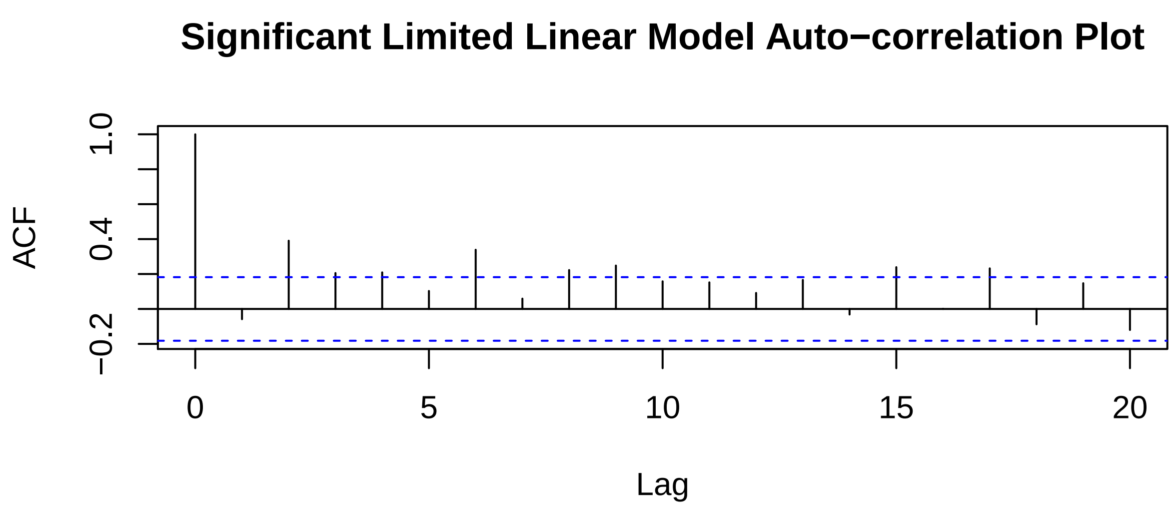
Linear regression diagnostic plots are shown below. Residuals appear to be normally distributed and variance seems to be fairly constant.



Looking at the inverse response plot, there does appear to be a good linear pattern to the predicted response versus actual.



Again, running an auto-correlation analysis with R's *acf* function shows a possible auto-correlation issue with lag 2 and 6.



5.3.2 In Depth: Poisson Significant Limited Model

The mathematical form of the Poisson Significant Limited Model is as follows:

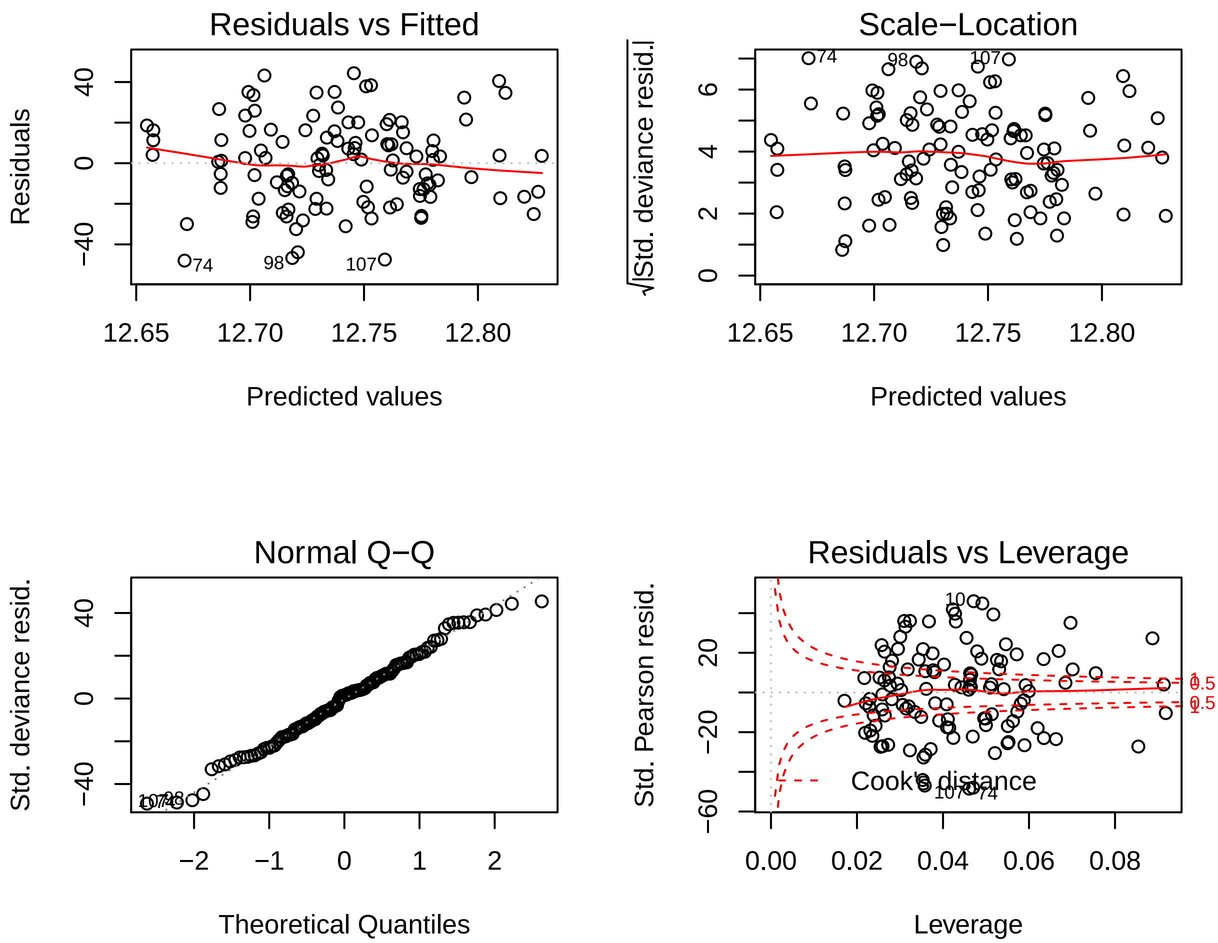
$$
\begin{aligned}
log\left(E(y_{births}|x)\right) = \quad 13.1113316 \quad & +0.0073299x_{Month} \\
& +0.0077043x_{Month9Ago} \\
& -0.000021x_{FEMALE\_25\_34} \\
& -0.0064324x_{UnemploymentRate}
\end{aligned}
$$

We can interpret the coefficients in the following manner. Holding all other predictors constant, for variable:

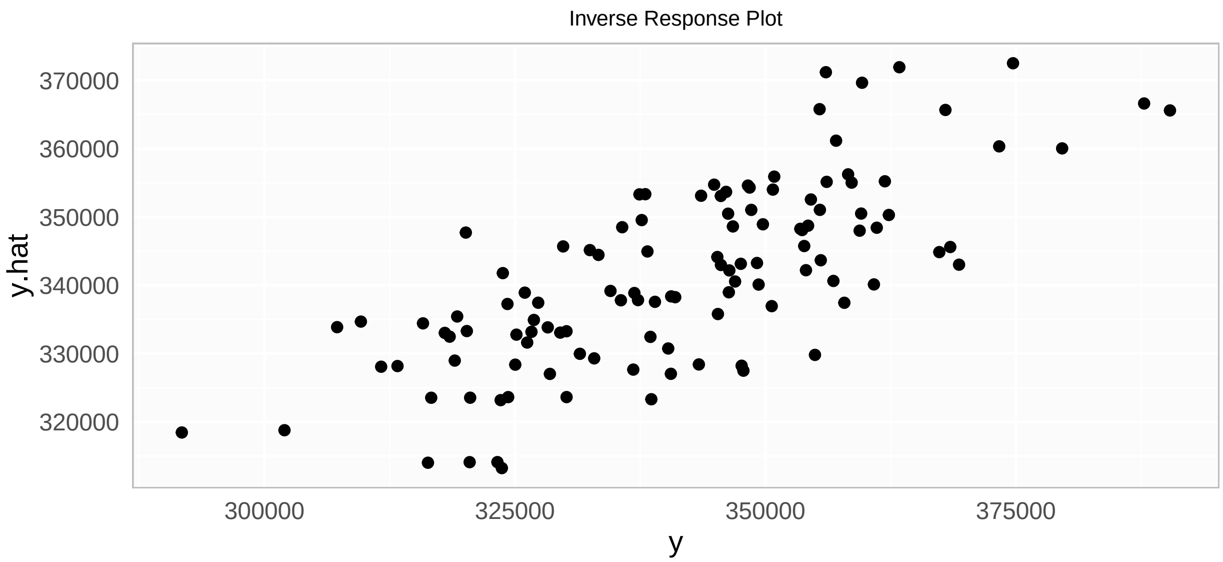
- *Month*, as the month of the year increased, a $e^{0.0073299} = 1.0073568$ times increase in births would occur.
- *Month9Ago*, as the 9 month lagged month of the year increased, a $e^{0.0077043} = 1.007734$ times increase in births would occur.
- *FEMALE_25_34*, a unit increase in the population of females age 25-34 would yield a $e^{-0.000021} = 0.999979$ times decrease in births.

- *UnemploymentRate*, a unit increase in the *UnemploymentRate* related to a $e^{-0.0064324} = 0.9935882$ times decrease in births.

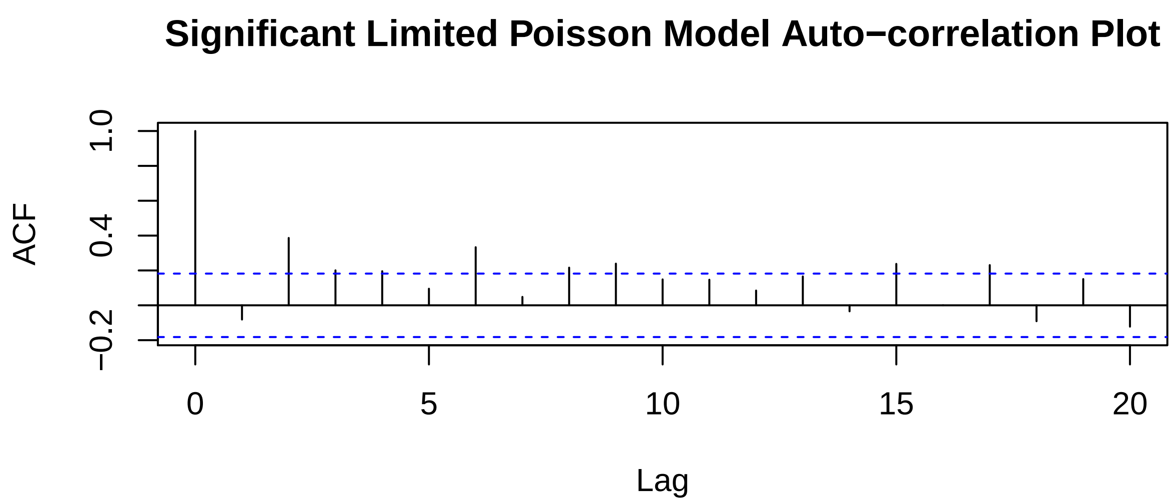
Regression diagnostic plots are shown below. Residuals appear to be normally distributed and variance seems to be fairly constant. The Leverage plot in the lower right shows many points which exceed Cook's distance and suggest points of high leverage.



Looking at the inverse response plot, there does appear to be a good linear pattern to the predicted response versus actual.



Again, running a more targeted auto-correlation analysis with R's *acf* function shows the same possible auto-correlation issue with lag 2 and 6.



6 Summary

Given the data set inputs we chose for this study, the Significant Limited multiple linear regression model offers the best outcome of those we analyzed. With that said, it has limitations.

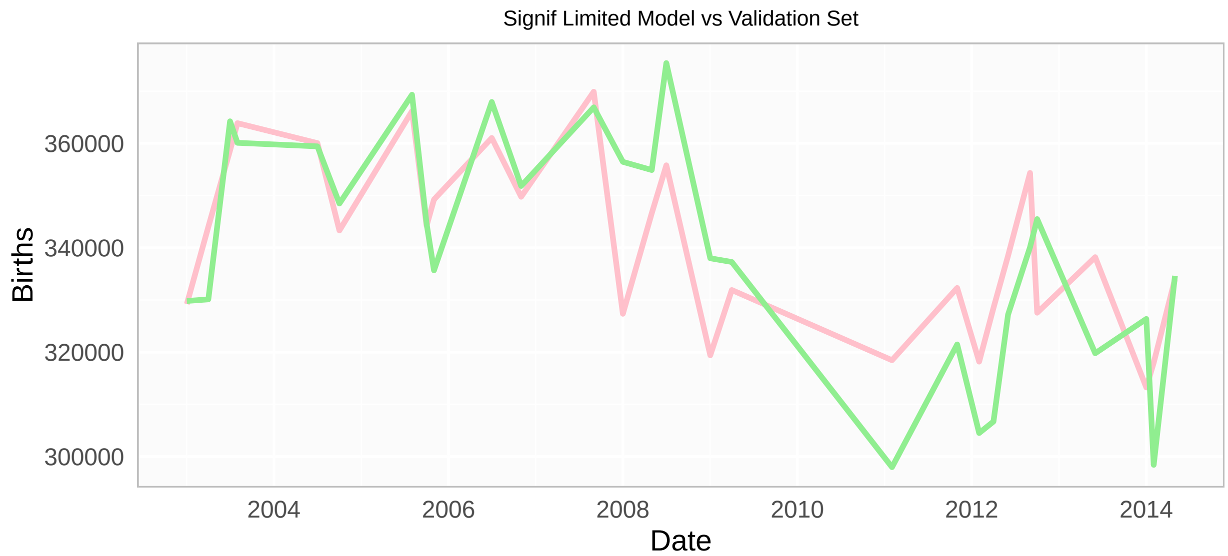
The model does not account for population growth sufficiently, but instead reduces births as more females are present in the age range of 25 - 34. On the otherhand, the unemployment rate coefficient indicates that as more people are unemployed, less pregnancies will result. For us, this is an intuitive result. Another important point is that there appears to be a change in birth activity begining during the 2007/2008 recession. This change may affect the relationship which predictor variables have with the dependent variable, *Births*.

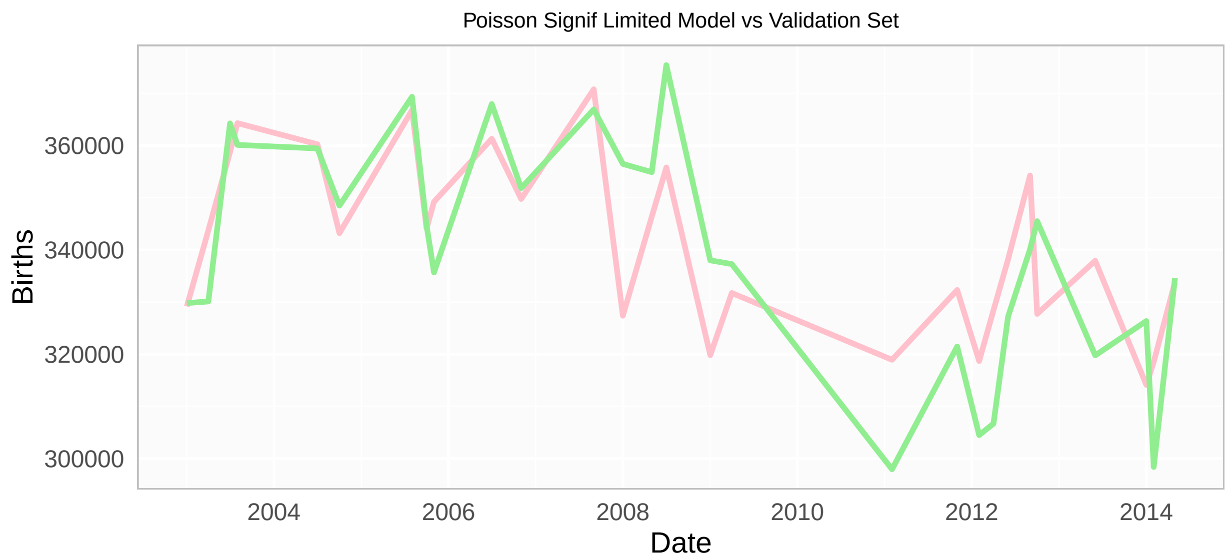
In terms of future work, we feel the earnings data set was too broad in the sense that it applied to women at all education levels and one large age range. Focusing this data set, possibly including multiple variables across various education levels and age ranges, would be an area which could yield improvement. Additionally, determining an appropriate factor for capturing population growth seems like an important next step. Our model appears to capture the United States holiday season between Thanksgiving and New Year's Day fairly well. Generalizing this somehow for culture agnostic modeling is another area for possible improvement.

7 Appendix: Supplemental Tables & Figures

7.1 Visualization: Model vs Validation Set

The following plots show visually the performance of the significant models against the validation data set.





- Green = Validation Set
- Pink = Model Prediction

8 Appendix: Raw Code

```
library(ggplot2)
library(reshape2)
```

```
library(dplyr)
library(plyr)
library(lubridate)
library(MASS)
library(pscl)

myTheme <- theme(axis.ticks=element_blank(),
                 panel.border = element_rect(color="gray", fill=NA),
                 panel.background=element_rect(fill="#FBFBFB"),
                 panel.grid.major.y=element_line(color="white", size=0.5),
                 panel.grid.major.x=element_line(color="white", size=0.5),
                 plot.title=element_text(size="8"))

rotateXaxisLabels45 <- theme(axis.text.x = element_text(angle = 45, hjust = 1))
rotateXaxisLabels90 <- theme(axis.text.x = element_text(angle = 90, hjust = 1))

dfAgeBuckets <- data.frame(lwr=c(15,25,35),
                           upr=c(25,35,45))

#
#### FUNCTION: loadBirthData ####
#
loadBirthData <- function(filename, path="./data/%s", sumByYrMonth=TRUE)
{
  # Load the Natality data
  birthFile <- sprintf(path, filename)
  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', # Age of Mother
                                       'character', # Age of Mother Code
                                       'character', # Marital Status
                                       'character', # Marital Status Code
                                       'character', # Education
                                       'character', # Education 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 <- dplyr::mutate(birthDataWoNa,
                                 Date = lubridate::parse_date_time(sprintf("%s-%s-01",
                                                                           Year.Code,
                                                                           Month.Code),
                                                                   orders="ymd"))
  # Data type conversion
  birthDataWoNa$Year.Code <- as.numeric(birthDataWoNa$Year.Code)
  birthDataWoNa$Month.Code <- as.numeric(birthDataWoNa$Month.Code)
  birthDataWoNa$Age.of.Mother <- as.factor(birthDataWoNa$Age.of.Mother)
```

```
  birthDataWoNa$Marital.Status <- as.factor(birthDataWoNa$Marital.Status)
  birthDataWoNa$Education <- as.factor(birthDataWoNa$Education)
  # Remove the extra effectively duplicated columns
  birthDataWoNa <- subset(birthDataWoNa, select = - c(Notes, Year, Month, Marital.Status.Code, Education.Co

  if(sumByYrMonth)
  {
    #by_yrmon <- group_by(birthDataWoNa, Year.Code, Month.Code, Date)
    #birthDataWoNa <- summarise(by_yrmon, sum(Births))

    birthDataWoNa <- aggregate(Births ~ Year.Code + Month.Code + Date, birthDataWoNa,
                               FUN=sum)
  }

  return (birthDataWoNa)
}

#
#### FUNCTION: loadCensusNnData ####
#
# Example:
#     df <- loadCensusNnData("US-EST00INT-01.csv")
#     summary(df)
#
loadCensusNnData <- function(filename, path)
{
  # Load the monthly totals Census data
  filepath <- sprintf(path, filename)
  df <- read.table(filepath,
                   header=TRUE,
                   sep=",",
                   fill=TRUE,
                   stringsAsFactors=FALSE)

  # Transform raw year/month columns into a Date column
  df <- dplyr::mutate(df,
                      Date = lubridate::parse_date_time(sprintf("%s-%s-01",
                                                                YEAR,
                                                                MONTH),
                                                        orders="ymd"))
  return (df)
}

#
#### FUNCTION: loadCensus00Data ####
#
# Example:
#     dfC00 <- loadCensus00Data()
#     summary(dfC00)
#     dfC00[dfC00$YEAR == 2003,]
#
loadCensus00Data <- function(path="./data/%s", verbose=TRUE, ageBuckets=dfAgeBuckets)
{
  fnMonthlyTotals <- "US-EST00INT-TOT.csv"
  fnAnnualTotals <- "US-EST00INT-ALLDATA.csv"

  # Load the monthly totals Census data
```

```
  df <- loadCensusNnData(fnMonthlyTotals, path)

  # Load the annual age/gender data
  filepath <- sprintf(path, fnAnnualTotals)
  dfAT <- read.table(filepath,
                   header=TRUE,
                   sep=",",
                   fill=TRUE,
                   stringsAsFactors=FALSE)


  dfAT999 <- dfAT[dfAT$MONTH == 7 & dfAT$AGE == 999, c("YEAR", "AGE", "TOT_POP", "TOT_FEMALE", "TOT_MALE")]
  dfAT999$GenderRatio <- dfAT999$TOT_FEMALE / (dfAT999$TOT_POP)

  # Join the gender ratio to the monthly data
  # and generate the female and male sub totals
  df <- plyr::join(df, dfAT999[,c("YEAR", "GenderRatio")])
  df$TOT_FEMALE <- df$GenderRatio * df$TOT_POP
  df$TOT_MALE <- df$TOT_POP - df$TOT_FEMALE

  # Remove April 2010, it is part of the 2010 census estimates
  df <- df[!(df$YEAR == 2010 & df$MONTH == 4), ]
  # Remove July 2010, it is part of the 2010 census estimates
  df <- df[!(df$YEAR == 2010 & df$MONTH == 7), ]

  #
  # Generate age buckets for the female population
  if(TRUE)
  {
    df7 <- dfAT[dfAT$MONTH == 7, c("YEAR", "AGE", "TOT_POP", "TOT_FEMALE", "TOT_MALE")]
    df <- generateRatioAgeBuckets(df7, df, ageBuckets, "YEAR", verbose=FALSE)
  }


  # show me
  #print(summary(dfAT))

  return (df)
}

#
#### FUNCTION: loadCensus10Data ####
#
#
# Example:
#    dfC10 <- loadCensus10Data("NC-EST2014-ALLDATA-R-File%02d.csv", 12)
#    summary(dfC10)
#   dfC10[dfC10$YEAR == 2010,]
#
loadCensus10Data <- function(filenameFmt, count, path="./data/%s", verbose=FALSE,
                             ageBuckets=dfAgeBuckets)
{
  # Load the Census data
  fn <- sprintf(filenameFmt, 1)
  if(verbose)
  {
    print(paste("Loading", fn))
```

```
  }

  df <- loadCensusNnData(fn, path)
  for(i in 2:count)
  {
    fn <- sprintf(filenameFmt, i)
    if(verbose)
    {
      print(paste("Loading", fn))
    }

    df <- rbind(df, loadCensusNnData(fn, path))
  }

  # Convert Census to regular April
  df[df$MONTH == 4.1, ]$MONTH <- 4

  # Remove April 1, 2010 estimates base
  df <- df[df$MONTH != 4.2, ]

  # Transform raw year/month columns into a Date column one final time, now that
  # we fixed up the 4.1 month
  df <- dplyr::mutate(df,
                       Date = lubridate::parse_date_time(sprintf("%s-%s-01",
                                                                 YEAR,
                                                                 MONTH),
                                                         orders="ymd"))

  # Subset to Monthly totals with subset of columns
  dfRet <- df[df$AGE == 999 ,c("MONTH", "YEAR", "TOT_POP", "TOT_FEMALE", "TOT_MALE", "Date")]

  # Generate Gender Ratio column
  dfRet$GenderRatio <- dfRet$TOT_FEMALE / (dfRet$TOT_POP)

  # Starting to bucket the fertile females count
  # ... Also might be good to bucket in smaller sets rather than one big one - DONE.
  if(TRUE)
  {
    dfRet <- generateAgeBuckets(df, dfRet, ageBuckets, "Date")
  }

  return (dfRet)
}

#
#### FUNCTION: generateAgeBuckets ####
#
generateAgeBuckets <- function(data, destData, ageBuckets, keyCol, verbose=FALSE)
{
  for(i in 1:nrow(ageBuckets))
  {
    # Pull out the # of females ages X - Y for the year
    dfFxy <- data[ageBuckets[i,]$lwr <= data$AGE & data$AGE < ageBuckets[i,]$upr, c("TOT_FEMALE", keyCol, "
    if(verbose)
    {
      print(summary(dfFxy))
    }
```

```
    dfFxySum <- aggregate(as.formula(paste("TOT_FEMALE ~ ", keyCol)), dfFxy, FUN=sum)
    newCol <- paste0("FEMALE_", ageBuckets[i,]$lwr, "_", ageBuckets[i,]$upr - 1)
    colnames(dfFxySum) <- c(keyCol, newCol)
    if(verbose)
    {
      print(summary(dfFxySum))
    }

    # Join to our result set
    destData <- plyr::join(destData, dfFxySum, by=keyCol)
  }

  return(destData)
}

generateRatioAgeBuckets <- function(data, destData, ageBuckets, keyCol, verbose=FALSE)
{
  if(verbose)
  {
    print("Src Data")
    print(head(data))
  }

  # Separate out some totals for use creating the ratios
  dfTotFm <- data[data$AGE == 999, c(keyCol, "TOT_FEMALE")]

  # Loop over the desired age buckets.
  for(i in 1:nrow(ageBuckets))
  {
    # Pull out the # of females ages X - Y for the year
    dfFxy <- data[ageBuckets[i,]$lwr <= data$AGE & data$AGE < ageBuckets[i,]$upr, c("TOT_FEMALE", keyCol, "
    if(verbose)
    {
      print("Age Rows")
      print(head(dfFxy))
    }
    # Aggregate to get age sums
    dfFxySum <- aggregate(as.formula(paste("TOT_FEMALE ~ ", keyCol)), dfFxy, FUN=sum)
    newCol <- paste0("FEMALE_", ageBuckets[i,]$lwr, "_", ageBuckets[i,]$upr - 1)
    colnames(dfFxySum) <- c(keyCol, newCol)
    if(verbose)
    {
      print("Aggregated Age Rows")
      print(head(dfFxySum))
    }
    # Join age bucket to total females so we can generate ratios
    dfFxySum <- plyr::join(dfFxySum, dfTotFm, by=keyCol)
    dfFxySum$Ratio <- dfFxySum[,newCol] / dfFxySum$TOT_FEMALE
    if(verbose)
    {
      print("Ratios with Aggregated Age Rows")
      print(head(dfFxySum))
    }
    # Compute monthly age based buckets from the ratio
    dfMonthTotFmCnt <- plyr::join(destData[,c(keyCol, "TOT_FEMALE")], dfFxySum[,c(keyCol, "Ratio")], by=key
    destData[,newCol] <- dfMonthTotFmCnt$TOT_FEMALE * dfMonthTotFmCnt$Ratio
```

```
  }

  return(destData)
}
#
#### FUNCTION: scaleCensusTotalPop ####
#
scaleCensusTotalPop <- function(data)
{
  scalar <- 1000.0
  data$TOT_POP <- data$TOT_POP / scalar
  data$TOT_FEMALE <- data$TOT_FEMALE / scalar
  data$TOT_MALE <- data$TOT_MALE / scalar

  data$FEMALE_15_24 <- data$FEMALE_15_24 / scalar
  data$FEMALE_25_34 <- data$FEMALE_25_34 / scalar
  data$FEMALE_35_44 <- data$FEMALE_35_44 / scalar

  return(data)
}

#
#### FUNCTION: loadEarningsData ####
#
# Women's weekly earnings
#
# Example:
#    df <- loadEarningsData("Earnings-2003-2015.csv")
#    summary(df)
#
loadEarningsData <- function(filename, path="./data/%s", verbose=FALSE)
{
  # Load the monthly totals Census data
  filepath <- sprintf(path, filename)
  df <- read.table(filepath,
                   header=TRUE,
                   sep=",",
                   fill=TRUE,
                   stringsAsFactors=FALSE)
  # Melt to querterly long form
  df <- reshape2::melt(df, measure.vars=c("Qtr1","Qtr2","Qtr3","Qtr4"), variable.name="Qtr")
  # Simplify to numeric
  df$Qtr  <- as.character(df$Qtr)
  df$Qtr[df$Qtr == "Qtr1"] <- 1
  df$Qtr[df$Qtr == "Qtr2"] <- 2
  df$Qtr[df$Qtr == "Qtr3"] <- 3
  df$Qtr[df$Qtr == "Qtr4"] <- 4
  df$Qtr  <- as.numeric(df$Qtr)
  colnames(df) <- c("Year", "Qtr", "Earnings")

  # Convert to wide monthly format
  df$M1 <- 0
  df$M2 <- 0
  df$M3 <- 0
  for(i in 1:4)
  {
    df$M1[df$Qtr == i] <- 1 + ((i - 1) * 3)
```

```
    df$M2[df$Qtr == i] <- 2 + ((i - 1) * 3)
    df$M3[df$Qtr == i] <- 3 + ((i - 1) * 3)
  }

  # Melt to monthly long form
  df <- reshape2::melt(df, measure.vars=c("M1","M2","M3"), variable.name="MonthVar")
  colnames(df) <- c("Year", "Qtr", "Earnings", "MonthVar", "Month")
  df <- subset(df, select=-c(MonthVar))
  df <- df[order(df$Year, df$Month), ]

  # Transform raw year/month columns into a Date column
  df <- dplyr::mutate(df,
                       Date = lubridate::parse_date_time(sprintf("%s-%s-01",
                                                                  Year,
                                                                  Month),
                                                          orders="ymd"))

  return (df)
}

#
#### FUNCTION: loadUnemploymentData ####
#
# Example:
#    df <- loadUnemploymentData("UnemploymentRate-2003-2015.csv")
#    summary(df)
#
loadUnemploymentData <- function(filename, path="./data/%s")
{
  # Load the Unemployment data
  dataFile <- sprintf(path, filename)
  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)
}


#### exploreVar Function ####
exploreVar <- function(data, varName, respName, jitter=FALSE,
```

```
                       binwidth=diff(range(data[,varName], na.rm=TRUE))/sqrt(nrow(data)),
                       na.rm=TRUE,
                       rotateLabels=FALSE,
                       xunits=NA)
{
  xlabel <- varName
  if(!is.na(xunits))
  {
    xlabel <- paste(varName, "(", xunits, ")")
  }

  varStats <- data.frame(min=min(data[,varName], na.rm=na.rm),
                         mean=mean(data[,varName], na.rm=na.rm),
                         stdev=sd(data[,varName], na.rm=na.rm),
                         median=median(data[,varName], na.rm=na.rm),
                         max=max(data[,varName], na.rm=na.rm))

  g1 <- ggplot(data) +
    geom_histogram(aes_string(x=varName), binwidth = binwidth) +
    labs(title=paste("Distribution of\n", varName, "Variable"),
         x=xlabel) +
    myTheme

  position <- "identity"
  if(jitter)
  {
    position <- "jitter"
  }

  g2 <- ggplot(data) +
    geom_point(aes_string(x=varName, y=respName), alpha=0.4, position=position) +
    labs(title=paste("Scatter plot of\n", varName, "vs", respName),
         x=xlabel) +
    myTheme

  g3 <- ggplot(data) +
    geom_boxplot(aes_string(rep(varName, nrow(data)), varName)) +
    labs(title=paste("Box plot of\n", varName), x="", y="") +
    myTheme

  if(rotateLabels)
  {
    g1 <- g1 + rotateXaxisLabels45
    g2 <- g2 + rotateXaxisLabels45
    g3 <- g3 + rotateXaxisLabels45
  }


  return (list(varStats, g1, g2, g3))
}

#### coefficientsPrep Function ####
coefficientsPrep <- function(smlm)
{
  coef <- smlm$coefficients[,c(1,4)]
  rownames(coef) <- c("Intercept", rownames(coef)[2:nrow(coef)])
```

```
  coefSigNdx <- (coef[,2] < 0.05)
  if(length(coefSigNdx) > 0)
  {
    rownames(coef)[coefSigNdx] <- paste(rownames(coef)[coefSigNdx], "*")
  }

  return(coef)
}

crossValidate <- function(model, cvdata, responseCol, bPrintSummary)
{
  cvPredict <- predict(model, newdata=cvdata)
  #head(cvPredict)
  cvCombined <- cbind(cvdata, cvPredict)
  cvCombined$PredictError <- cvCombined$cvPredict - cvCombined[,responseCol]
  cvCombined$SqE <- cvCombined$PredictError^2
  MSE <- mean(cvCombined$SqE, na.rm=TRUE)

  if(bPrintSummary)
  {
    print(summary(cvCombined[,c(responseCol, "cvPredict", "PredictError", "SqE")]))
  }

  return(MSE)
}

crossValidateGLM <- function(model, cvdata, responseCol, bPrintSummary)
{
  cvPredict <- predict(model, newdata=cvdata, type="response")
  #head(cvPredict)
  cvCombined <- cbind(cvdata, cvPredict)
  cvCombined$PredictError <- cvCombined$cvPredict - cvCombined[,responseCol]
  cvCombined$SqE <- cvCombined$PredictError^2
  MSE <- mean(cvCombined$SqE, na.rm=TRUE)

  if(bPrintSummary)
  {
    print(summary(cvCombined[,c(classCol, "cvPredict")]))
  }

  return(MSE)
}

mse <- function(sm) {
  mse <- mean(sm$residuals^2)
  return(mse)
}

library(ggplot2)
library(plyr)
set.seed(020275)
### Load the data helper functions
dlfFile <- "./Project/NatalityModels-DataLoadFuncs.R"
if(!file.exists(dlfFile))
{
  dlfFile <- paste0("../", dlfFile)
}
```

```
source(dlfFile)

### Daniel's Raw Code Sandbox...
dataPath <- "./data/"
if(!dir.exists(dataPath))
{
  dataPath <- paste0("../", dataPath)
}
dataPath <- paste0(dataPath, "%s")

# Load the 2007-2014 birth data
birthData <- loadBirthData("Natality, 2007-2014.txt", path=dataPath)
#summary(birthData)

# Load the 2003-2006 birth data
birthData2 <- loadBirthData("Natality, 2003-2006.txt", path=dataPath)
#summary(birthData2)

# Combine the 2 sets of birth data
allBirthData <- rbind(birthData, birthData2)
#summary(allBirthData)

# Load 2000s census data
censusData0010 <- loadCensus00Data(path=dataPath)
censusData0010 <- scaleCensusTotalPop(censusData0010)
#summary(censusData0010)

# Load 2010s census data
censusData1015 <- loadCensus10Data("NC-EST2014-ALLDATA-R-File%02d.csv", 12, path=dataPath)
censusData1015 <- scaleCensusTotalPop(censusData1015)
#summary(censusData1015)

# Combine the 2000-2010 census with the 2010-2015 estimates
allCensusData <- rbind(censusData0010, censusData1015)

gGenderRatio <- ggplot(allCensusData) +
  geom_line(aes(x=Date, y=GenderRatio)) +
  labs(title="Gender Ratio 2000 - 2015") +
  myTheme
#gGenderRatio

gFmPop <- ggplot(allCensusData) +
  geom_line(aes(x=Date, y=TOT_FEMALE)) +
  labs(title="Female Population 2000 - 2015") +
  myTheme
#gFmPop

# Load the women's earnings data
earningsData <- loadEarningsData("Earnings-2003-2015.csv", path=dataPath)
gEarnings <- ggplot(earningsData) +
  geom_line(aes(x=Date, y=Earnings)) +
  labs(title="Women's Weekly Earnings 2003 - 2015") +
  myTheme
#gEarnings

# Load unemployment rate
urateData <- loadUnemploymentData("UnemploymentRate-2003-2015.csv", path=dataPath)
```

```
gUnemployment <- ggplot(urateData) +
  geom_line(aes(x=Date, y=UnemploymentRate))  +
  labs(title="Unemployment Rate 2003 - 2015") +
  myTheme
#gUnemployment


# Combine all together
allData <- plyr::join(allBirthData, allCensusData, by="Date")
allData <- plyr::join(allData, earningsData, by="Date")
allData <- plyr::join(allData, urateData, by="Date")
allData$Month9Ago <- month(allData$Date - months(9))
allData <- allData[order(allData$Date),]
allData <- allData[,c("Year",
                      "Month",
#                       "Age.of.Mother",
#                       "Age.of.Mother.Code",
#                       "Marital.Status",
#                       "Education",
                      "Births",
                      "Date",
                      "TOT_POP",
                      "GenderRatio",
                      "TOT_FEMALE",
                      "TOT_MALE",
                      #"FEMALE_15_30",
                      #"FEMALE_30_50",
                      "FEMALE_15_24",
                      "FEMALE_25_34",
                      "FEMALE_35_44",
                      "Earnings",
                      "UnemploymentRate",
                      "Month9Ago")]
#summary(allData)
# Row and column counts
ncolAllData <- ncol(allData)
nrowAllData <- nrow(allData)

#allData$moIndex <- month.abb[allData$Month]

# Data Exploration
missingVals <- sapply(allData, function(x) sum(is.na(x)))
missingValsPerc <- sapply(allData, function(x) sum(is.na(x))/length(x)*100)
dfMissingVals <- data.frame(Missing=missingVals, Percent=missingValsPerc)
#summary(missingVals)

# Correlation
colsForCor <- c("Year", "Month",  "Births", "TOT_POP", "GenderRatio", "TOT_FEMALE",
                "TOT_MALE", "FEMALE_15_24","FEMALE_25_34",
                "FEMALE_35_44", "Earnings", "UnemploymentRate")
corMatrix <- cor(allData[,colsForCor], use="complete.obs")


# Subset all data into training and validation data sets
cvSample <- sample(nrow(allData),  nrow(allData) * 0.20)
# Validation set
crossValData <- allData[cvSample,]
```

```
#crossValData <- fillInMissingWithMedian(crossValData, FALSE)
# Model set is subset to only complete cases.
modelData <- allData[-cvSample,]
completeModelData <- modelData[complete.cases(modelData), ]


# Initial linear model with all variables
lmAllVars <- lm(Births ~ Month + . - Year - Date, data=modelData)
smLmAllVars <- summary(lmAllVars)
vfAllVars <- faraway::vif(lmAllVars)

pdAllVars <- predict(lmAllVars, se.fit=TRUE)
pdModelData <- cbind(modelData, model=pdAllVars$fit)
gPdAllVars <- ggplot(pdModelData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme
#gPdAllVars

# Significant variables model
lmSigVars <- lm(Births ~ TOT_POP + GenderRatio + TOT_FEMALE + FEMALE_15_24 + FEMALE_25_34 + FEMALE_35_44 +
                data=modelData)
smLmSigVars <- summary(lmSigVars)
vfSigVars <- faraway::vif(lmSigVars)

pdSigVars <- predict(lmSigVars, se.fit=TRUE)
pdModelData <- cbind(modelData, model=pdSigVars$fit)
gPdSigVars <- ggplot(pdModelData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme
#gPdSigVars

# High Cor variables model
lmHighCorVars <- lm(Births ~ FEMALE_25_34 + UnemploymentRate + FEMALE_35_44 + Earnings + Month + TOT_FEMALE
                data=modelData)
smLmHighCorVars <- summary(lmHighCorVars)
vfHighCorVars <- faraway::vif(lmHighCorVars)

pdHighCorVars <- predict(lmHighCorVars, se.fit=TRUE)
pdModelData <- cbind(modelData, model=pdHighCorVars$fit)
gPdHighCorVars <- ggplot(pdModelData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme
#gPdHighCorVars

# Step model
lmStep <- step(lmAllVars, trace=0)
smLmStep <- summary(lmStep)
vfStep <- faraway::vif(lmStep)

pdStepVars <- predict(lmStep, se.fit=TRUE)
pdModelData <- cbind(modelData, model=pdStepVars$fit)
gPdStep <- ggplot(pdModelData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme +
  labs(title="Step Model")
#gPdStep
#smLmStep
```

```
# Signif-HighVIFS variables model
lmSigVifVars <- lm(Births ~ Month + GenderRatio + FEMALE_25_34 + FEMALE_35_44 + Earnings,
                   data=modelData)
smLmSigVifVars <- summary(lmSigVifVars)
vfSigVifVars <- faraway::vif(lmSigVifVars)

pdSigVifVars <- predict(lmSigVifVars, se.fit=TRUE)
pdModelData <- cbind(modelData, model=pdSigVifVars$fit)
gPdSigVifVars <- ggplot(pdModelData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme +
  labs(title="Signif Vars - High VIFs Model")
#gPdSigVifVars
#smLmSigVifVars

# Signif-Limited variables model
lmSigLimVars <- lm(Births ~ Month + Month9Ago + FEMALE_25_34 + UnemploymentRate,
                   data=modelData)
smLmSigLimVars <- summary(lmSigLimVars)
vfSigLimVars <- faraway::vif(lmSigLimVars)

pdSigLimVars <- predict(lmSigLimVars, se.fit=TRUE)
pdModelData <- cbind(modelData, model=pdSigLimVars$fit)
gPdSigLimVars <- ggplot(pdModelData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme +
  labs(title="Signif Limited Model")
#gPdSigLimVars
#smLmSigLimVars
#vfSigLimVars

# Signif-Limited with Interaction variables model
lmSigLimInterVars <- lm(Births ~ Month + Month9Ago + FEMALE_25_34 + UnemploymentRate + Month9Ago:FEMALE_25_
                   data=modelData)
smLmSigLimInterVars <- summary(lmSigLimInterVars)
vfSigLimInterVars <- faraway::vif(lmSigLimInterVars)


# Poison Count Regression
pmSigLimVars <- glm(Births ~ Month + Month9Ago + FEMALE_25_34 + UnemploymentRate,
                    family=poisson, modelData)
smPmSigLimVars <- summary(pmSigLimVars)
vifPmSigLimVars <- faraway::vif(pmSigLimVars)

pdPmSigLimVars <- predict(pmSigLimVars, type="response")
pdModelData <- cbind(modelData, model=pdPmSigLimVars)
gPdSigLimVars <- ggplot(pdModelData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme +
  labs(title="Signif Limited Count Model")

# Poisson Regression via StepAIC
# Poison Count Regression
pmAllVars <- glm(Births ~ Month + . - Year - Date,
                    family=poisson, modelData)
```

```r
smpmAllVars <- summary(pmAllVars)

pmStep <- stepAIC(pmAllVars, direction="backward", trace=0)
smPmStep <- summary(pmStep)
#smPmStep

pmStepAICSuggested <- glm(Births ~ Month + TOT_POP + GenderRatio + TOT_FEMALE + FEMALE_15_24 +
                            FEMALE_25_34 + FEMALE_35_44 + Earnings + UnemploymentRate +
                            Month9Ago, family=poisson, modelData)

smPmStepAICSuggested <-summary(pmStepAICSuggested)
#smPmStepAICSuggested
vifPmStep <- faraway::vif(pmStepAICSuggested)

## Negative Binomial Models
nbm <- glm.nb(Births ~ Month + TOT_POP + GenderRatio + TOT_FEMALE + FEMALE_15_24 +
                 FEMALE_25_34 + FEMALE_35_44 + Earnings + UnemploymentRate +
                 Month9Ago, data=modelData)
smNbm <- summary(nbm)
vifNbm <- faraway::vif(nbm)

stepNbm <- stepAIC(nbm, direction="backward", trace=0)
#stepNbm$anova


nbmStepAICSuggested <- glm.nb(Births ~ Month + TOT_POP + GenderRatio + FEMALE_25_34 + UnemploymentRate +
                                Month9Ago, data=modelData)
vifStepNbm <- faraway::vif(nbmStepAICSuggested)
smNbmStepAIC <- summary(nbmStepAICSuggested)

# AR
#arModel <- ar(ts(modelData), method="burg")
#arModel

# Validation
showSummary <- FALSE
responseCol <- "Births"
cvAllLm <- crossValidate(lmAllVars, crossValData, responseCol, showSummary)
cvSignifLm <- crossValidate(lmSigVars, crossValData, responseCol, showSummary)
cvHighCorLm <- crossValidate(lmHighCorVars, crossValData, responseCol, showSummary)
cvStep <- crossValidate(lmStep, crossValData, responseCol, showSummary)
cvSigMinus <- crossValidate(lmSigVifVars, crossValData, responseCol, showSummary)
cvSigLim <- crossValidate(lmSigLimVars, crossValData, responseCol, showSummary)
cvPmSigLim <- crossValidateGLM(pmSigLimVars, crossValData, responseCol, showSummary)
cmPmStep <- crossValidateGLM(pmStepAICSuggested, crossValData, responseCol, showSummary)
cvNbm <- crossValidateGLM(nbm, crossValData, responseCol, showSummary)
cvSigLimInterVars <- crossValidate(lmSigLimInterVars, crossValData, responseCol, showSummary)

#
#
#
cvLmResults <- data.frame(Model=c("All Variables",
                                  "Significant",
                                  "High Cor",
                                  "Step",
                                  "Significant Minus",
                                  "Significant Limited",
```

```
                                "Poisson Signif Ltd",
                                "Poisson Step",
                                "Neg Binomial Step",
                                "Signif Ltd w/ Interaction"),
                    Val.Error=c(cvAllLm,
                                cvSignifLm,
                                cvHighCorLm,
                                cvStep,
                                cvSigMinus,
                                cvSigLim,
                                cvPmSigLim,
                                cmPmStep,
                                cvNbm,
                                cvSigLimInterVars),
                    R2=c(smLmAllVars$adj.r.squared,
                         smLmSigVars$adj.r.squared,
                         smLmHighCorVars$adj.r.squared,
                         smLmStep$adj.r.squared,
                         smLmSigVifVars$adj.r.squared,
                         smLmSigLimVars$adj.r.squared,
                         NA,
                         NA,
                         NA,
                         smLmSigLimInterVars$adj.r.squared),
                    AIC=c(AIC(lmAllVars),
                          AIC(lmSigVars),
                          AIC(lmHighCorVars),
                          AIC(lmStep),
                          AIC(lmSigVifVars),
                          AIC(lmSigLimVars),
                          pmSigLimVars$aic,
                          pmStep$aic,
                          nbm$aic,
                          AIC(lmSigLimInterVars)),
                    Variables=c(length(lmAllVars$coefficients) - 1,
                                length(lmSigVars$coefficients) - 1,
                                length(lmHighCorVars$coefficients) - 1,
                                length(lmStep$coefficients) - 1,
                                length(lmSigVifVars$coefficients) - 1,
                                length(lmSigLimVars$coefficients) - 1,
                                length(pmSigLimVars$coefficients) - 1,
                                length(pmStep$coefficients) - 1,
                                length(nbm$coefficients) - 1,
                                length(lmSigLimInterVars$coefficients) - 1),
                    VIF=c("BAD", "BAD","BAD", "BAD", "BAD", "OK", "OK", "BAD", "BAD", "BAD"))

# Significant Limited Model
pdSigLimVarsCV <- predict(lmSigLimVars, se.fit=TRUE, newdata=crossValData)
pdCVData <- cbind(crossValData, model=pdSigLimVarsCV$fit)

gPdSigLimVarsCV <- ggplot(pdCVData) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme +
  labs(title="Signif Limited Model vs Validation Set", y="Births")
#gPdSigLimVarsCV

# Poisson Significant Limited Model
```

```
pdPmSigLimVarsAll <- predict(pmSigLimVars, type="response", newdata=allData)
pdAllDataPm <- cbind(allData, model=pdPmSigLimVarsAll)

gPdPmSigLimVarsAll <- ggplot(pdAllDataPm) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme +
  labs(title="Signif Limited Model vs Full Data Set", y="Births")


# Poisson Significant Limited Model
pdPmSigLimVarsCV <- predict(pmSigLimVars, type="response", newdata=crossValData)
pdCVDataPm <- cbind(crossValData, model=pdPmSigLimVarsCV)

gPdPmSigLimVarsCV <- ggplot(pdCVDataPm) +
  geom_line(aes(x=Date, y=model), colour="pink", size=1) +
  geom_line(aes(x=Date, y=Births), colour="lightgreen", size=1) + myTheme +
  labs(title="Poisson Signif Limited Model vs Validation Set", y="Births")


#library(leaps)
#lmSubsCdc <- leaps::regsubsets(Births ~ Month.Code + Age.of.Mother + Marital.Status + Education, data=allB
#summary(lmSubsCdc)

#step(lmCdc)
```

9 References

Amialchuk, A. "Relative Wage Changes and Fertility in the US". In: Eastern Economic Journal 39(2) (2013). DOI: 10.1057/eej.2013.2.

Bureau of Labor Statistics. Labor Force Statistics from the Current Population Survey 2003-2015 - LNS14000000. Accessed: April 24, 2016. 2015. URL: http://data.bls.gov/timeseries/LNS14000000.

— Median wkly earnings, Emp FT, Wage & sal wrkrs, Women - LEU0252882700. Accessed: March 10, 2016. 2015. URL: http://data.bls.gov/cgi-bin/surveymost?le.

Census Bureau. Monthly Postcensal Resident Population, by single year of age, sex, race, and Hispanic origin. Accessed: April 24, 2016. 2015. URL: http://www.census.gov/popest/data/national/asrh/2014/2014-nat-res.html.

— National Intercensal Estimates (2000-2010). Accessed: April 24, 2016. 2010. URL: http://www.census.gov/popest/data/intercens

Centers for Disease Control and Prevention. Natality public-use data on CDC WONDER Online Database for years 2003-2006 available March 2009. Accessed: March 1, 2016. 2009. URL: http://wonder.cdc.gov/natality-v2006.html.

— Natality public-use data on CDC WONDER Online Database for years 2007-2014 available February 2016. Accessed: March 1, 2016. 2016. URL: http://wonder.cdc.gov/natality-current.html.

Dittenhafer, D. U.S. Births & Unemployment Rate 2007 - 2012. 2014. URL: https://github.com/dwdii/DataAcqMgmt/raw/master/I
Dittenhafer-USBirthsAnalysis.pdf.

Dittenhafer, D. and J. Hink. NatalityModels. 2016. URL: https://github.com/dwdii/NatalityModels.

Morgan, S. and M. Taylor. Low Fertility at the Turn of the Twenty-First Century. Aug. 2006. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2849172/.