Data Analytics II

Group project

Thomas Lopez
Rosa Caminal
David Herbert
Lucas Heral
Magdeleine Courtois

OBAMA-CLINTON DATASET

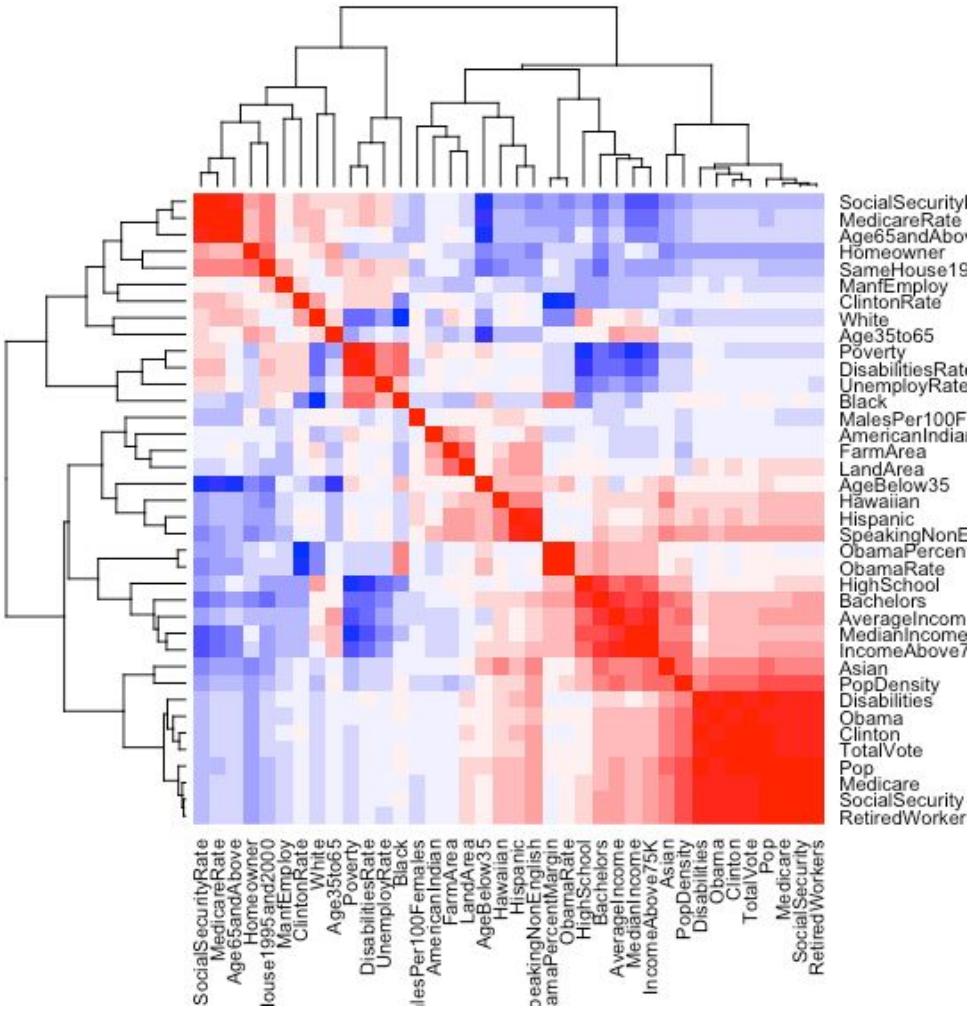
THE PROBLEM:

The Obama-Clinton 2008 Elections' Dataset addresses the problem of finding the characteristics that may impact voting patterns and predictions. The specific problem that we will focus on is concerning the states that are still yet to vote – specifically analysing Obama's likelihood to win in those states. Our target attribute will be Obama's rate of vote. We can derive this from the TotalVote and Obama variables by computing: (100*Obama/TotalVote). We will use regression analysis to identify the most important attributes that may impact Obama voters' patterns. Through a thorough analysis of these attributes, our aim is to predict the final vote outcome for Obama at the end of these elections.

UNDERSTANDING THE DATA:

By performing a simple dimension sort on R, we see that the size of our dataset is formed by 2868 rows and 41 columns. We have 1737 observations of "known" data and 1131 of "unknown" data. The dataset is based on the U.S. census regarding demographic data for each county, combined with partial election results.

Based on the heatmap below, we identified attributes likely to be relevant to our problem – mostly consisting of demographic data - such as race, income variables and academic experience. Positively correlated variables are in red while negatively correlated are in blue:

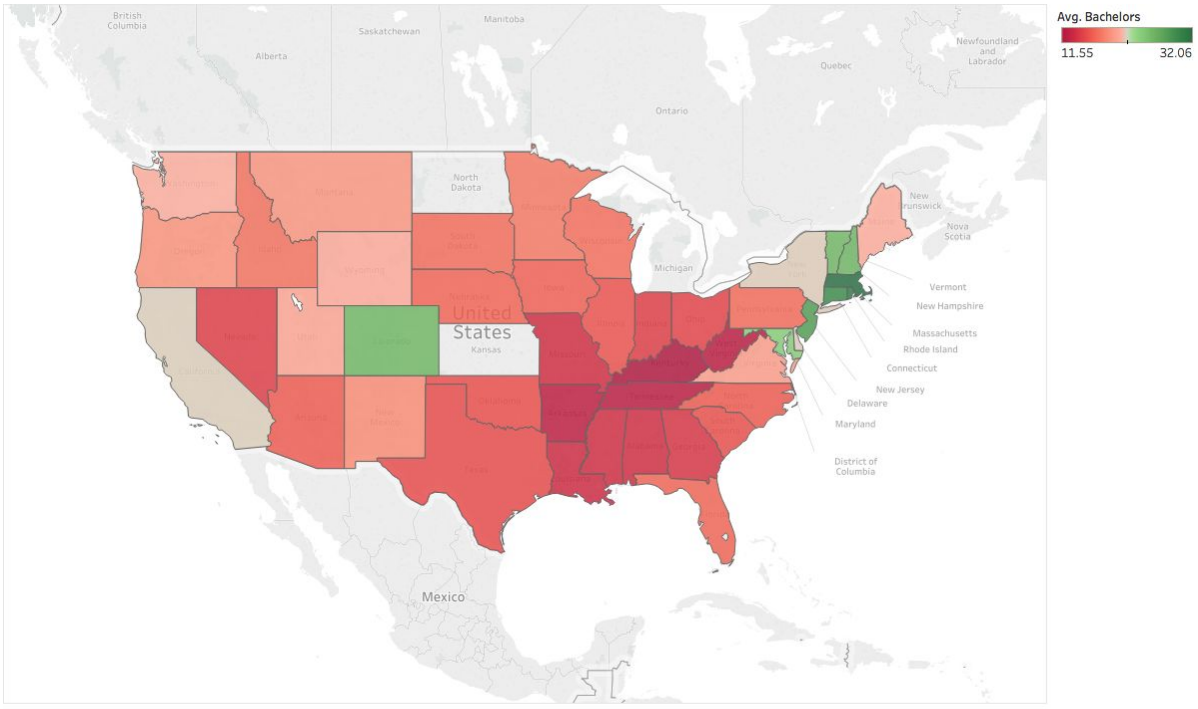


Therefore, we computed the highest positive correlation values for our target variable ObamaRate:

Black	0.447173666604092
Bachelors	0.365941362947178
IncomeAbove75K	0.302365527607841
HighSchool	0.269964657407263
MedianIncome	0.249561645444397

- **Tableau:**

To explore more those correlations, we started to compare ObamaRate within states with Bachelors attribute. We looked at the average of bachelors on the different states. Some of states with a low ObamaRate coincidentally had a lower average of bachelors degrees. However, it is also clear by just looking at the map that other states with low average of bachelors degrees in the South had a high ObamaRate.



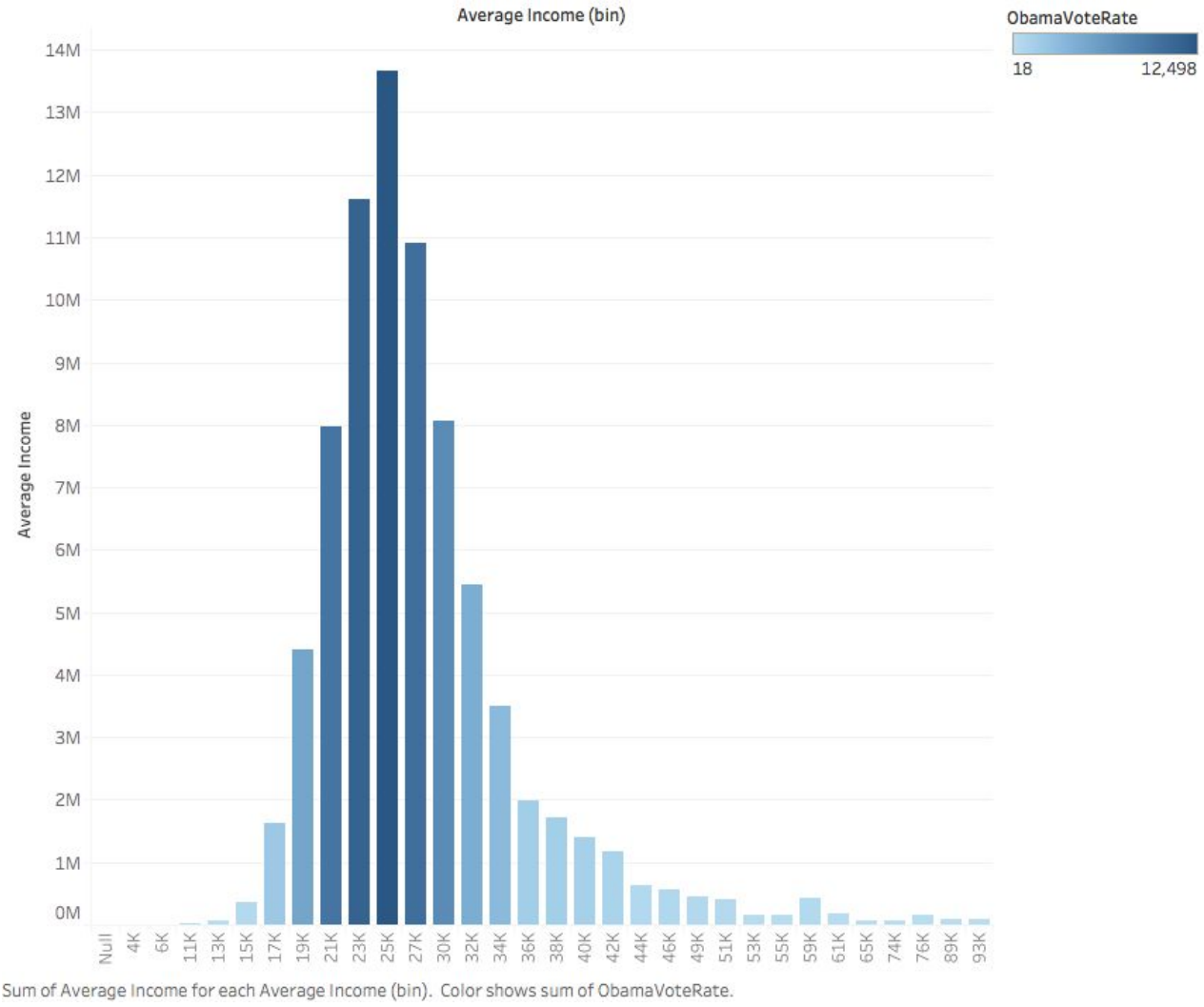
Map based on Longitude (generated) and Latitude (generated). Color shows average of Bachelors. Details are shown for State. The data is filtered on sum of Median Income, which keeps all values.

To further explore this idea, we decided to scatterplot the ObamaRate against the average of bachelors coloured by Region.

Average of ObamaVoteRate vs. average of Bachelors. Color shows details about Region. Details are shown for County.

As can be seen with the trend lines, the higher the bachelors average the higher the ObamaRate per county in all of the Regions. This is especially true in the West were the correlation was the highest.

We then explored the average income of voters. By plotting the average income distribution by county, we found that counties with an average income between 21k-30k have a distinctly higher Obama Vote Rate.



This could help indicate that the low-middle class as according to Pew's definition (Hoffower,2018) of Middle-class Americans form the majority of Obama's voters.

- **R:**

We computed an aggregation table in R, showing some representative values such as the mean Bachelors value in the counties in which each candidate wins:

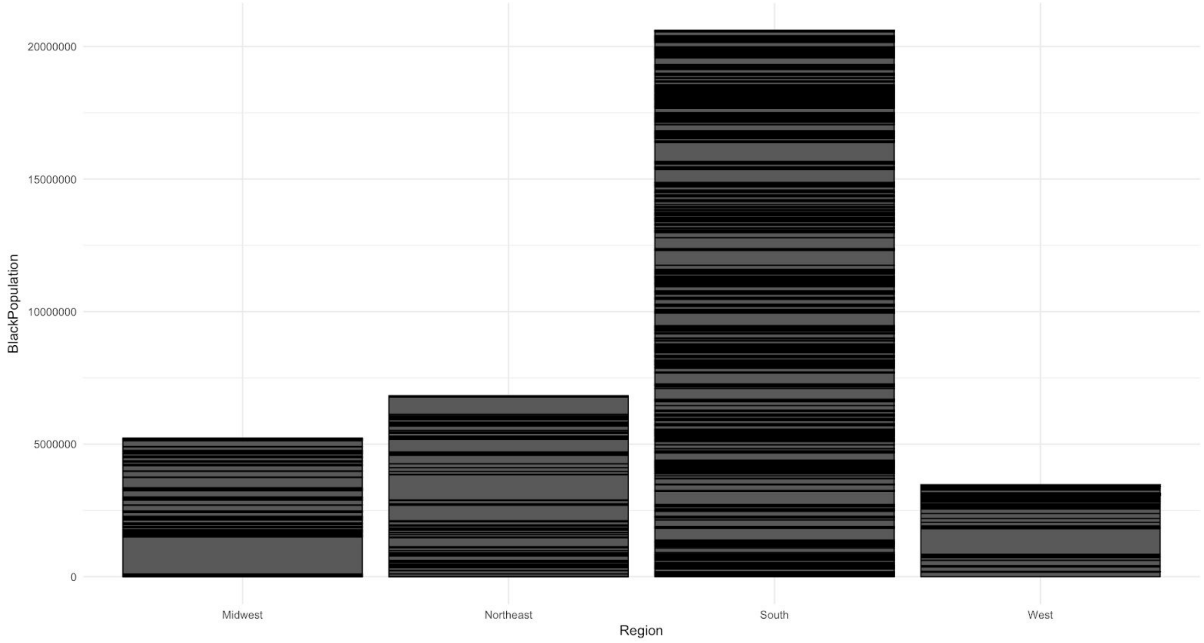
Winner	Bachelors	IncomeAbove75K	HighSchool	MedianIncome	Poverty
Clinton	14.79014	12.82738	75.91216	37955.26	13.89058
Obama	19.51446	16.50535	79.30109	42290.66	13.00498

We can clearly see that for those attributes, the mean is higher where Obama wins than where Clinton wins. This highlights the relevance of these attributes for Obama specifically.

The following scatterplot shows the different counties of the US with their rate for Obama's vote against average income of that specific county. We have also fitted a regression line on the plot:

This showed that the larger the average income in a county, the higher the rate of Obama's vote. Moreover, it showed a very interesting vote distribution among the Southern counties – a very polarised vote, showing both the lowest and highest rates.

To explain this result, we looked at the impact of the most correlated attribute, that is Black, with ObamaRate. As you can see on the plot below, the populations of black minorities is higher in the Southern region than all others:



DATA PREPARATION PROCESS:

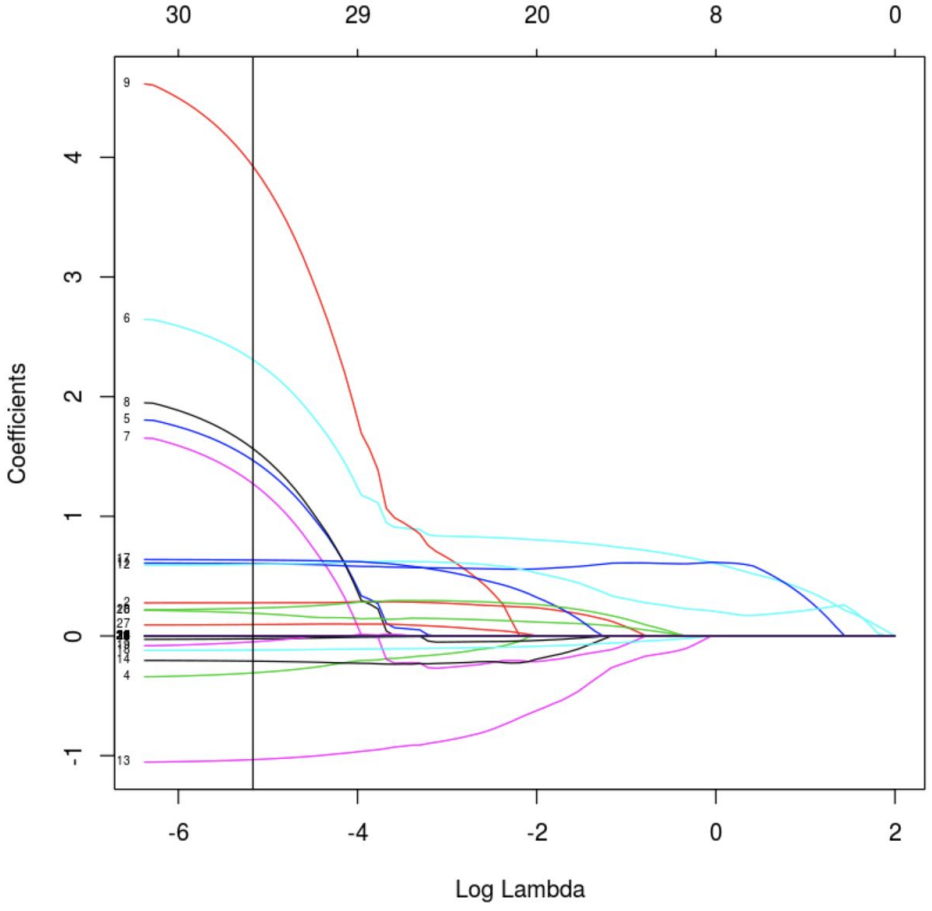
- Loading and Preparing the data: We load the data into R via the internet using Github's content portal
- Derived Attributes: Our entire project concerned looking at the ObamaRate
- Inputting Missing Data: This section was done in seperate parts depending on the attribute we were dealing with. As AverageIncome was roughly similar to MedianIncome, we replaced missing values in AverageIncome by their MedianIncome entry and then utilised this attribute predominantly. Thereafter, we replaced all other attributes that were missing values with a "0". Lastly, for any other missing values found we simply deleted the relevant entries.
- Converting Dates: We formatted the ElectioDate attribute into the correct "date" datatype.
- Splitting dataset: we had to split the data into two seperate sets, one with "known" vote data and another with "unknown" vote data.
- Test and training sets: Thereafter, we split the "known" vote data into a training and test set. We used an indice value of 80% meaning the training data consisted on 80% of our "known" vote data and the test data consisted on 20% of our "known" data set.

GENERATING AND TESTING PREDICTION MODELS:

We looked at regression models in order to predict our target variable - ObamaRate. We used the RMSE in order to rank our analysis.

Firstly, we executed some basic linear regression models, as well as Backward and Forward Stepwise models, iterating through different variables. This allowed us to learn more about the attributes. Our findings indicated that the Lasso Regularization with lambda optimisation model achieved the lowest RMSE.

Lasso Regularisation (Optimised Lambda):

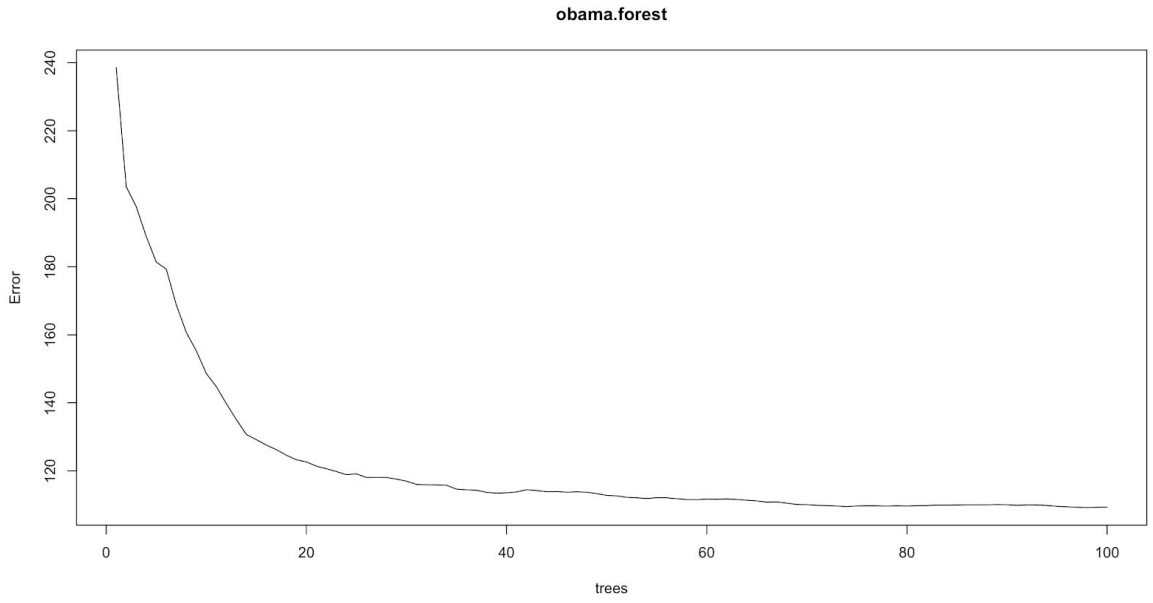



```
33 x 1 sparse Matrix of class "dgCMatrix"
                                  1
(Intercept)           -1.693989e+02
MalesPer100Females    -2.356838e-02
AgeBelow35             2.774744e-01
Age35to65              .
Age65andAbove         -3.085672e-01
White                  1.467226e+00
Black                  2.308267e+00
Asian                  1.272183e+00
AmericanIndian         1.565548e+00
Hawaiian               3.924599e+00
Hispanic               1.899399e-01
HighSchool             6.036524e-01
Bachelors              5.978114e-01
Poverty               -1.033419e+00
IncomeAbove75K        -2.102622e-01
MedianIncome          -2.156562e-04
AverageIncome          8.098588e-05
UnemployRate           6.347873e-01
ManfEmploy            -1.180310e-01
SpeakingNonEnglish    -4.896825e-02
Medicare               1.391579e-04
MedicareRate           2.167344e-04
SocialSecurity        -3.020654e-04
SocialSecurityRate    -2.501727e-04
RetiredWorkers         .
Disabilities           3.321417e-05
DisabilitiesRate      -1.019683e-03
Homeowner              9.496062e-02
SameHouse1995and2000   2.325263e-01
Pop                    1.639882e-05
PopDensity            -2.550717e-04
LandArea               1.322232e-03
FarmArea              -7.993659e-04
```

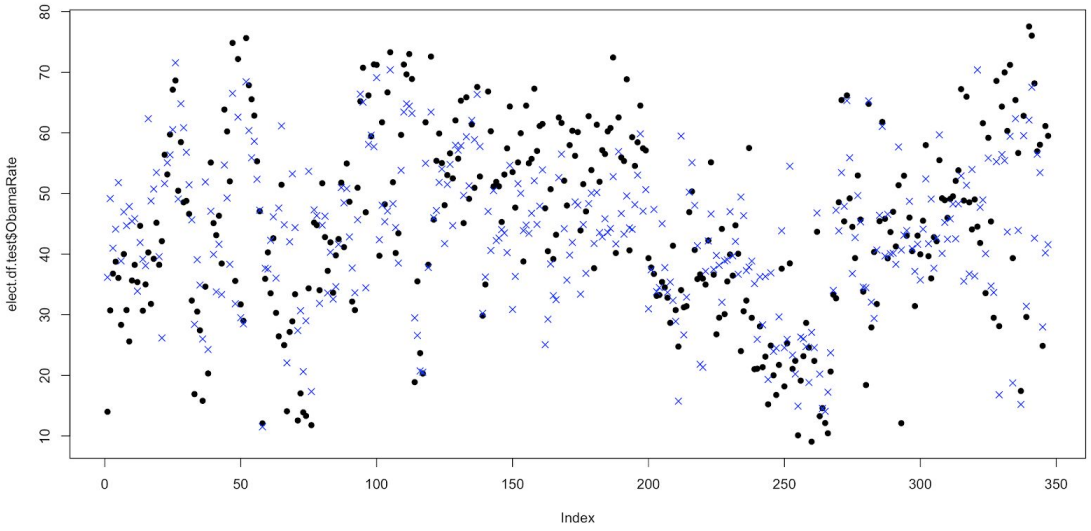
In this more sophisticated linear model, we started to see more variables contributing to our target. Variables such as ‘AverageIncome’, ‘HomeOwner’, ‘HighSchool’, ‘Bacherlors’ and ‘Black’ appear – contributing highly to our target variable prediction. This final output achieved a MAE of 8.929 and a RMSE of 11.14.

Random Forest

Running a Random Forest allowed us to achieve an even lower RMSE, concretely of 9.15224. This model took into account all variables.

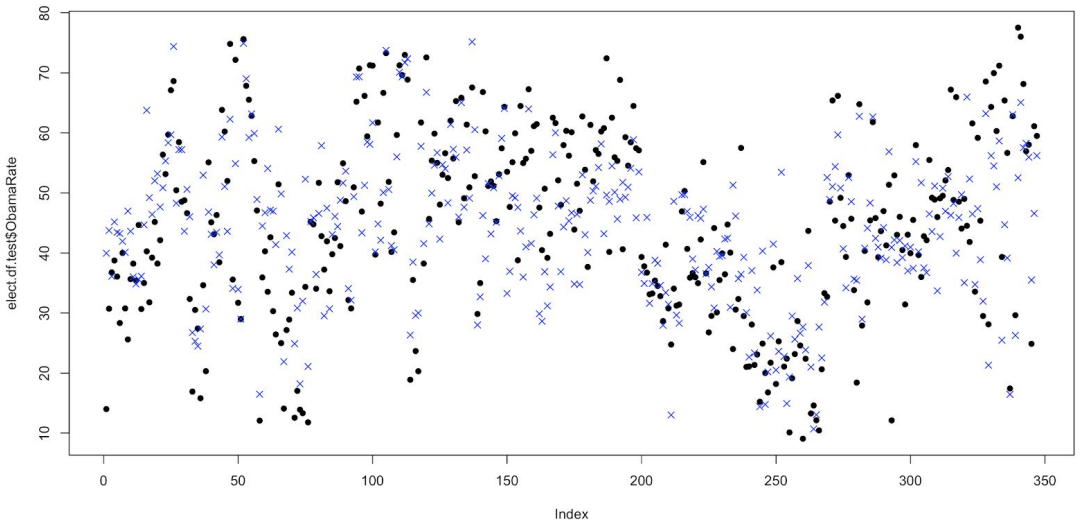


The black dots on the graph represent the actual points of data of our target variable, while the blue crosses are the predictions of the model. As we can see, the prediction is relatively accurate:



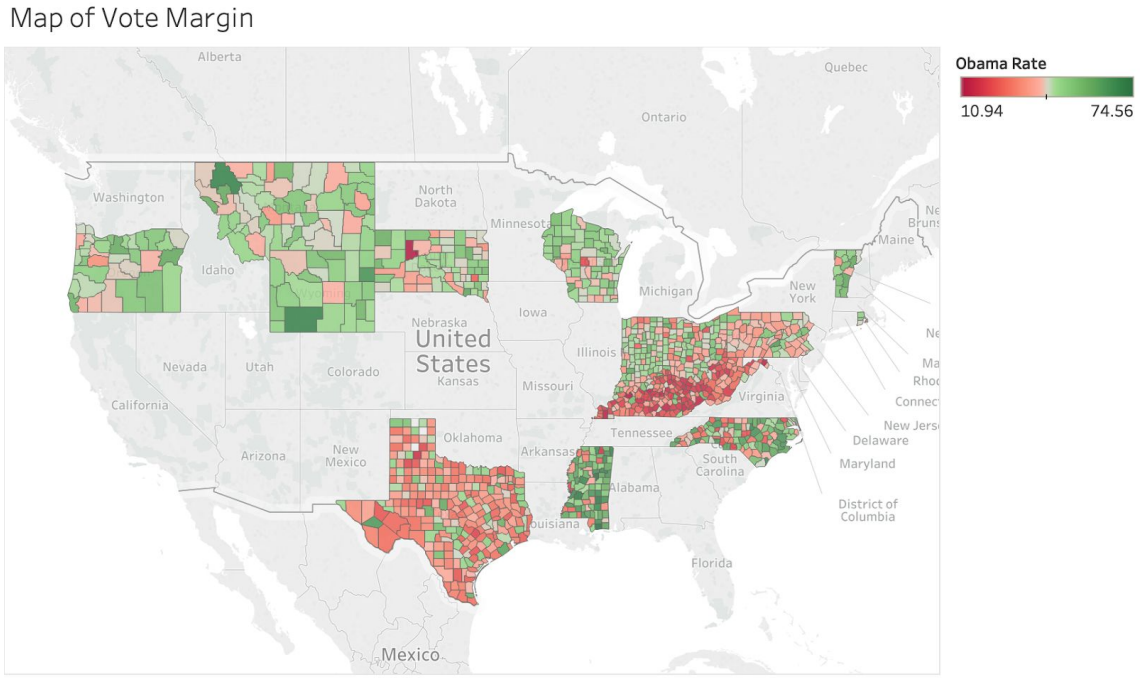
Support Vector Machine

To finalise our prediction models, we executed a SVM – given the complexity of this algorithm, we expected it to return the best prediction results. Although not far from the Random Forest's RMSE, it was still a bit higher – 9.5216. Yet, when decreasing the number of variables the error-rate increased slightly.



CONCLUSIONS AND RECOMMENDATIONS:

Using our most accurate model, we predicted the election results in the states left to vote:



Mapa basado en Longitud (generado) y Latitud (generado). El color muestra suma de Obama Rate. Se muestran detalles para State1 y County1.

On average, Obama gets an average of 40.47% in the remaining counties.

We do not have a TotalVote variable available for our unknown dataset – considering the average participation rate from the counties that already voted (TotalVote*100/Pop = 10.8%), and applying this percentage over the Population, these votes resulted being 5,432,978 for Clinton and 3,847,834 for Obama.

In the counties voted, as we know from the 'known' dataset, Obama wins 10,730,582 vs. 10,375,963. With the predicted votes from our model, and adding them to the known dataset, Obama loses the elections for 1,230,525 votes.

Features that seem to impact Obama voting rates most is the voters' income, educational level and race. In fact, people having a high income and higher education tends to vote for Obama. Additionally, black minorities tend to vote more for Obama.

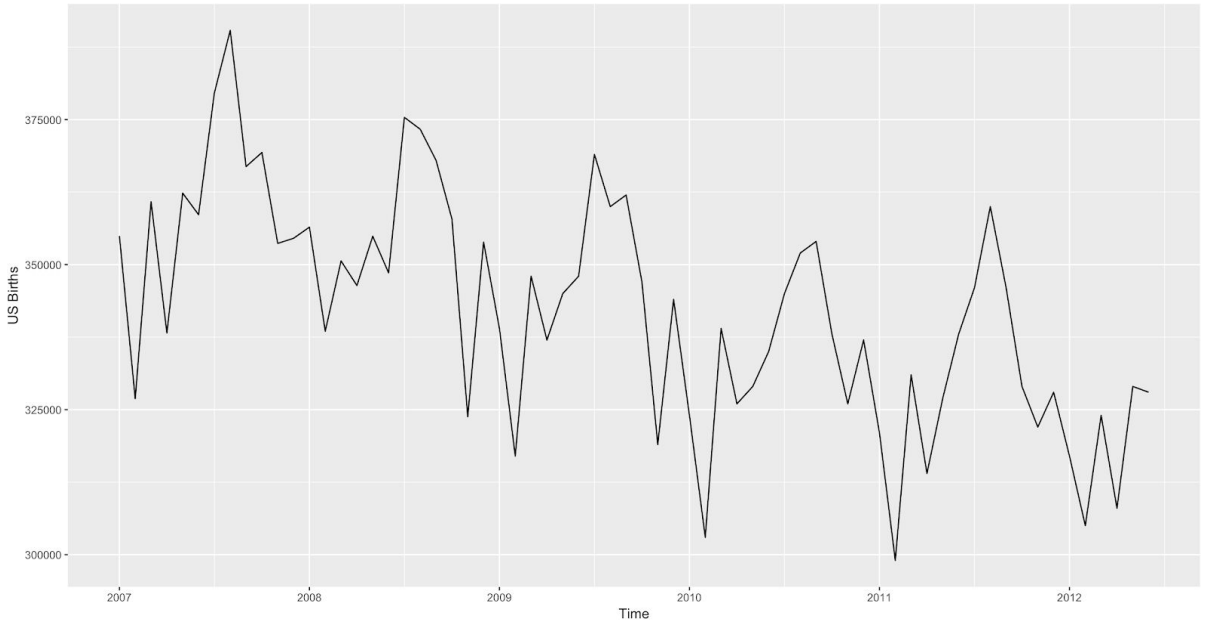
Thus, in order to win, we would recommend Obama should focus on counties where the mean income and educational level is lower, like counties in South. From the map, we can see that Obama performs quite well overall, except in Texas, Kentucky and West Virginia. On the other hand, states like Pennsylvania, South Dakota and Montana are areas where the vote may be indecisive. By focusing on delivering a strong discourse in these regions for the audiences with the social traits that we have analysed, Obama edge Clinton and overcome the 1,230,525 vote differential that he may face according to our prediction.

NICU & US BIRTH DATASETS

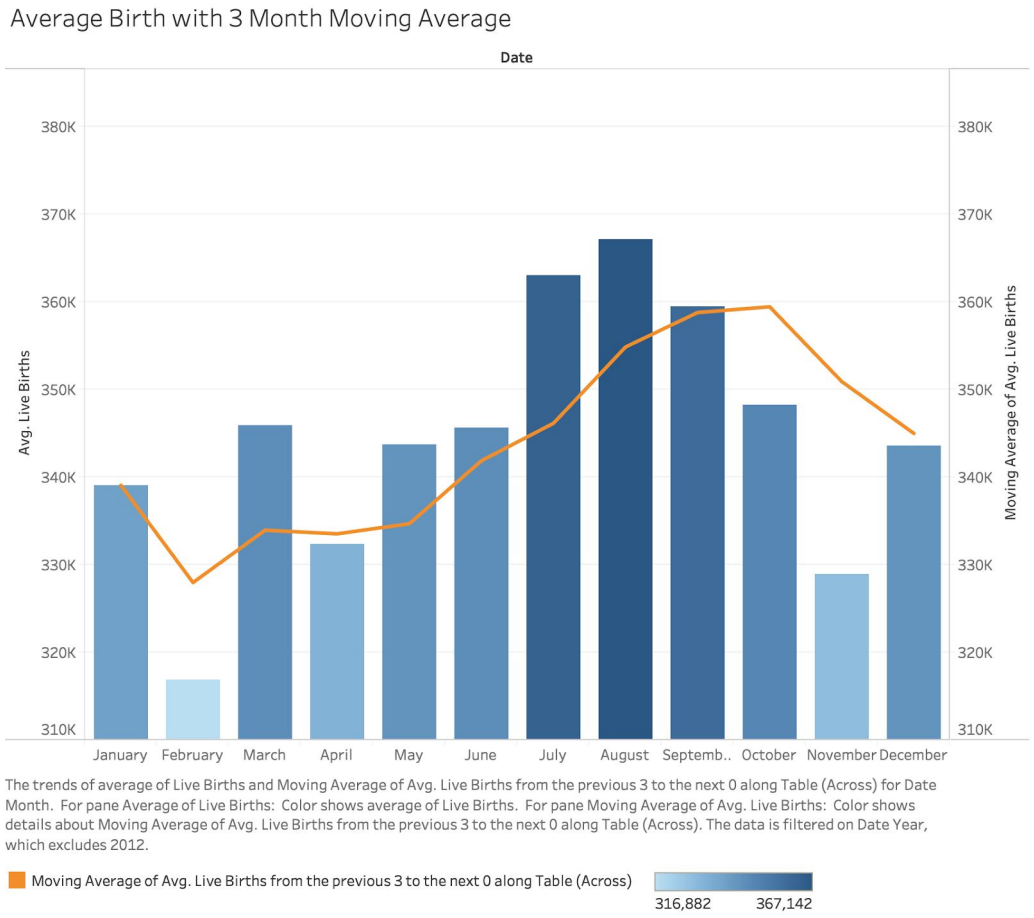
In order to use the US Birth dataset in conjunction with the NICU dataset, we used the Join functionality of Tableau combined with the DATEPARSE function to merge entries by their respective months. Moreover, we used R code to process the data in order to generate a separate Year and Month column attribute from the original date data. We did the same in Tableau using the DATEPARSE function and thereafter begun our analysis.

Seasonality and trend graphs below gives us a great idea about the yearly trends of births. Clearly, there is a stark rise around June/July periods where births spike and then decrease towards the beginning and end of a year.

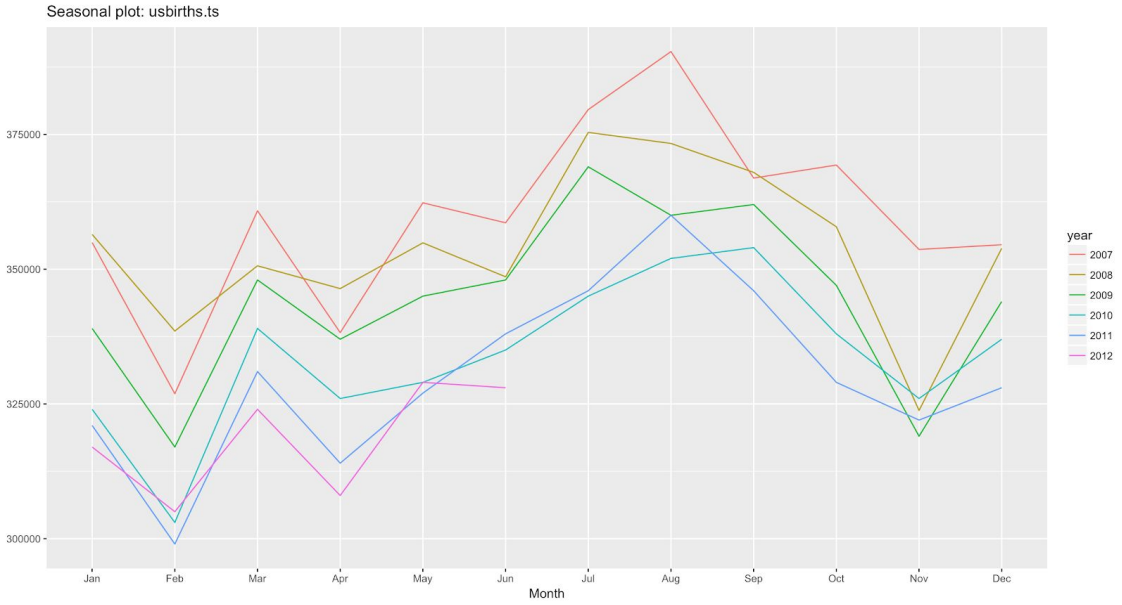
US Births:



To further this, we aggregated the births-per-month and produced the following visual adding in a three month moving average (excluding 2012). This indicated undoubtedly that there is a definite increase in births over the July-September period.



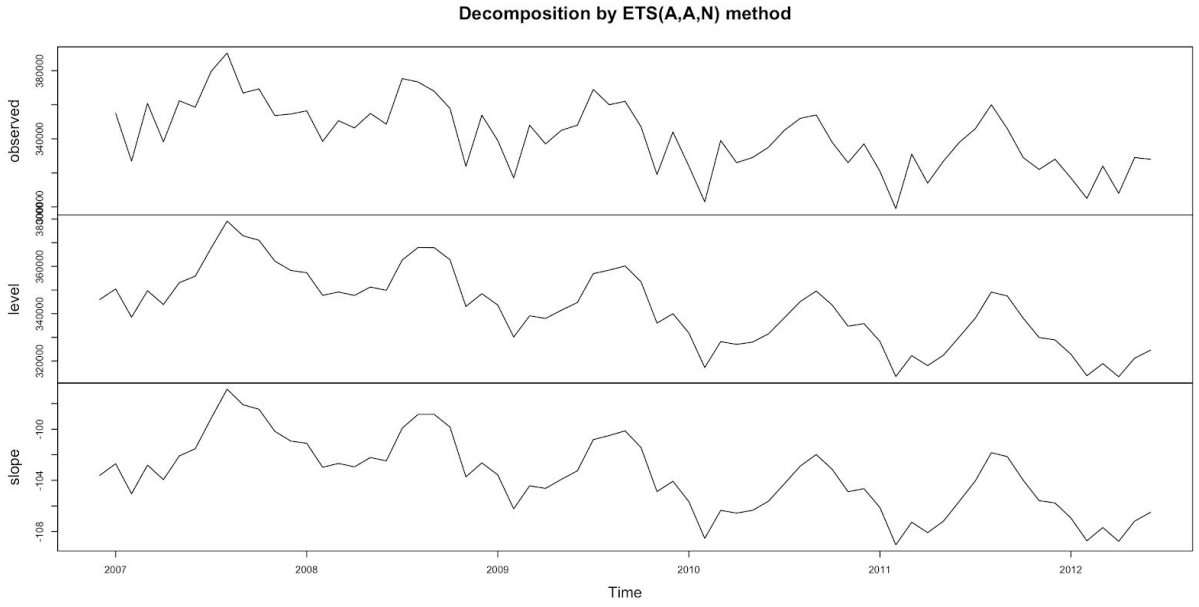
Moreover, we wanted to understand what was happening to the birth rate on a yearly basis. By comparing the yearly trend we also found out that from 2007 - 2012 although the seasonality trend remains somewhat constant, there was a definite drop in the birth count year-on-year:



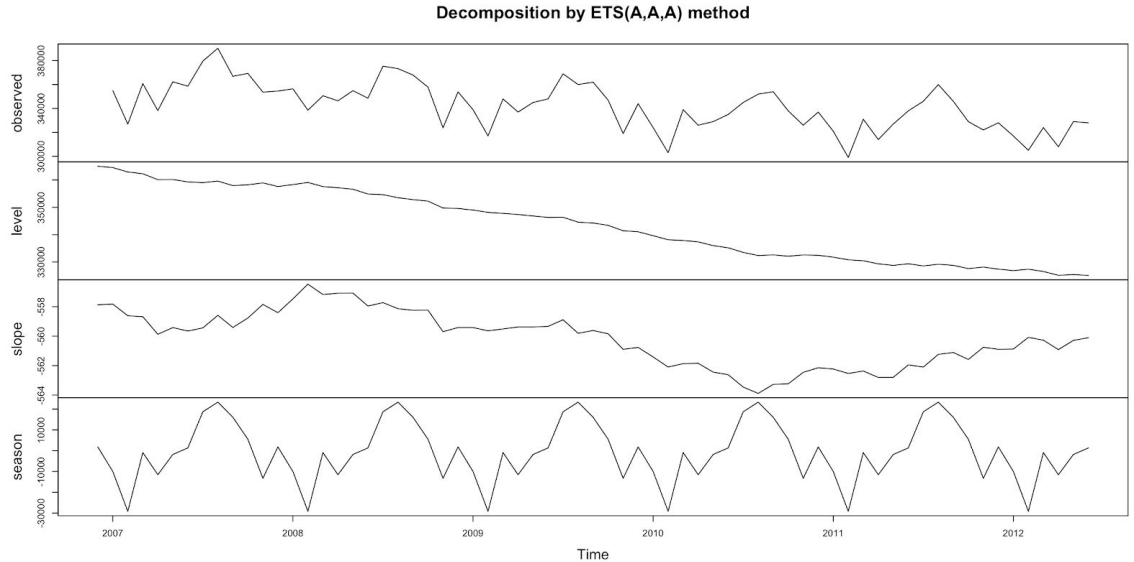
Likely reasons for this increase could be understood by looking at what is happening 40 weeks prior, the natural gestation period. This refers to the Christmas period which is synonymous with baby-making as statistically, more people have sex over winter and holiday periods (Specktor, 2014). Furthermore, the drop in yearly rates can be mostly attributed to the recession of 2008. People struggled financially which directly affected their attitude towards growing families. This was particularly apparent in migrant workers - where mexican migrant worker births dropped a staggering 23% from 2007-2008 (Bahrampour, 2012).

Following on from this, we fitted both a trend (AAN) model and seasonality (AAA) model to the US Birth Dataset and noted their RMSE rates - 15584.27975 and 5622.293199 respectively. The AAA, as expected, returned a more accurate result as it accounts for the seasonality aspect.

AAN Model:

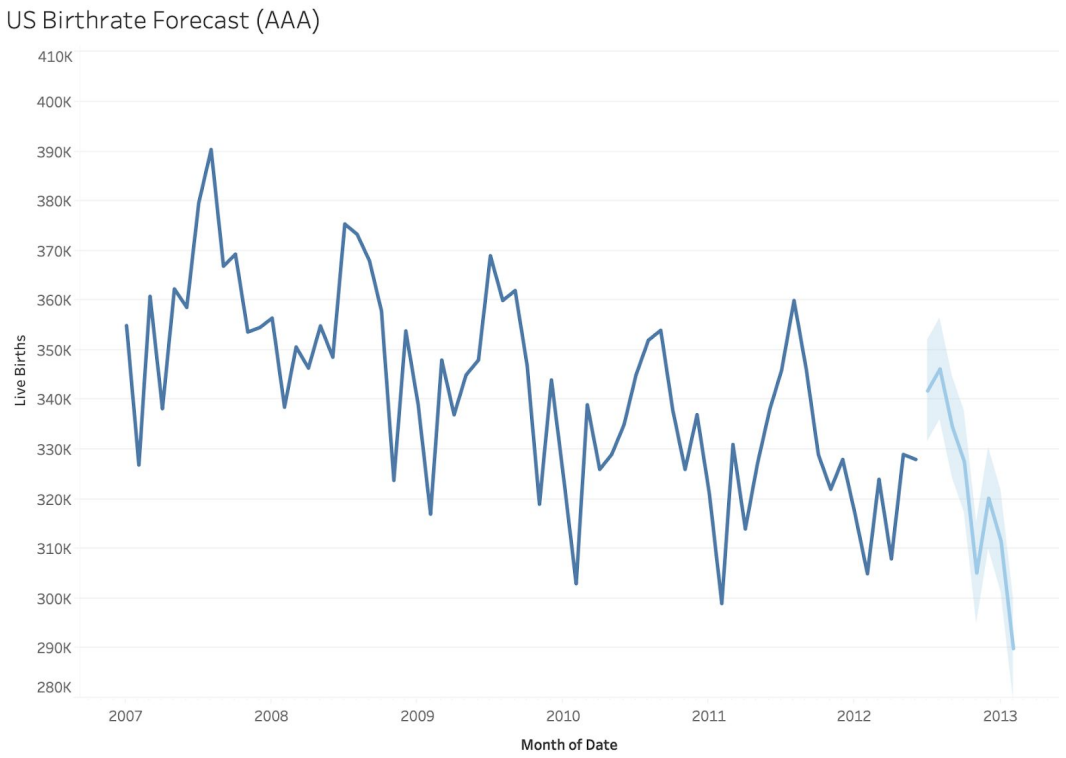


AAA Model:

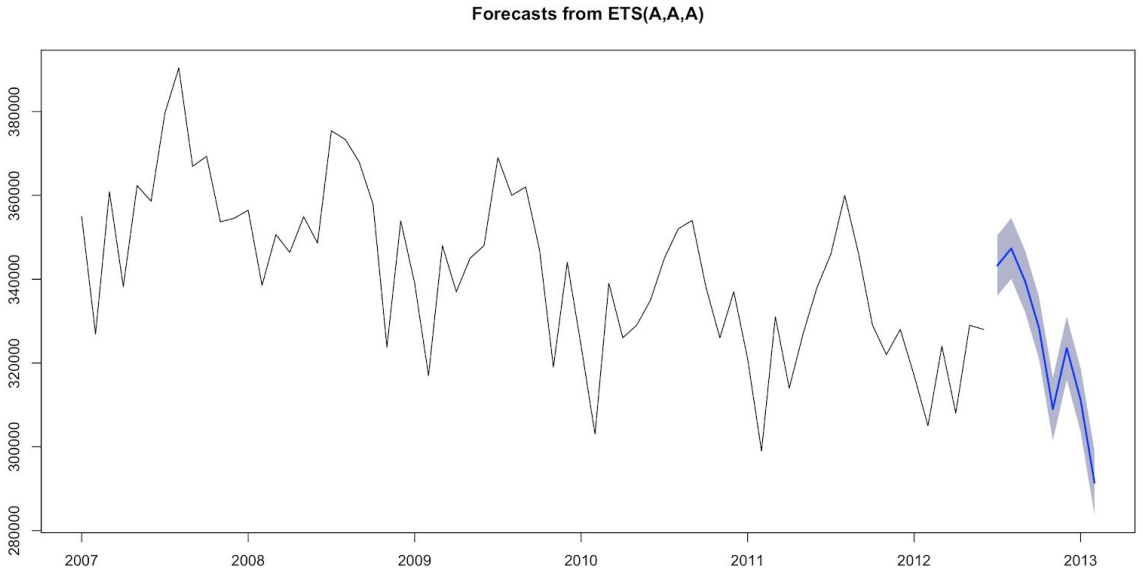


We then utilised the AAA model to forecast US births to February 2013 and plotted this forecast with 80% confidence levels producing the graphs below. Both gave very similar predictions:

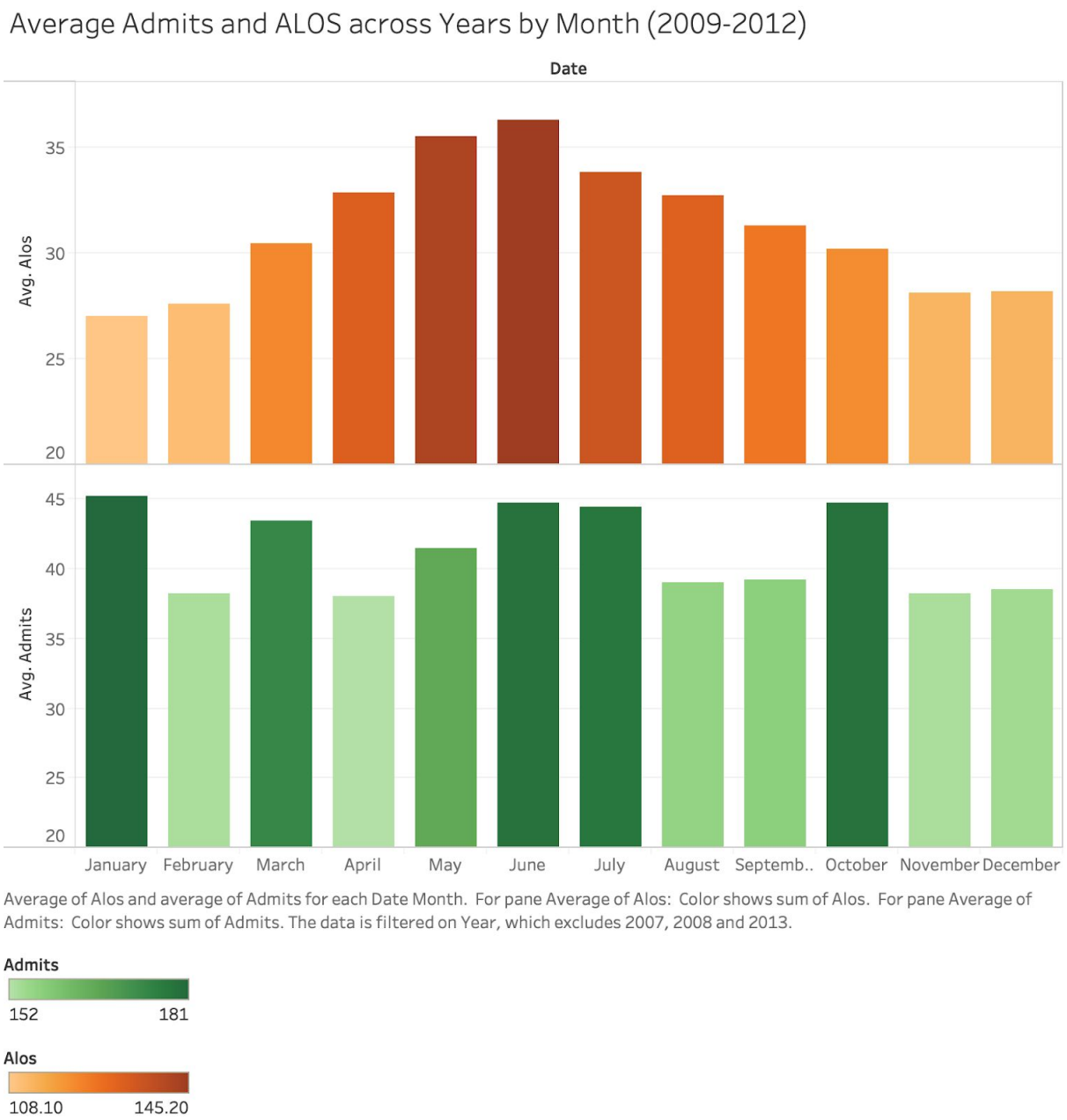
Tableau - Forecast (AAA)



R - Forecast (AAA)

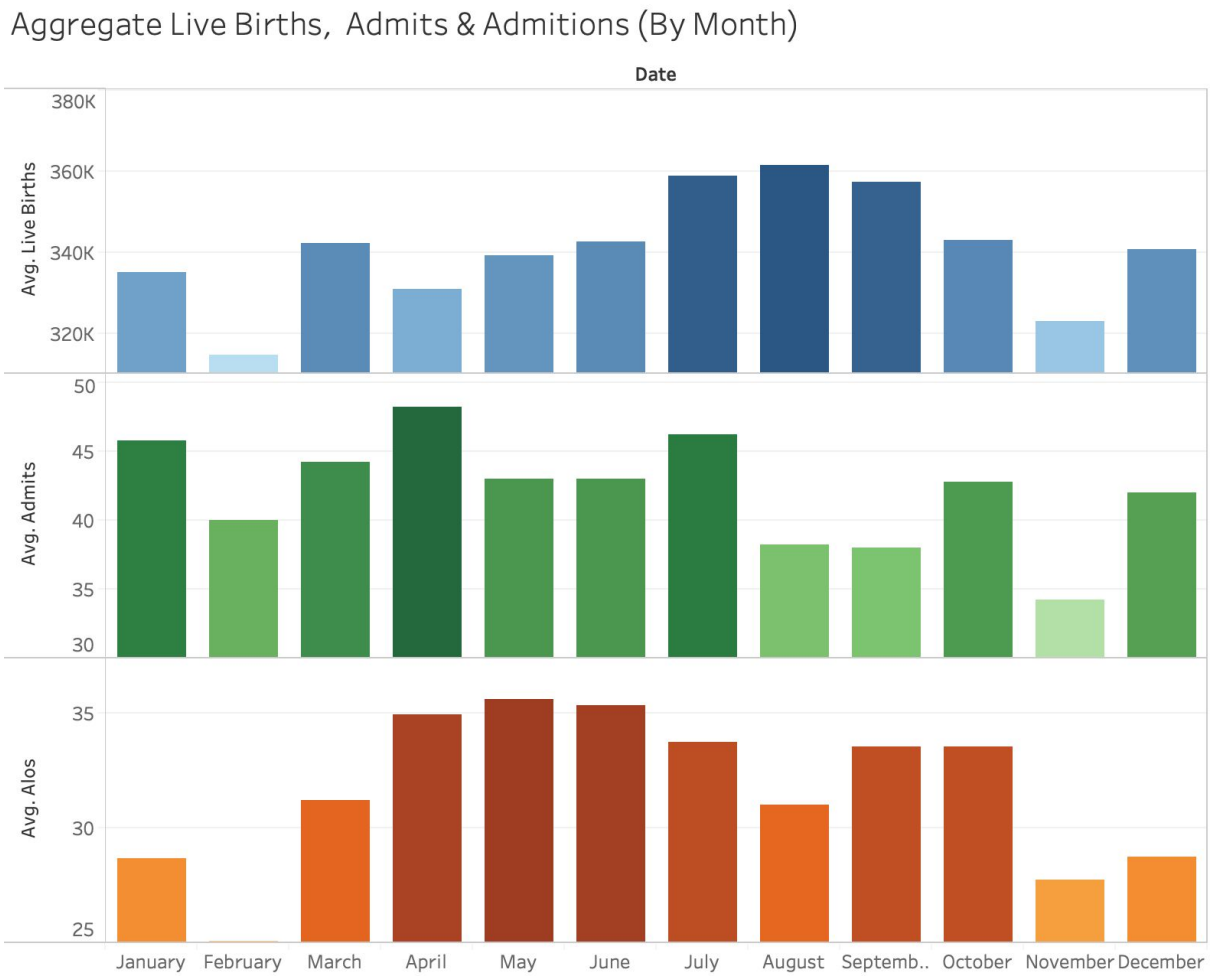


Next, we compared the seasonality patterns in US births, NICU admissions and NICU ALOS. We already knew about the increase in US births converging around the July-September period. As such, we looked at the average length of stay and admissions by aggregating the data by months. This indicated a clear pattern whereby length of stay increased over the April-June period. Similarly the admissions with the highest months also seemed to spike in that period.



This was strange as it seemed to proceed the months where the US Birth data spiked. However, by joining the data sets and visualising the graphs side-by-side, it became visually clear that the spike in births was preceded by increases in

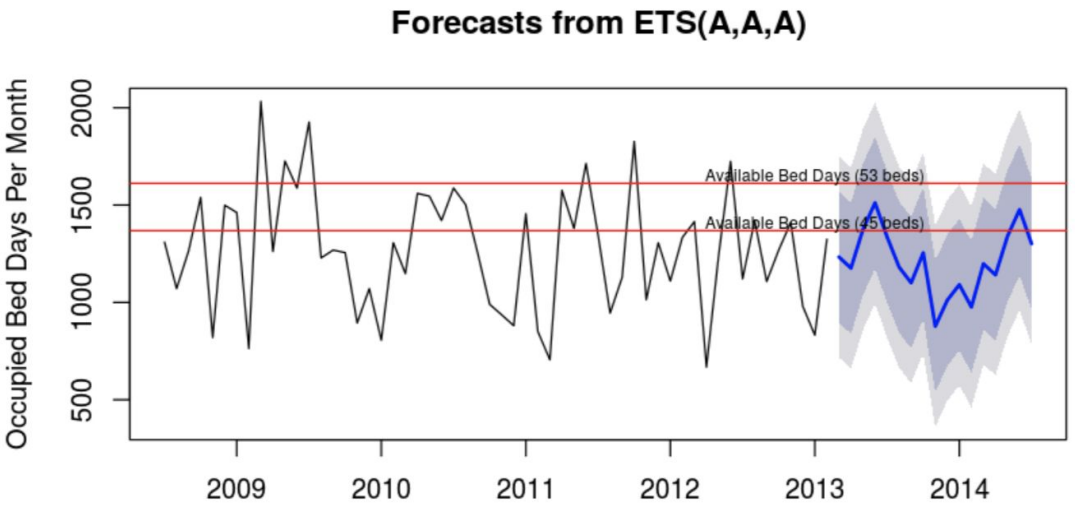
admissions and increased ALOS. This made perfect sense as the primary reason NICU's receive babies is for prematurity. As such, the months preceding the rise in US BIrth data would be the primary time whereby the NICU admissions and length of stay increased.



Recommendations

Through a combination of our analysis and that presented in Week 3 lecture, our recommendation to the COO of Garfield Children's would be to stop the current extra bed funding and rather invest this $1M earmarked for the project in the rest of the children's hospital. Our reasoning for this is based on multiple facets. Firstly, as can be seen from the adapted graph below, we only require turning away babies from the hospital in the busiest period: June and July at which time this number is below the unwritten rule of "3 turnaways". Only at this time would the number of beds be insufficient. Furthermore, the hospital can look at seasonality trends and based on US Births steadily decreasing year-on-year, allocating funds to this unit, which may only help for a select few months a year may be inefficient. This is compounded by following years possibly not requiring this increase based on our downward trending

birth forecast. A such, we believe that the funds would be better utilised by investing in other parts of the children's hospital as it would provide positive year-round utilisation.



Graph from Lecture 3 slides, by Dr Andrew Whiter

Word count: 1999

References:

Bahrampour, T. (2012). U.S. birthrate plummets to its lowest level since 1920. [online] The Washington Post. Available at: https://www.washingtonpost.com/local/us-birth-rate-plummets-to-its-lowest-since-1920/2012/11/29/ee7e8d16-3a3f-11e2-b01f-5f55b193f58f_story.html?noredirect=on&utm_term=.341e77ea52cc [Accessed 5 Feb. 2019].

Hoffower, H. (2018). *Silicon Valley is so expensive that people who make $400,000 think they're middle-class — here's what the middle class actually is in the 25 largest US cities*. [online] Business Insider. Available at: https://www.businessinsider.com/middle-class-income-us-city-san-francisco-2018-2 [Accessed 6 Feb. 2019].

Specktor, B. (2014). Why September Is the Most Popular Month for Birthdays | Reader's Digest. [online] Reader's Digest. Available at: https://www.rd.com/culture/september-popular-birth-month/ [Accessed 5 Feb. 2019].

R Code Appendix:

```
################### FIRST PART OF THE PROJECT ###################

# Load the data file directly from the internet

library(curl)
x <- curl(
"https://raw.githubusercontent.com/awhiter/TeachingRepos/master/datasets/Ob
ama.csv")
elect.df  <- read.csv(x)


# Create new derived target attributes
elect.df$ObamaRate <- 100 * elect.df$Obama / elect.df$TotalVote

# Imputing missing values:
# Missing values for AverageIncome are replaced by the
# MedianIncome for that same record

elect.df$AverageIncome <- ifelse(is.na(elect.df$AverageIncome),
                                 elect.df$MedianIncome,
                                 elect.df$AverageIncome)

# Missing values for the following list of attributes
# are replaced by 0.

for (attr in c("Black","Asian","AmericanIndian","ManfEmploy",
               "Disabilities","DisabilitiesRate","FarmArea"))
    {elect.df[[attr]] <- ifelse(is.na(elect.df[[attr]]),
                                 0,
                                 elect.df[[attr]])}

# There still remain several attributes with 1 or 2 missing values.
# It turns out that all these final missing values are in 2 records.
# The following codes removes these records entirely.

elect.df <- elect.df[is.na(elect.df$HighSchool)==FALSE,]
elect.df <- elect.df[is.na(elect.df$Poverty)==FALSE,]


#Convert ElectionDate column to the "Date" data type
elect.df$ElectionDate <- as.Date(elect.df$ElectionDate,
                                 format="%m/%d/%Y")
```

```
#Creating known and unknown data

elect.df.known <- elect.df[elect.df$ElectionDate <
                           as.Date("2/19/2008", format = "%m/%d/%Y"), ]

elect.df.unknown <- elect.df[elect.df$ElectionDate >=
                             as.Date("2/19/2008", format = "%m/%d/%Y"), ]

#Quick veiw of how many rows are in each data set

nrow(elect.df.known)
nrow(elect.df.unknown)

# Find the number of rows in the known dataset
nKnown <- nrow(elect.df.known)

# Set the seed for a random sample
#set.seed(201)

# Randomly sample 80% of the row indices in the known dataset
rowIndicesTrain <- sample(1:nKnown,
                          size = round(nKnown*0.8),
                          replace = FALSE)

# Split the training set into the training set and the test set using these
indices.

elect.df.training <- elect.df.known[rowIndicesTrain, ]

elect.df.test <- elect.df.known[-rowIndicesTrain, ]


# first set library path to include the following directory
# if running on Azure LinuxDataScience VM ...

.libPaths('/home/vmuser/R/x86_64-pc-linux-gnu-library/3.2')

# The Metrics package includes the mae and rmse functions.
# Install Metrics if needed..
# install.packages("Metrics")

library(Metrics)
library(MicrosoftML)

genError <- function(prediction, actual)
    cat('MAE =', signif(mae(actual,prediction),4),
        ' RMSE =', signif(rmse(actual,prediction),4), "\n")
```

```
################### GRAPHS:

#Correlation heatmap:
ClintonRate <- elect.df$ClintonRate
ObamaPercentMargin <- elect.df$ObamaPercentMargin
res<-cor(elect.df[,c(7:42)],use="complete.obs")
col<- colorRampPalette(c("blue", "white", "red"))(20)
heatmap(x = res, col = col, symm = TRUE)

#Correlation table:
targetCol <- which(names(elect.df)=="ObamaRate")
startCol <- which(names(elect.df)=="MalesPer100Females")
endCol <- which(names(elect.df)=="DisabilitiesRate")

sort(
  cor(elect.df[,c(targetCol, startCol:endCol)],use="complete.obs")[1,],
  decreasing = TRUE) [1:6]

#Aggregation Table:
elect.df$Winner <- ifelse(elect.df$Obama>elect.df$Clinton,
                          "Obama",
                          "Clinton")
aggregate(cbind(Bachelors, IncomeAbove75K, HighSchool, MedianIncome,
Poverty) ~ Winner,
          data=elect.df,
          FUN=mean)

#Plot 2:
ggplot(Obama, aes(x=ObamaRate,y=AverageIncome)) +
geom_point(aes(color=Region)) + geom_smooth(method='lm')

#Plot 3:
BlackPopulation <- elect.df$Black/100*elect.df$Pop

ggplot(elect.df, aes(x=Region, y=BlackPopulation))+
  geom_bar(stat="identity", color="black")+
  scale_fill_manual(values=c("#999999", "#E69F00", "#56B4E9"))+
  theme_minimal()
```

```
################### PREDICTION MODELS:

#Basic Lasso Regulization Analysis

lm.lasso <- glmnet(xknown, yknown, family = "gaussian")

plot(lm.lasso, xvar = "lambda", label = TRUE)

coef(lm.lasso, s = exp(0))

#Using cross validation to optimise Lamdfor Lasso Regulization Model

set.seed(101)
lm.lasso.cv <- cv.glmnet(xknown, yknown, nfolds = 5, family = "gaussian")

#This gives the value for an optimised lamda value
lm.lasso.cv$lambda.min
(minLogLambda <- log(lm.lasso.cv$lambda.min))

#Plotting optimised Lamda Lasso Regulization

plot(lm.lasso, xvar = "lambda", label = TRUE)
abline(v = log(lm.lasso.cv$lambda.min))

coef(lm.lasso.cv, s = "lambda.min")
# Coefficients of the regularized linear regression with an optimal lambda.

# Determining the generalised error rate for this lasso model

xtest <- as.matrix(elect.df.test[, startCol:endCol])

lm.lasso.cv.pred <- predict(lm.lasso.cv, newx = xtest, s = "lambda.min")

genError(lm.lasso.cv.pred, elect.df.test$ObamaRate)

#rpart graph

library(rpart)
library(rpart.plot)    # install.packages("rpart.plot") is needed

rt <- rpart(ObamaRate ~ HighSchool + Poverty + Bachelors + AverageIncome +
UnemployRate,
            data = elect.df.training)  # Fits a regression tree.

prp(rt, type = 1, extra = 1) #Plots Tree
```

```
# GBM Boost:

library(gbm)
elect.df.boost <- elect.df.training
elect.df.boost$County <- NULL
elect.df.boost$State <- NULL
elect.df.boost$Region <- NULL
elect.df.boost$FIPS <- NULL
elect.df.boost$ElectionDate <- NULL
elect.df.boost$Obama <- NULL
elect.df.boost$Clinton <- NULL
elect.df.boost$ElectionType = NULL

obama.boost = gbm(ObamaRate ~ . , data = elect.df.boost, n.trees = 100,
shrinkage = 0.01, interaction.depth = 4)
obama.boost
summary(obama.boost, cBars = 5, order=TRUE)


# Random Forest:

library(randomForest)
obama.forest <- randomForest(ObamaRate ~ . , data = elect.df.boost,
importance=TRUE, ntree=100)
# RMSE 9.15224
obama.forest <- randomForest(ObamaRate ~ Black + HighSchool + Bachelors +
DisabilitiesRate + SocialSecurityRate + White + IncomeAbove75K + Poverty +
MedianIncome, data = elect.df.boost, importance=TRUE, ntree=100)
# RMSE 9.81717
obama.forest.pred = predict(obama.forest,newdata=elect.df.test)
rmse = sqrt(mean((elect.df.test$ObamaRate-obama.forest.pred)^2))
plot(obama.forest)

plot(elect.df.test$ObamaRate,pch=16)
points(obama.forest.pred, col = "blue", pch=4)

# SVM:
library(e1071)

model_svm <- svm(ObamaRate ~ Black + HighSchool + Bachelors +
DisabilitiesRate + SocialSecurityRate + White + IncomeAbove75K + Poverty +
MedianIncome + Homeowner + AverageIncome + UnemployRate, data)
# 9.71428
model_svm <- svm(ObamaRate ~ ., elect.df.boost)
# 9.5216
pred <- predict(model_svm, elect.df.test)

rmse = sqrt(mean((elect.df.test$ObamaRate-pred)^2))
plot(elect.df.test$ObamaRate,pch=16)
points(pred, col = "blue", pch=4)
```

```
# Applying Prediction:
obama.forest.pred.final <- predict(obama.forest,elect.df.unknown)
solution <- data.frame(County = elect.df.unknown$County, ObamaRate =
obama.forest.pred.final)
write.csv2(solution,file="solution.csv",row.names=FALSE)


################### SECOND PART OF THE PROJECT ###################

# US births TS Analysis:
usbirths.df <- read_csv("~/Downloads/UCL/YEAR II/Term 2/Data Analytics
II/Week III/US_Births.csv")

usbirths.ts <- ts(usbirths$`Live Births`,
                  start = c(2007, 1),
                  end = c(2012, 6),
                  freq = 12)

library(ggplot2)

# Plots:
autoplot(usbirths.ts, ylab = "US Births")
ggseasonplot(usbirths.ts)

# AAN:
rmse.ets <- function (etsmodel) cat("RMSE = ", sqrt(etsmodel$mse))
rmse.ets(usbirths.ets.AAN)
(usbirths.ets.AAN <- ets(usbirths.ts, model = "AAN"))
rmse.ets(usbirths.ets.AAN)
plot(usbirths.ets.AAN)


usbirths.ets.AAN.pred <- forecast(usbirths.ets.AAN, h = 8)
plot(usbirths.ets.AAN.pred)
# plot the forecasts for the AAN model:
par(mfrow = c(1, 2))
plot(admits.ets.AAN.pred)
admits.ets.AAN.pred$mean[17]

# AAA:
rmse.ets(usbirths.ets.AAA)
(usbirths.ets.AAA <- ets(usbirths.ts, model = "AAA"))
rmse.ets(usbirths.ets.AAA)
plot(usbirths.ets.AAA)


# AAA Prediction Graph:
usbirth.ets.AAA.pred <- forecast(usbirths.ets.AAA, h = 8, level = 80)
plot(usbirth.ets.AAA.pred)
```