DATA ANALYTICS II

2021

GIULIA ZHANG 19112842
HOIJIN KIM 19081830
TANIA TURDEAN 19004997
MINHEA MACOVEI 19030283
THAIS PARISOT 19074092
NINON LAVOLLE 19098458

Statistics:

Pages	27
Words	1.974
Characters (no spaces)	9.589
Characters (with spaces)	11.509
Paragraphs	126
Lines	447

☑ Include footnotes and endnotes

Table of Contents

Obama - Clinton Case Study

Section 1: The Problem

1.1 Problem: Obama-Clinton case study

Obama-Clinton case study aims at understanding and analyzing patterns about Obama and Clinton's voters within the democratic primaries in order to make predictions related to areas which have not voted yet. (1)

1.2 Subproblem: Electorate Segmentation Research

The chosen sub-problem that is the focus of this report in the analysis of **Obama's likelihood to win** in areas which have not voted yet.

- Therefore, we had to create a derived target variable "**ObamaWin**" that shows Obama's victory with values (1-win, 0-loss).

In order to help the Obama's team to make more informed election campaigns.

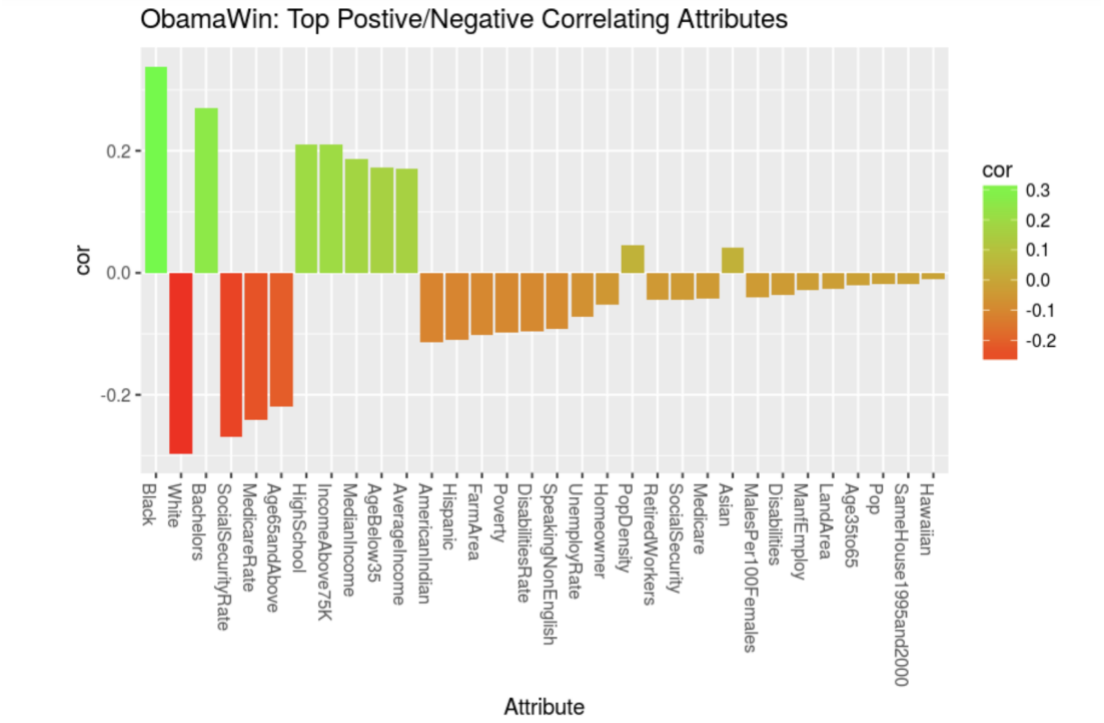
- we have created visualisations and predictive models to learn more about the voters and adapt to their needs.

Our target variable will be Obama's victory **"ObamaWin".**

- assigning 1 to countries where he won and 0 to those where Clinton won.

Our focus would be on 3 attributes: ***income, race and age.***

- as they are the most significant features lying under Obama's victory. This has been decided using the correlation graph created in R , shown below.



Section 2: Understand the Data

2.1 Nature, size and source of data

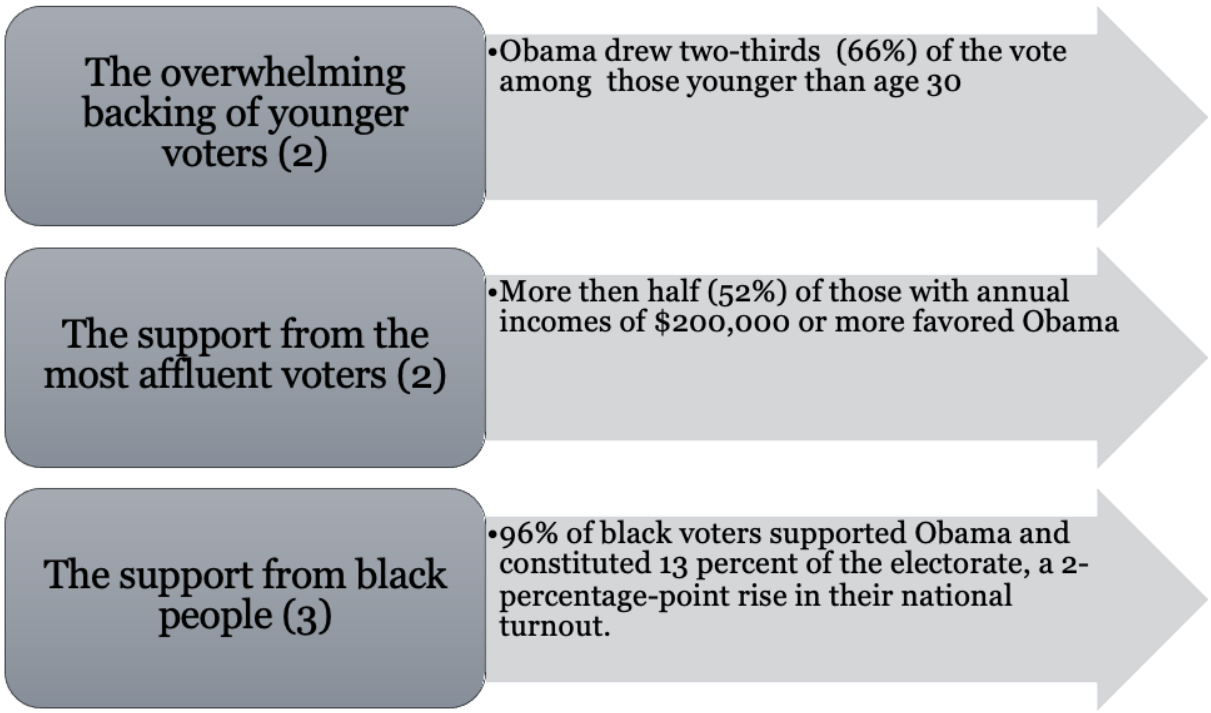
The Obama-Clinton 2008 Election's Dataset comes from the U.S. census Bureau which shows demographic information for each county. It has 2868 rows and 41 columns. From the 2868 rows, there are 1737 observations of "known data" and 1131 of "unknown data". The attributes are categorical, numerical or date-format. (1)

Additionally, we have created 2 other columns which represent the target variable. One column with ObamaWin with the values ("win" and "lose") and another column with the values ("1" and "0").

2. 2 Data attributes

2.2.1 Obama's likelihood to win: most relevant attributes

Background research: Inside Obama's victory
According to an analysis of National Election Pool exit polls, the critical factors in Obama's victory were:



2.2.2. Studying the relevant data attributes

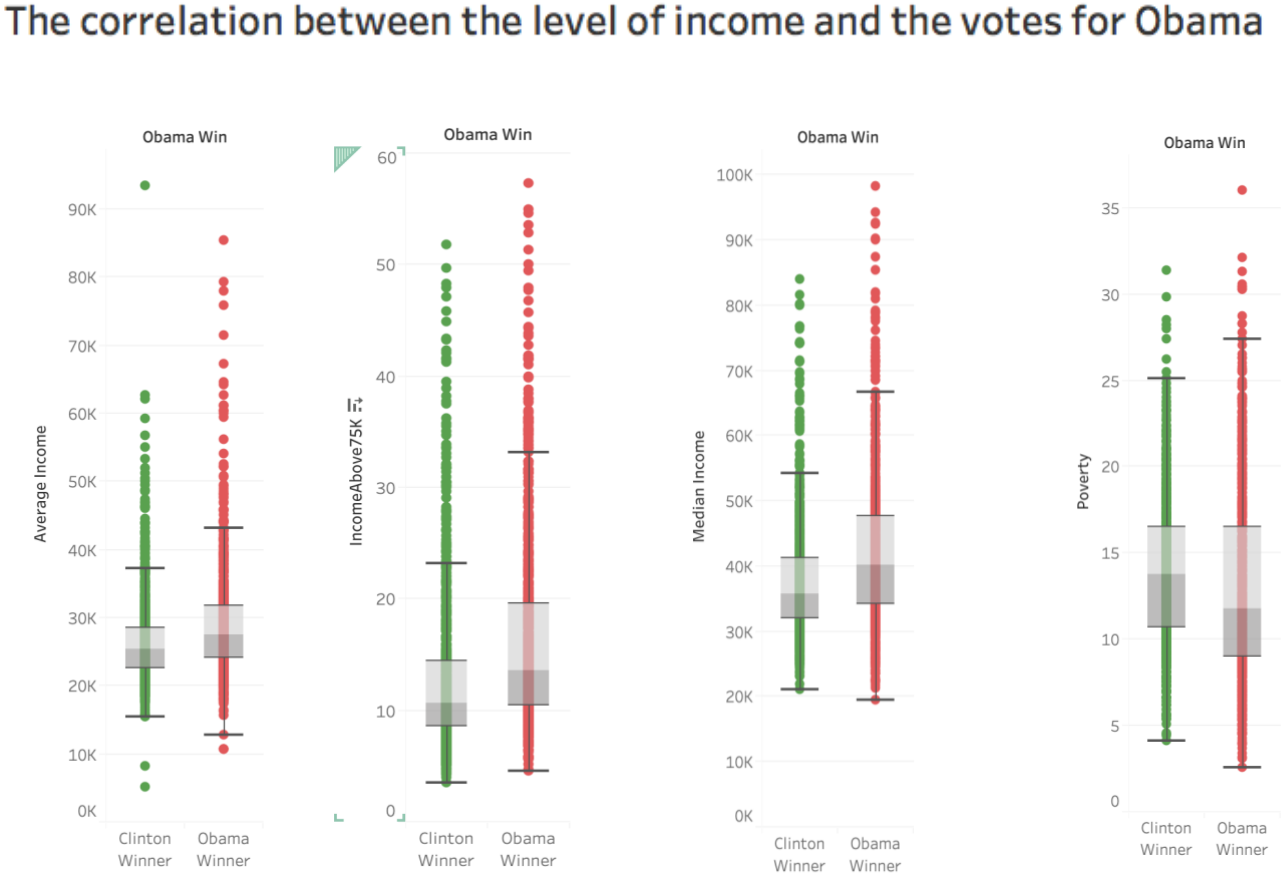
To support our discussion, we created 3 visualisations in Tableau and 4 visualisations in R.

2.2.2.1. Income

Income is represented in the dataset by 4 numerical attributes. AverageIncome and MedianIncome represent the median and the average for income for each County. IncomeAbove75K and Poverty are represented in percentages for each county.

The level of wealth is an important factor in election results. The four box plots prove that, on average, people with higher incomes are more likely to vote for Obama. Also, poverty is more likely to affect people voting for Clinton.

On a closer look, the box plots also show how data is distributed. For example, we can see that for Poverty and counties in which Obama lost, the average is 14%, the 25th percentile is 11% and the 75th percentile is 16%.



(Tableau)

The aggregated table breaks down the box plot into the 4 regions. We can see for example that Clinton edged Obama in 3 out of 4 regions for poverty voters.

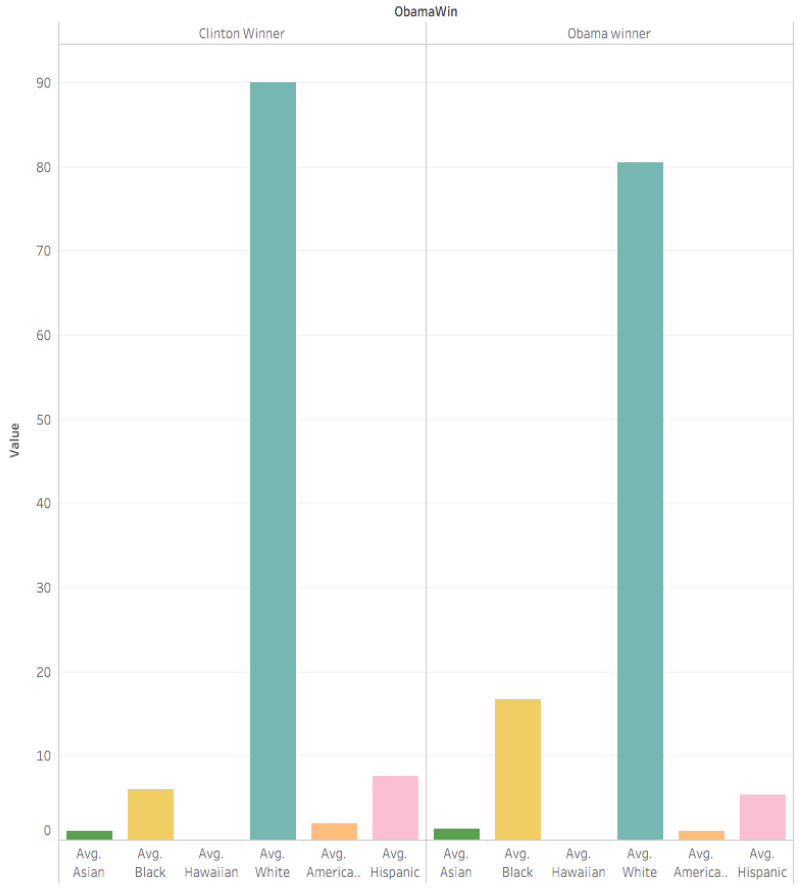
<Income>

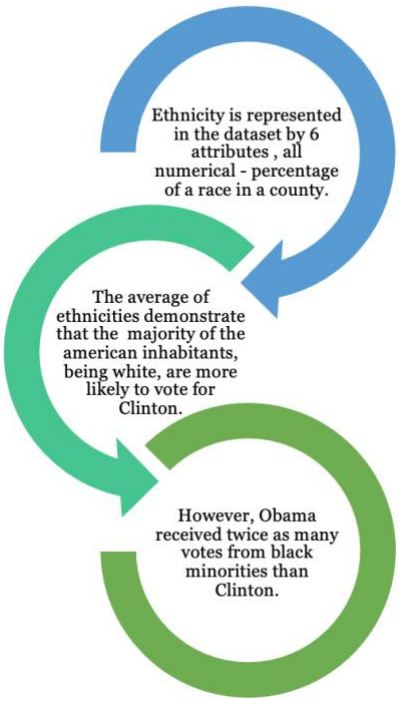
		ObamaWin		
	Region	0	1	na
Avg. IncomeAbove75K	Midwest	10	15	15
	Northeast	22	24	16
	South	12	16	12
	West	15	18	12
Avg. Poverty	Midwest	12	10	11
	Northeast	11	9	11
	South	15	16	17
	West	15	11	13
Avg. Median Income	Midwest	37,059	43,474	42,119
	Northeast	49,244	52,479	42,611
	South	35,585	39,765	34,593
	West	39,296	44,207	38,173
Avg. Average Income	Midwest	25,881	29,087	28,758
	Northeast	33,792	38,141	30,308
	South	25,007	27,120	25,135
	West	26,330	29,872	28,906

Avg. IncomeAbove75K, Avg. Poverty, Avg. Median Income and Avg. Average Income broken down by ObamaWin vs. Region. (Tableau)

2.2.2.2. Ethnicity

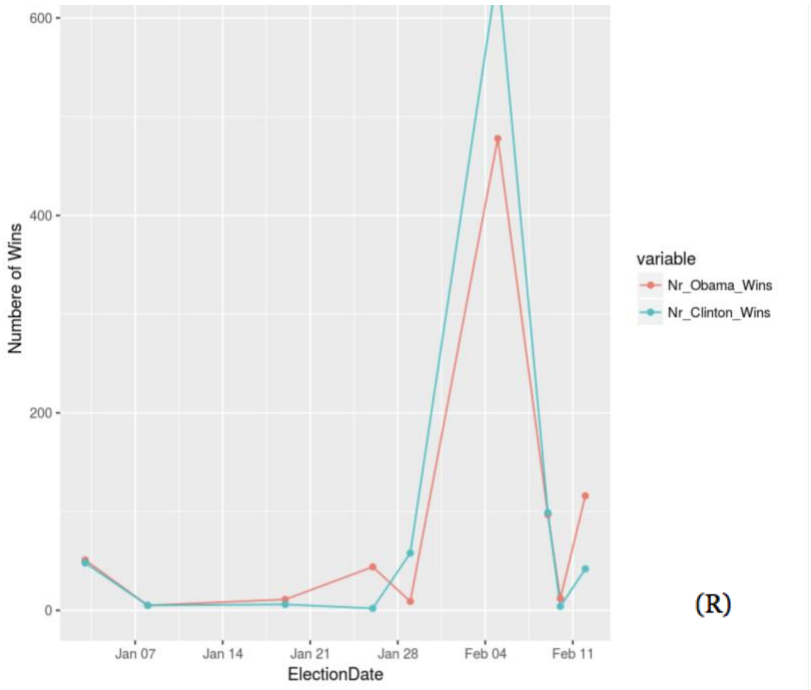
The correlation between ethnicities and votes for Obama



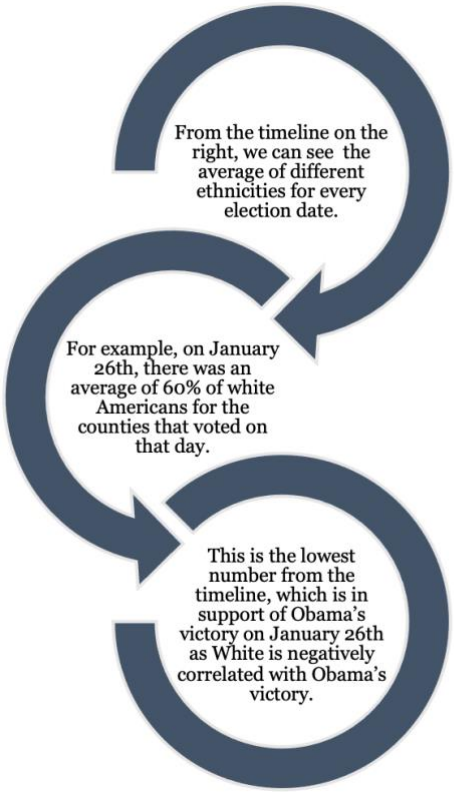


(Tableau)

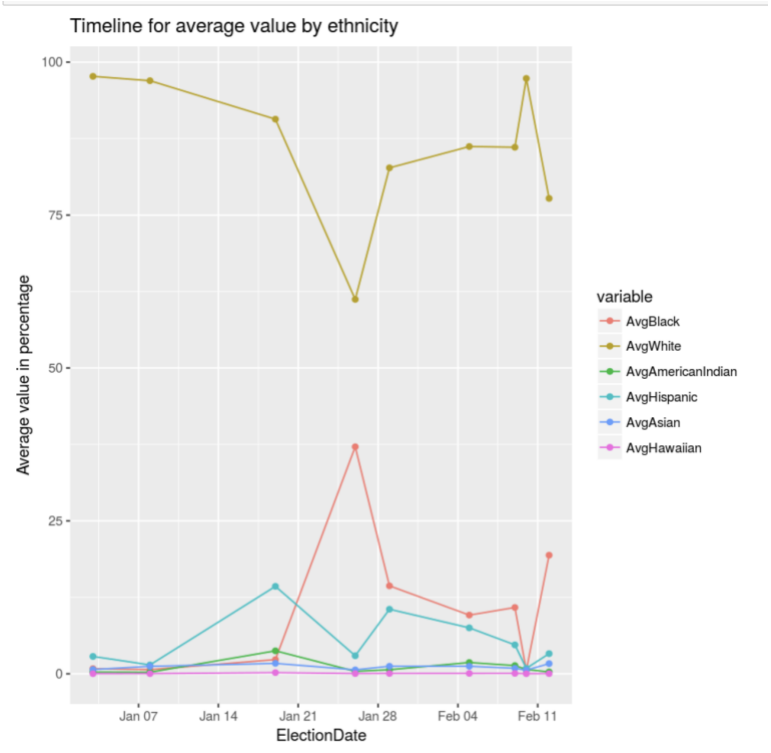
From the timeline below, we can see the number of wins per candidate in the counties that already voted. On January 26th, Obama won in more counties than Clinton.



(R)



(R)



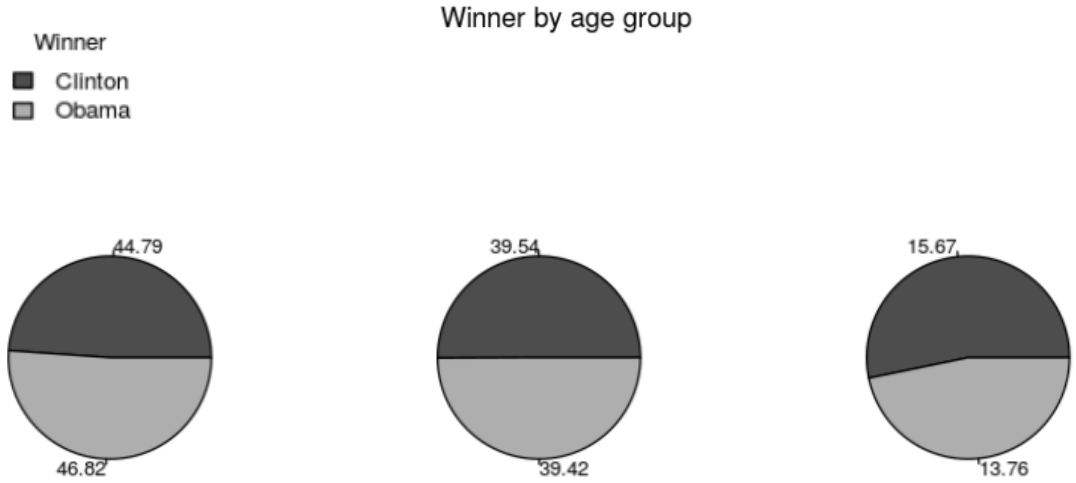
Looking at both timelines, we can see how the lines for the number of Obama wins and average black go in the same direction most of the time.

2.2.2.3 Age

The visual and table for this section were made in R.

Age is represented in the dataset by 3 attributes (different age groups), all numerical. They represent the percentage of people in that age group in every county.

We can see from the pie charts that Clinton edged Obama in drawing voters over 65 or that Obama edged Clinton in drawing voters below 35.

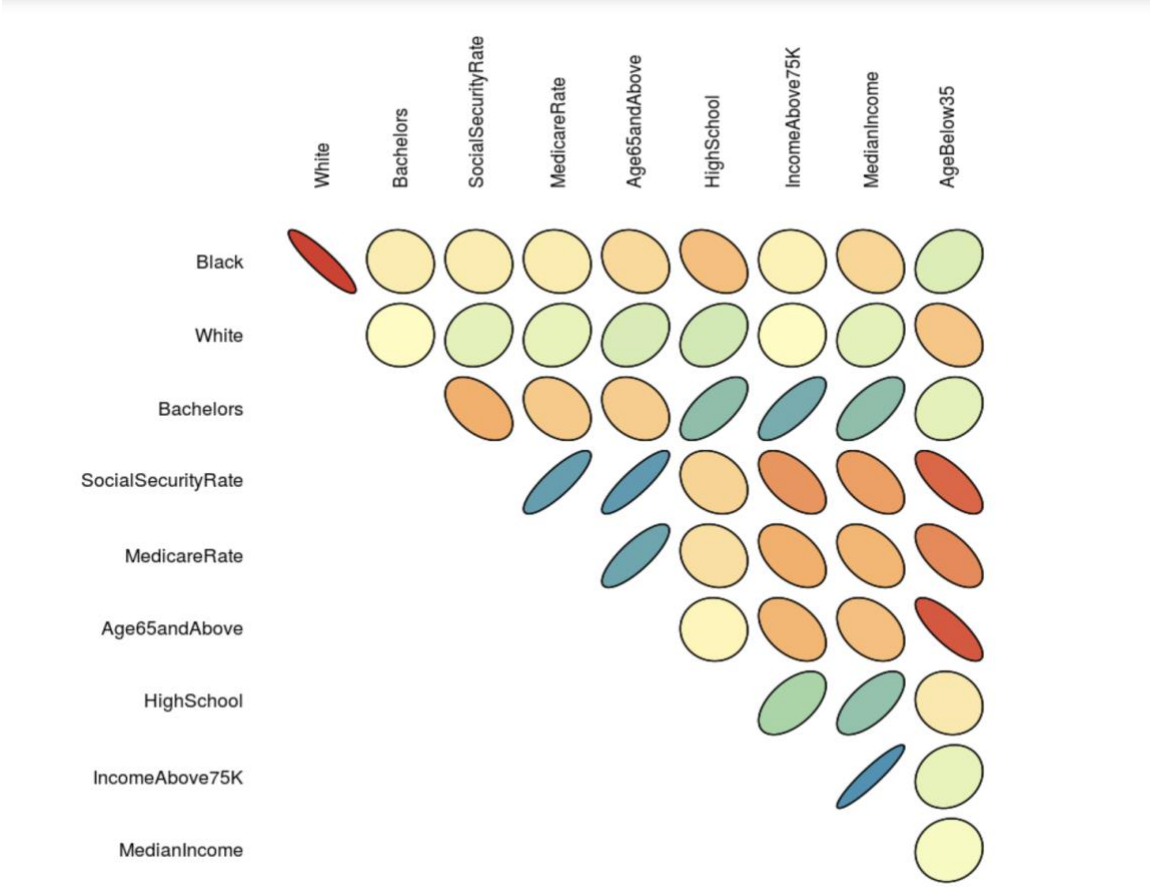


The aggregated table breaks down the pie chart into 4 regions. We can see for example that Clinton edged Obama in 3 out of 4 regions for voters aged above 65 (one of the regions had the same mean).

ObamaWin	Region	AgeBelow35	Age35to65	Age65andAbove
Lose	Midwest	43	40	18
Win	Midwest	45	39	16
Lose	Northeast	45	41	14
Win	Northeast	43	43	14
Lose	South	45	40	15
Win	South	48	39	13
Lose	West	48	38	14
Win	West	47	40	13

2. 3 Correlation between variables

The R visual below shows the correlations between the top 10 most correlated attributes with ObamaWin (see 1.2.)



We can see that Black is highly correlated with White (negatively) - more black people means less white people. Also, IncomeAbove75K is highly correlated with MedianIncome (positively) - they increase in the same direction.

These correlations helped us improve the best prediction model in Section 4 by removing the most highly correlated attributes.

See appendix 2.3. for correlations between all the variables.

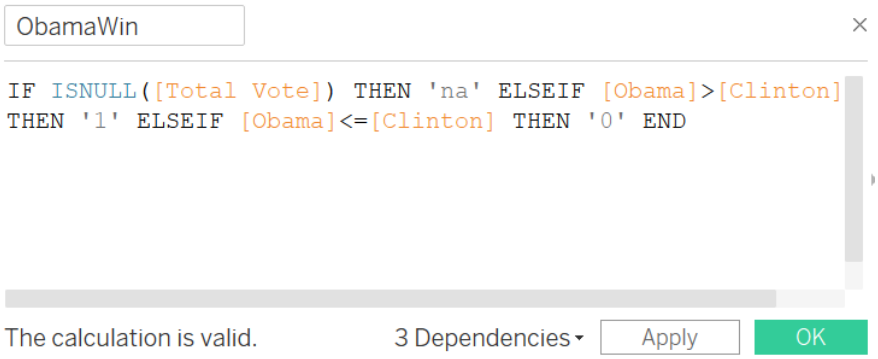
Section 3: Data pre-processing

In order to build and train models, the raw data has to be pre-processed. This process has been done both in R and Tableau and it consists of 2 main parts: Data Manipulation and Data Splitting.

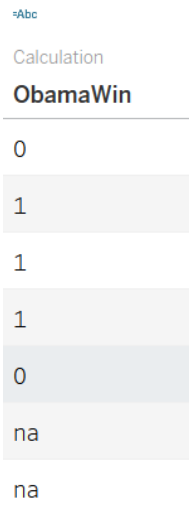
3. 1 Data manipulation

3.1.1 Data manipulation: Tableau

In Tableau, a new calculated field called 'ObamaWin' was created to represent whether Obama won or lost the total number of votes in different counties.



If the number of total votes is not available, 'na' is given. If the number of votes for Obama is greater than that of Clinton, 1 is shown in the column, and lastly, if the number of votes for Obama is smaller or equal to that of Clinton, 0 is the result shown.



Therefore, 1 represents Obama's win in that county, 0 means drew or Obama's lose and 'na' is for the counties that have not voted yet. The number of votes for counties with 'na' is going to be predicted using prediction models in R.

3.1.2 Data manipulation: R

Create target variable for Obama's likelihood to win

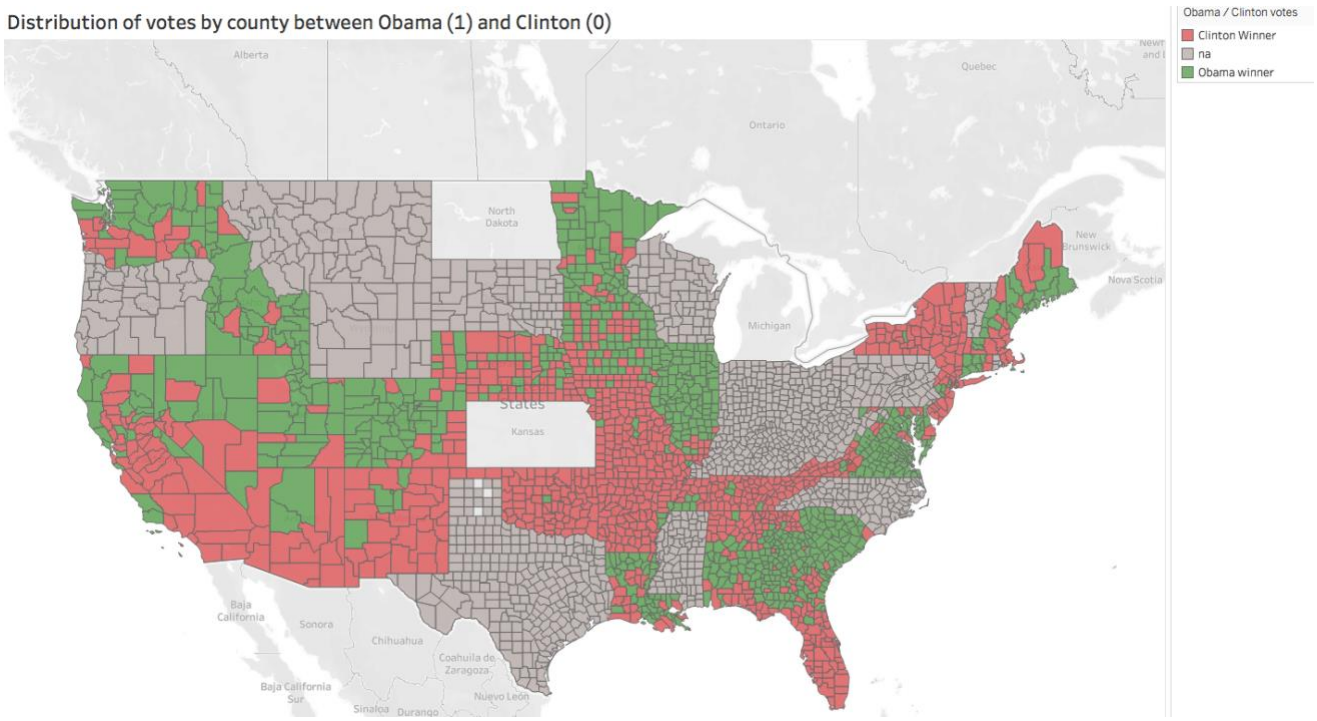
We created two columns for the target variable . One called ObamaWin with the values ("win" and "lose) and another one called ObamaWinNr with the values ("1" and "0).

```
for (row in 1:nrow(elect.df)) {
     Obamav <- elect.df[row, "Obama"]
     Clintonv<- elect.df[row, "Clinton"]
    if(is.na(Obamav))
        {elect.df[row,"ObamaWin"]<-NA
         elect.df[row,"ObamaWinNr"]<-NA
        }
    else if(Obamav>Clintonv)
        {elect.df[row,"ObamaWin"]<-"Win"
        elect.df[row,"ObamaWinNr"]<-1}
    else
        {elect.df[row,"ObamaWinNr"]<-0
        elect.df[row,"ObamaWin"]<-"Lose"}
        }
```

```
elect.df
```

emales	⋯	Disabilities	DisabilitiesRate	Homeowner	SameHouse1995and2000	Pop	PopDensity	LandArea	FarmArea	ObamaWin	ObamaWinNr
	⋯	90	1145	75.0	66.0	7714	13.5	570	373	Win	1
	⋯	78	1829	74.7	65.4	4192	9.9	425	238	Lose	0
	⋯	186	1265	76.4	64.1	14796	23.1	659	326	Win	1
	⋯	453	3315	74.0	60.2	13422	27.0	516	236	Lose	0
	⋯	79	1223	79.2	70.4	6278	14.2	444	261	Win	1
	⋯	269	996	79.4	60.8	26962	37.6	718	401	Lose	0
	⋯	2835	2252	68.9	54.5	126106	222.4	572	275	Win	1
	⋯	359	1350	75.8	60.6	26584	46.5	574	313	Lose	0
	⋯	170	718	78.2	62.0	23837	54.4	440	255	Win	1
	⋯	277	1318	78.2	62.9	21045	36.8	573	340	Win	1

We showed this in Tableau. The counties in grey represent the places which have not voted yet.
The counties in red represent the places which have voted for Clinton whereas those in green represent the areas which have voted for Obama.



Imputing missing values

1. Check what values are missing:

```
#check missing values
countNAs <- function (v) sum(ifelse(is.na(v),1,0))

elect.countNAs <- sapply(elect.df, countNAs)

elect.countNAs[elect.countNAs != 0]
```

TotalVote	1131
Clinton	1131
Obama	1131
Black	80
Asian	94
AmericanIndian	99
HighSchool	1
Bachelors	1
Poverty	1
IncomeAbove75K	2
MedianIncome	1
AverageIncome	30
UnemployRate	1
ManfEmploy	293
SpeakingNonEnglish	1
Medicare	1
MedicareRate	1
SocialSecurity	1
SocialSecurityRate	1
RetiredWorkers	1
Disabilities	8
DisabilitiesRate	8
Homeowner	2
SameHouse1995and2...	1
LandArea	1
FarmArea	87
ObamaWin	1131
ObamaWinNr	1131

   So all the attributes except TotalVote, Clinton, Obama, Obama Win and ObamaWinNr have missing values.

2. Missing values for AverageIncome are replaced by the MedianIncome for that same record, whereas missing values for"Black","Asian","AmericanIndian","ManfEmploy", "Disabilities","DisabilitiesRate","FarmArea" are replaced by 0.

```
# 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]])}
```

3. We checked again the missing values and realised that there still remain several attributes with 1 or 2 missing values. It turns out that all these final missing values are in 2 records. Therefore we removed these records entirely and checked again.

```
countNAs <- function (v) sum(ifelse(is.na(v),1,0))

elect.countNAs <- sapply(elect.df, countNAs)

elect.countNAs[elect.countNAs != 0]
```

TotalVote	1131
Clinton	1131
Obama	1131
HighSchool	1
Bachelors	1
Poverty	1
IncomeAbove75K	2
MedianIncome	1
AverageIncome	1
UnemployRate	1
SpeakingNonEnglish	1
Medicare	1
MedicareRate	1
SocialSecurity	1
SocialSecurityRate	1
RetiredWorkers	1
Homeowner	2
SameHouse1995and2...	1
LandArea	1
ObamaWin	1131
ObamaWinNr	1131

4. It turns out that all these final missing values are in 2 records. Therefore, we removed these records entirely and checked again.

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

```
countNAs <- function (v) sum(ifelse(is.na(v),1,0))

elect.countNAs <- sapply(elect.df, countNAs)

elect.countNAs[elect.countNAs != 0]
```

TotalVote	1130
Clinton	1130
Obama	1130
ObamaWin	1130
ObamaWinNr	1130

Convert data type

Converted the “ElectionDate” column to the “Date” data type

```
In [176]:

elect.df$ElectionDate <- as.Date(elect.df$ElectionDate,
                                 format="%m/%d/%Y")
elect.df$ElectionDate
```

2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-08
2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-19
2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19
2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-26	2008-01-26	2008-01-26	2008-01-26
2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26
2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26
2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26

Create datasets

Created known and unknown vote data. The known dataset represents the counties that voted. It would be split into train (to build models) and test (to evaluate models). The unknown vote data set represents a set of data records for which the target attribute is unknown as those counties did note vote yet. Prediction on the unknown data would be made on section 4.

Known vote dataset:

In [177]:
```
elect.df.known <- elect.df[elect.df$ElectionDate <
                           as.Date("2/19/2008", format = "%m/%d/%Y"), ]
elect.df.known
```

County	State	Region	FIPS	ElectionDate	ElectionType	TotalVote	Clinton	Obama	MalesPer100Females	⋯	RetiredWorkers	Disabilities
Adair	IA	Midwest	19001	2008-01-03	Caucuses	75	22	24	96.7	⋯	1230	90
Adams	IA	Midwest	19003	2008-01-03	Caucuses	50	18	7	96.8	⋯	720	78
Allamakee	IA	Midwest	19005	2008-01-03	Caucuses	80	25	33	104.5	⋯	2245	186
Appanoose	IA	Midwest	19007	2008-01-03	Caucuses	60	17	10	94.0	⋯	2050	453
Audubon	IA	Midwest	19009	2008-01-03	Caucuses	48	16	17	94.7	⋯	1150	79
Benton	IA	Midwest	19011	2008-01-03	Caucuses	80	23	23	98.4	⋯	3105	269
Black Hawk	IA	Midwest	19013	2008-01-03	Caucuses	420	117	179	92.7	⋯	14960	2835
Boone	IA	Midwest	19015	2008-01-03	Caucuses	140	49	43	95.9	⋯	3490	359
Bremer	IA	Midwest	19017	2008-01-03	Caucuses	100	28	35	93.5	⋯	3280	170
Buchanan	IA	Midwest	19019	2008-01-03	Caucuses	150	47	53	98.9	⋯	2485	277

Unknown vote dataset:

In [178]:
```
elect.df.unknown <- elect.df[elect.df$ElectionDate >=
                             as.Date("2/19/2008", format = "%m/%d/%Y"), ]
elect.df.unknown
```

	County	State	Region	FIPS	ElectionDate	ElectionType	TotalVote	Clinton	Obama	MalesPer100Females	⋯	RetiredWorkers	Disabiliti
1738	Hawaii	HI	West	15001	2008-02-19	Caucuses	NA	NA	NA	100.0	⋯	19245	3796
1739	Honolulu	HI	West	15003	2008-02-19	Caucuses	NA	NA	NA	99.1	⋯	102660	16523
1740	Kalawao	HI	West	15007	2008-02-19	Caucuses	NA	NA	NA	76.2	⋯	NA	NA
1741	Maui	HI	West	15009	2008-02-19	Caucuses	NA	NA	NA	100.9	⋯	13855	1570
1742	Adams	WI	Midwest	55001	2008-02-19	Primary	NA	NA	NA	116.2	⋯	4085	440
1743	Ashland	WI	Midwest	55003	2008-02-19	Primary	NA	NA	NA	97.5	⋯	2275	373
1744	Barron	WI	Midwest	55005	2008-02-19	Primary	NA	NA	NA	97.8	⋯	6695	821
1745	Bayfield	WI	Midwest	55007	2008-02-19	Primary	NA	NA	NA	103.0	⋯	2570	257
1746	Brown	WI	Midwest	55009	2008-02-19	Primary	NA	NA	NA	98.8	⋯	22210	3422
1747	Buffalo	WI	Midwest	55011	2008-02-19	Primary	NA	NA	NA	101.4	⋯	2050	204

3. 2 Data Splitting (R)

In order to build and evaluate models in R, we splitted the known vote data set into training and test sets.

1. We first found the number of rows in the known vote data set.
2. Then we set the seed (201) for a random sample.
3. We randomly sampled 75% of the row indices in the known dataset. This sample is represented by a list "rowIndicesTrain" of row numbers.
4. Then, we split the known set into the training set and test set using these indices.

```
# 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 75% of the row indices in the known dataset
rowIndicesTrain <- sample(1:nKnown,
                          size = round(nKnown*0.75),
                          replace = FALSE)
```

```
# Split the known 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, ]
```

Section 4: Generate and Test Prediction Models

To predict the people who are most likely to vote for Obama, we have constructed 3 predictive models: a Linear Regression and a Backward Stepwise Linear Regression with the most correlated attributes and a Classification tree.
We measured the accuracy rate as the percentage of wins out of actual wins and the percentage of losses out of actual losses using R's confusion matrix.

4.1. Prediction models

Accuracy

Accuracy	Model
0.69	linear regression
0.69	Stepwise with most correlated variables
0.77	Classification tree

Description

Model	Attributes and justification	Accuracy rate	Model output, insights	Steps that have been done to improve the model	Possible steps to improve the model
Linear Regression	The top 10 most correlated attributes with ObamaWin (see 1.2.)	69%	-The output offers insights about the significance, estimate, standard error and t value for each variable used. -The model predictions are continuous between 0 and 1 and not classes. Therefore, we had to round them to the appropriate class.	-	-Keep only the variables with the highest significance and estimates (Black, Income Above 75K) -Take out the variables that are correlated between each other (for example take out black and keep white, see 2.3)
Backward Stepwise	The top 10 most correlated attributes with ObamaWin (see 1.2.)	69%	-We can see from the model output that we eliminated the variables that least improve the in-sample error rate, so we can see that they are not that significant in predicting ObamaWin (*Medicare Rate, Age Below 35, Social Security Rate)* -The model predictions are continuous between 0 and 1. Therefore, we had to round them to the appropriate class.	-	-Keep only the variables with the highest significance and estimates and remove the variables highly correlated with each other's. - we could have removed *White;* it is the least significant and it is highly correlated with *Black*
Classification tree	We started with using the top 10 most correlated attributes with ObamaWin (see 1.2.) and they gave an error rate of 75%.	77%	-See the output of the model below. Not only do we have insights on 4 of the most important predictors in order (Black is the best, Highschool is the second, …) but we also know that if the % of black is less than 25, Obama loses or if the % of high school is less than 78 Obama losses. -See the second output below. The predictions were made in % for each class so we choose the one with the highest value.	-We removed 5 of the attributes using the correlation plot in 2.3., as they were highly correlated with other attributes used as predictors. This improved the error rate by 2% The final attributes are: *Black, Highschool, MedianIncome, MedicareRate and AgeBellow35*. -We found the best number of splits for the tree by plotting the error for different cp values.	-

Linear Regression output

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -1.864e+00  4.080e-01  -4.569 5.36e-06 ***
Black                2.303e-02  2.148e-03  10.723  < 2e-16 ***
White                6.850e-03  2.192e-03   3.125  0.00182 **
Bachelors            1.352e-02  2.909e-03   4.649 3.68e-06 ***
SocialSecurityRate -6.806e-06  7.124e-06  -0.955  0.33959
MedicareRate        -1.371e-07  4.018e-06  -0.034  0.97279
Age65andAbove       -8.789e-03  7.855e-03  -1.119  0.26340
HighSchool           1.618e-02  2.520e-03   6.420 1.91e-10 ***
IncomeAbove75K      -2.915e-02  5.793e-03  -5.032 5.54e-07 ***
MedianIncome         1.773e-05  4.144e-06   4.278 2.03e-05 ***
AgeBelow35          -1.752e-04  4.215e-03  -0.042  0.96686
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

1	0.480379533960963
6	0.570385151037576
7	0.641335803020555
11	0.441244315584941
27	0.363624111606389
31	0.559627628773246
35	0.408615482321577
38	0.536368567025936
41	0.564054668698223
43	0.413942536633769
44	0.538457017322972
46	0.485406990017553
49	0.371352239583446
50	0.547988236878845
51	0.614824525845139
64	0.441023651311812
66	0.452243858126761

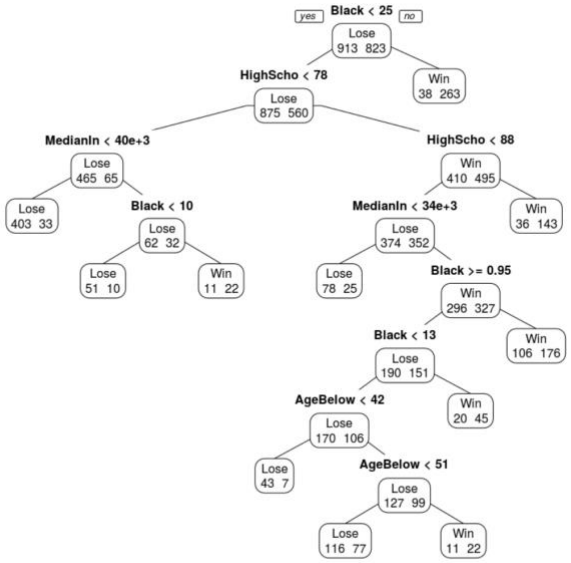
Stepwise backward output

```
Start:  AIC=-2282.12
ObamaWinNr ~ Black + White + Bachelors + SocialSecurityRate +
    MedicareRate + Age65andAbove + HighSchool + IncomeAbove75K +
    MedianIncome + AgeBelow35

                    Df Sum of Sq    RSS     AIC
- MedicareRate       1    0.0002 221.84 -2284.1
- AgeBelow35         1    0.0003 221.84 -2284.1
- SocialSecurityRate 1    0.1568 222.00 -2283.2
- Age65andAbove      1    0.2151 222.06 -2282.9
<none>                            221.84 -2282.1
- White              1    1.6776 223.52 -2274.3
- MedianIncome       1    3.1448 224.99 -2265.8
- Bachelors          1    3.7139 225.56 -2262.5
- IncomeAbove75K     1    4.3511 226.20 -2258.8
- HighSchool         1    7.0829 228.93 -2243.2
- Black              1   19.7584 241.60 -2173.0

Step:  AIC=-2284.12
ObamaWinNr ~ Black + White + Bachelors + SocialSecurityRate +
    Age65andAbove + HighSchool + IncomeAbove75K + MedianIncome +
    AgeBelow35

                    Df Sum of Sq    RSS     AIC
- AgeBelow35         1    0.0004 221.85 -2286.1
- SocialSecurityRate 1    0.1915 222.04 -2285.0
- Age65andAbove      1    0.2475 222.09 -2284.7
<none>                            221.84 -2284.1
- White              1    1.6778 223.52 -2276.3
- MedianIncome       1    3.1542 225.00 -2267.7
- Bachelors          1    3.8283 225.67 -2263.8
- IncomeAbove75K     1    4.3591 226.20 -2260.8
- HighSchool         1    7.1451 228.99 -2244.8
- Black              1   19.7642 241.61 -2175.0

Step:  AIC=-2286.12
ObamaWinNr ~ Black + White + Bachelors + SocialSecurityRate +
    Age65andAbove + HighSchool + IncomeAbove75K + MedianIncome

                    Df Sum of Sq    RSS     AIC
- SocialSecurityRate 1    0.2085 222.05 -2286.9
- Age65andAbove      1    0.2771 222.12 -2286.5
<none>                            221.85 -2286.1
- White              1    1.6874 223.53 -2278.2
- MedianIncome       1    3.2885 225.13 -2269.0
- Bachelors          1    3.8309 225.68 -2265.8
- IncomeAbove75K     1    4.3606 226.21 -2262.8
- HighSchool         1    7.1585 229.00 -2246.8
- Black              1   19.8786 241.72 -2176.4

Step:  AIC=-2286.89
ObamaWinNr ~ Black + White + Bachelors + Age65andAbove + HighSchool +
    IncomeAbove75K + MedianIncome

                   Df Sum of Sq    RSS     AIC
<none>                          222.05 -2286.9
- White             1    1.5439 223.60 -2279.9
- Age65andAbove     1    3.3334 225.39 -2269.5
- MedianIncome      1    3.5686 225.62 -2268.1
- Bachelors         1    4.0496 226.10 -2265.4
- IncomeAbove75K    1    4.4720 226.53 -2262.9
- HighSchool        1    7.9607 230.01 -2243.0
- Black             1   19.6710 241.72 -2178.4
```

```
Call:
lm(formula = ObamaWinNr ~ Black + White + Bachelors + Age65andAbove +
    HighSchool + IncomeAbove75K + MedianIncome, data = elect.df.training)

Residuals:
     Min       1Q   Median       3Q      Max
-1.17344 -0.34302 -0.05882  0.38173  1.00453

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)    -1.947e+00  2.353e-01  -8.276 3.14e-16 ***
Black           2.278e-02  2.128e-03  10.707  < 2e-16 ***
White           6.477e-03  2.159e-03   3.000  0.00276 **
Bachelors       1.381e-02  2.843e-03   4.858 1.33e-06 ***
Age65andAbove  -1.529e-02  3.469e-03  -4.407 1.13e-05 ***
HighSchool      1.674e-02  2.457e-03   6.811 1.48e-11 ***
IncomeAbove75K -2.947e-02  5.774e-03  -5.105 3.80e-07 ***
MedianIncome    1.834e-05  4.022e-06   4.560 5.59e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4142 on 1294 degrees of freedom
Multiple R-squared:  0.315,     Adjusted R-squared:  0.3112
F-statistic: 84.99 on 7 and 1294 DF,  p-value: < 2.2e-16
```

1	0.45915101818686
6	0.566826391544139
7	0.640563630203552
11	0.429771420033102
27	0.354209666838106
31	0.558431695259838
35	0.40370380263824
38	0.538182904488333
41	0.55646051317139
43	0.407766977496232
44	0.538461545705053
46	0.482549328614769
49	0.365737240478119
50	0.546691505257223
51	0.61578590608276
64	0.442581577254267
66	0.438637219171519

Classification tree output

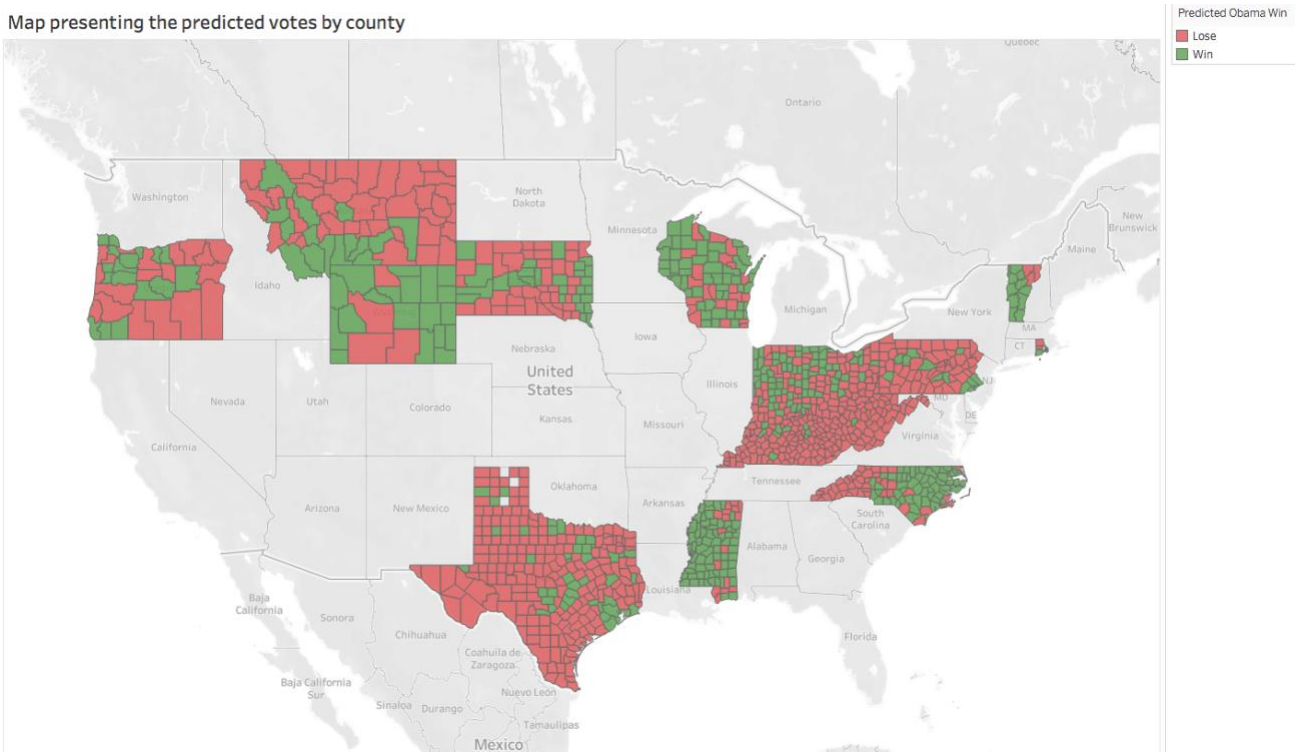


	Lose	Win
1	0.2011173	0.7988827
6	0.2011173	0.7988827
7	0.6010363	0.3989637
11	0.6010363	0.3989637
27	0.7572816	0.2427184
31	0.6010363	0.3989637
35	0.3758865	0.6241135

4.2. Best prediction model

The classification tree has the best accuracy rate 77%. There are 1130 counties that did not vote yet. This means the model will accurately predict the election results for 870 counties (77% x 1130).

We used the model for these predictions in R, saved the csv file and uploaded it in Tableau to display them in a map.

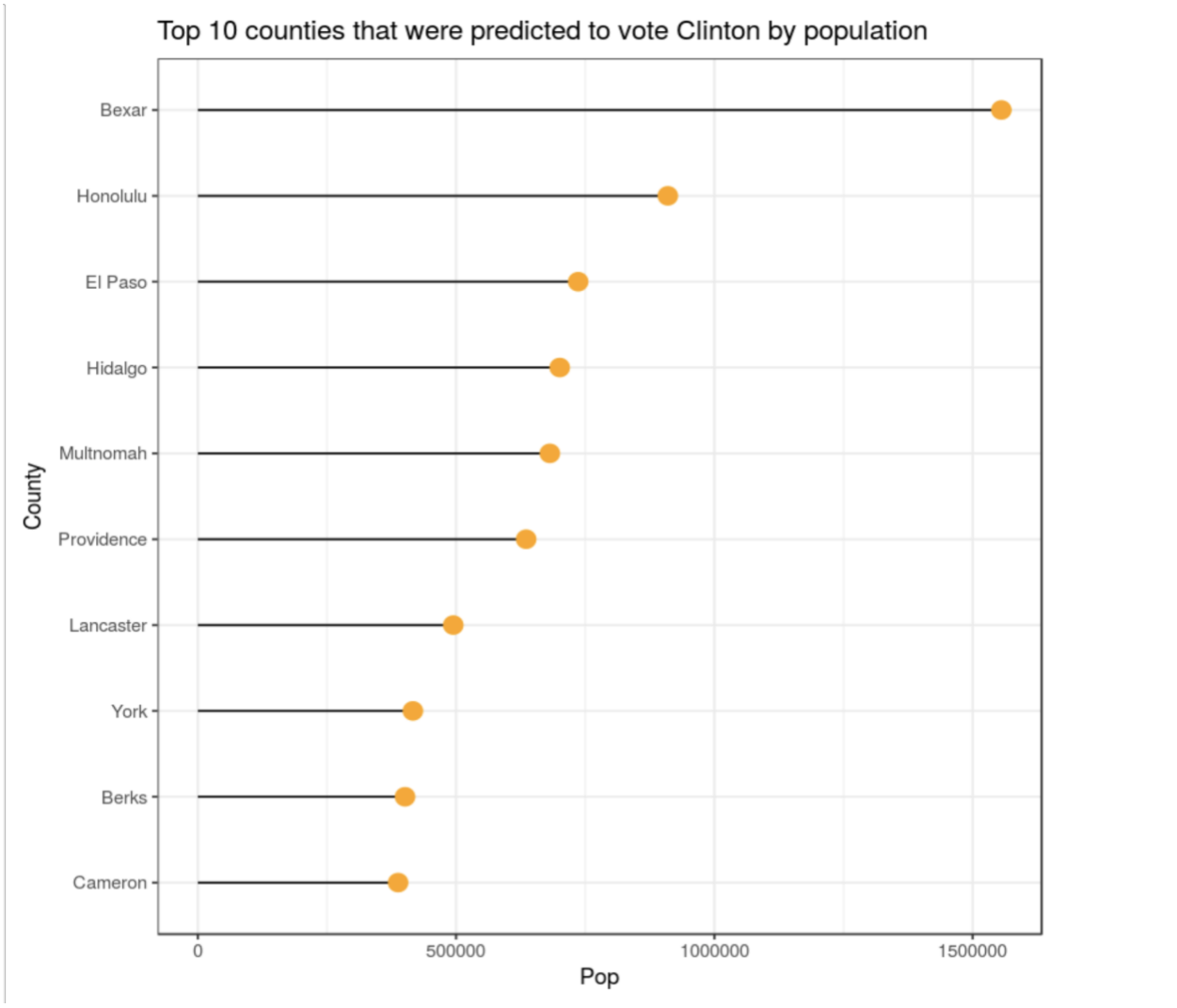


Section 5: Conclusions and Recommendations

To conclude, young people, black minorities and people with higher levels of income are more likely to vote for Obama.

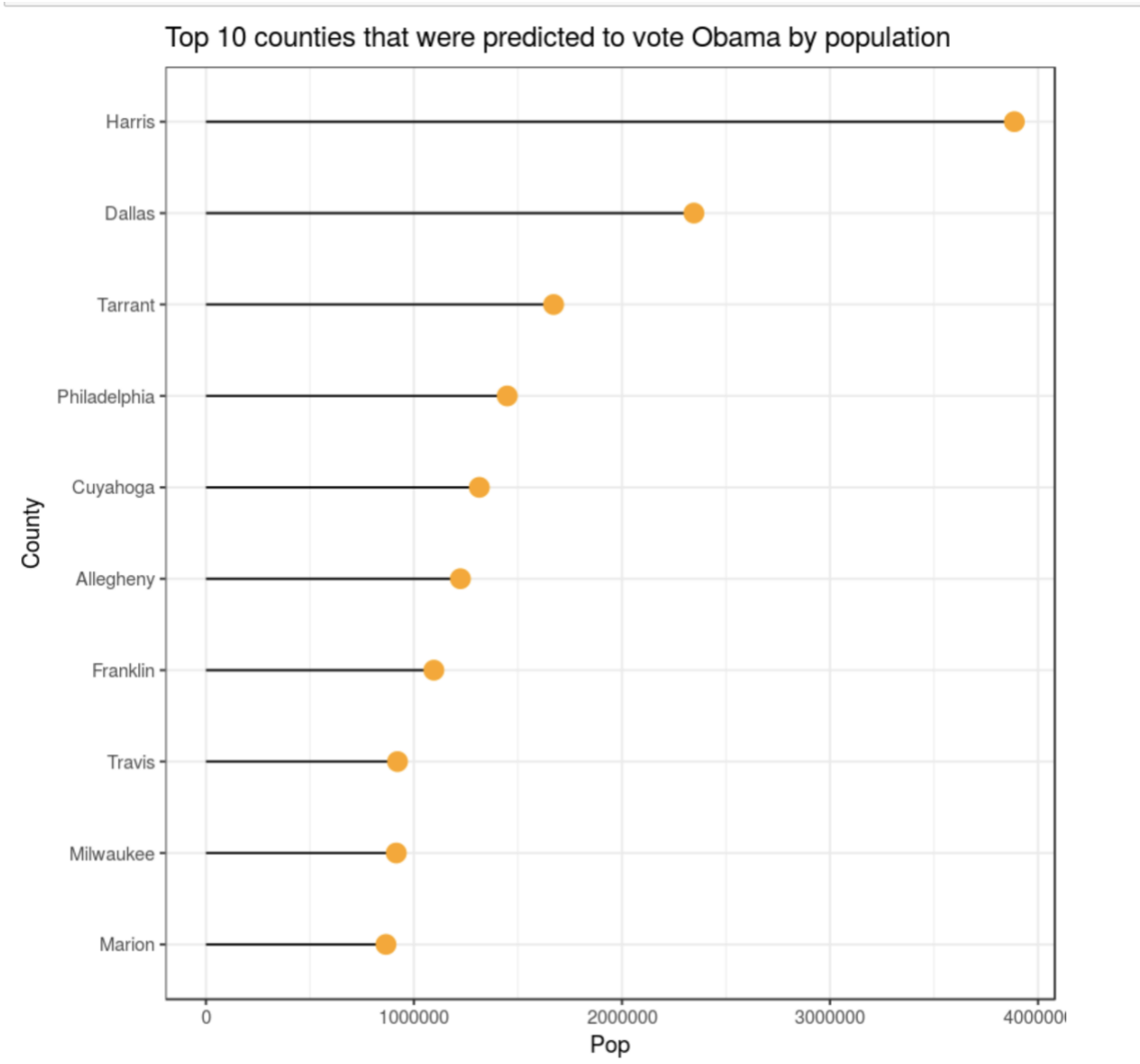
Since Clinton won more votes from poorer people, aged 65 and above, and people that are white, to win the election, we recommend Obama to focus his campaign on areas that gather these attributes.

To help Obama find the counties to focus on, we looked at the top 10 most populated counties that we predicted (with the classification tree) that will vote for Clinton.



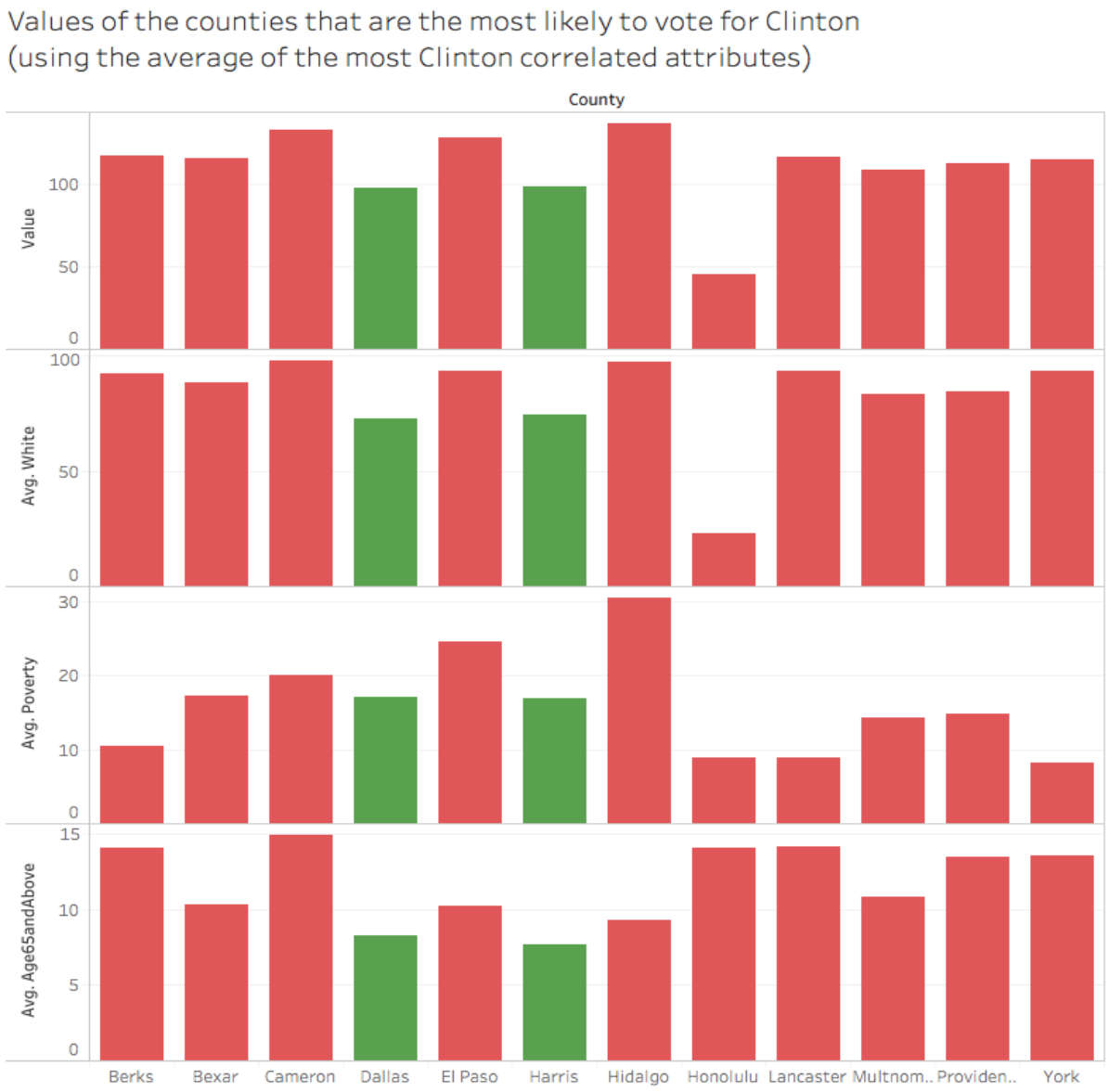
(R visual)

We also looked at the top 10 most populated counties that we predicted (with the classification tree) will vote for Obama.



(R visual)

Thus, we chose the first two counties (Harris and Dallas in green) to compare with Clinton's top 10 counties (in red) on the bar plot below.



(Tableau)

All the values used above are Clinton correlated attributes. The two green counties values are lower compared to those in red which proves the impact of the attributes on the results.

Therefore, we recommend Obama to improve his election campaign by focusing on 3 specific groups:

- **white people in every red county except Honolulu**
- **poverty in Hidalgo, El Paso, Cameron**
- **people that are aged 65 and above in the counties in red**

II - NICU Case Study

Section 1: US Births Data

For this case study, the US Births data and the NICU data have been used. The US Births data outlines the number of live births in each month and year.

Tableau - The join functionality as well as the DATEPARSE function were used to merge the two datasets.

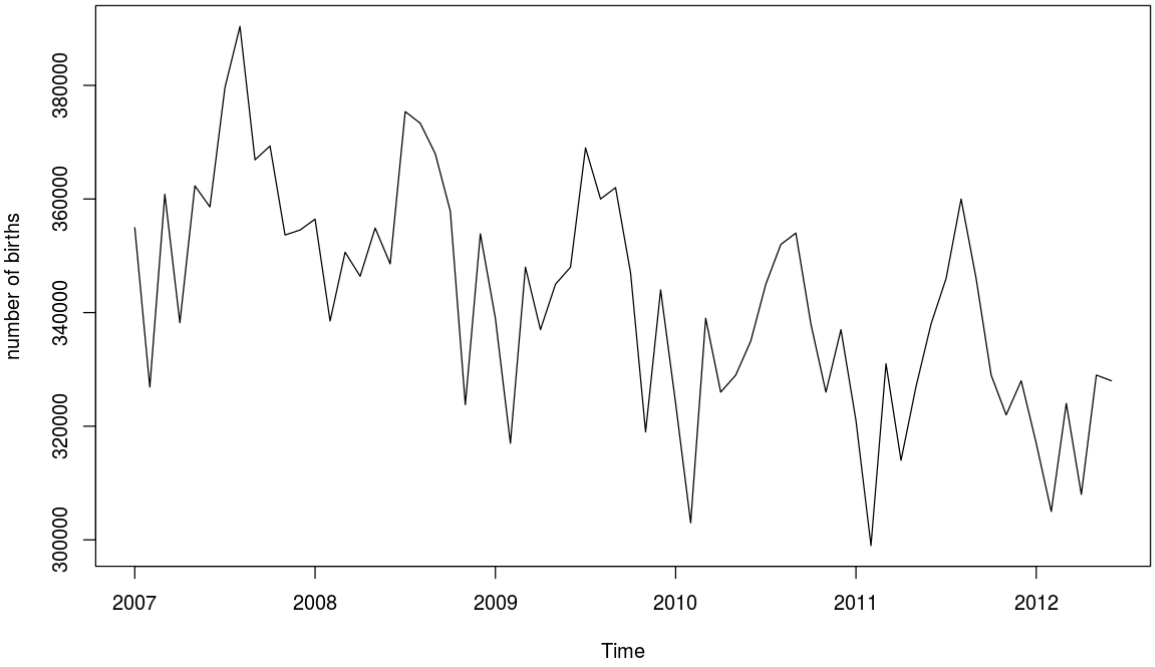
R- separating a column into multiple columns

The Yr_Mo data was first converted into a date format (4). This was then splited into three columns, Year, Month and Date using the code below:

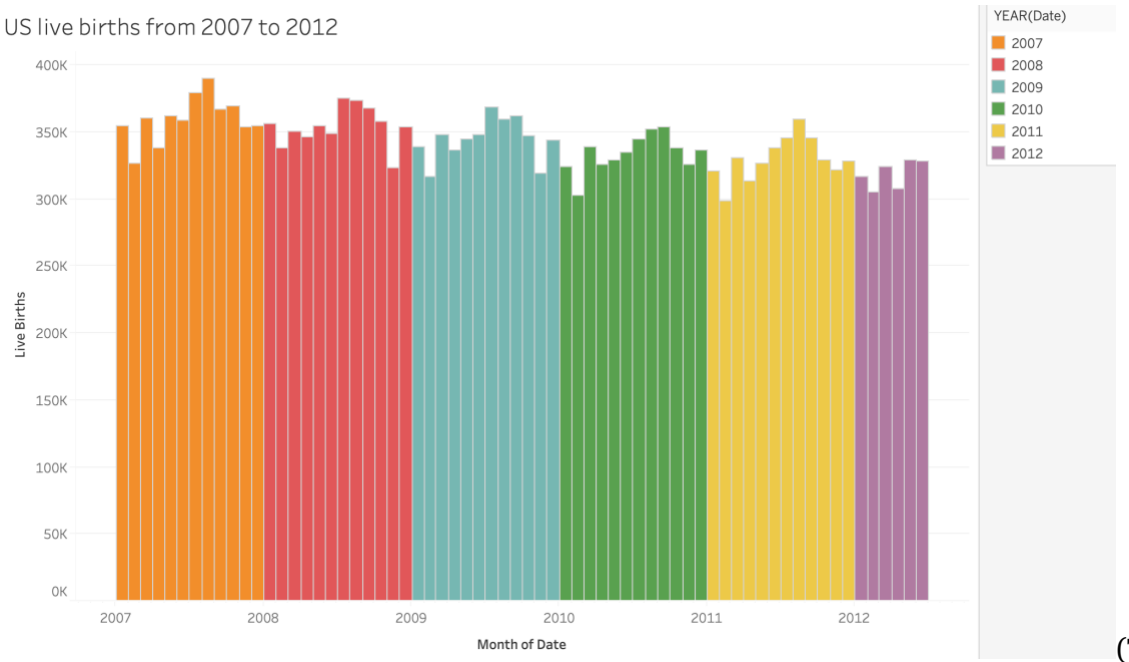
```
library(tidyr)
library(lubridate)

birth.df_2 <- separate(birth.df, Date, c("Year", "Month","Date"))
birth.df_2
```

Yr_Mo	Live.Births	Year	Month	Date
200701	354943	2007	01	01
200702	326891	2007	02	01
200703	360828	2007	03	01
200704	338224	2007	04	01
200705	362319	2007	05	01
200706	358606	2007	06	01
200707	379616	2007	07	01
200708	390378	2007	08	01

Section 2: Visuals for Seasonality Patterns

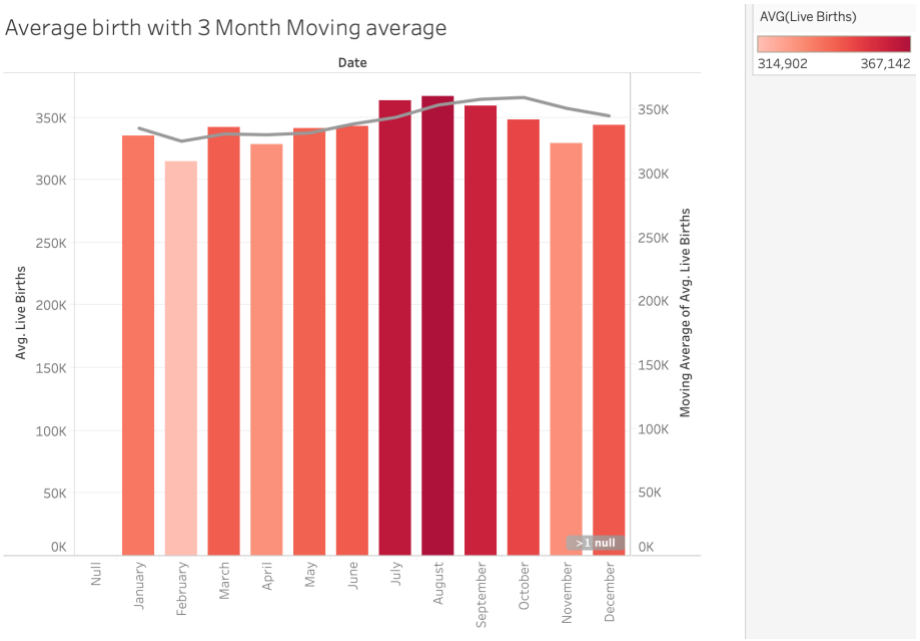


(R)



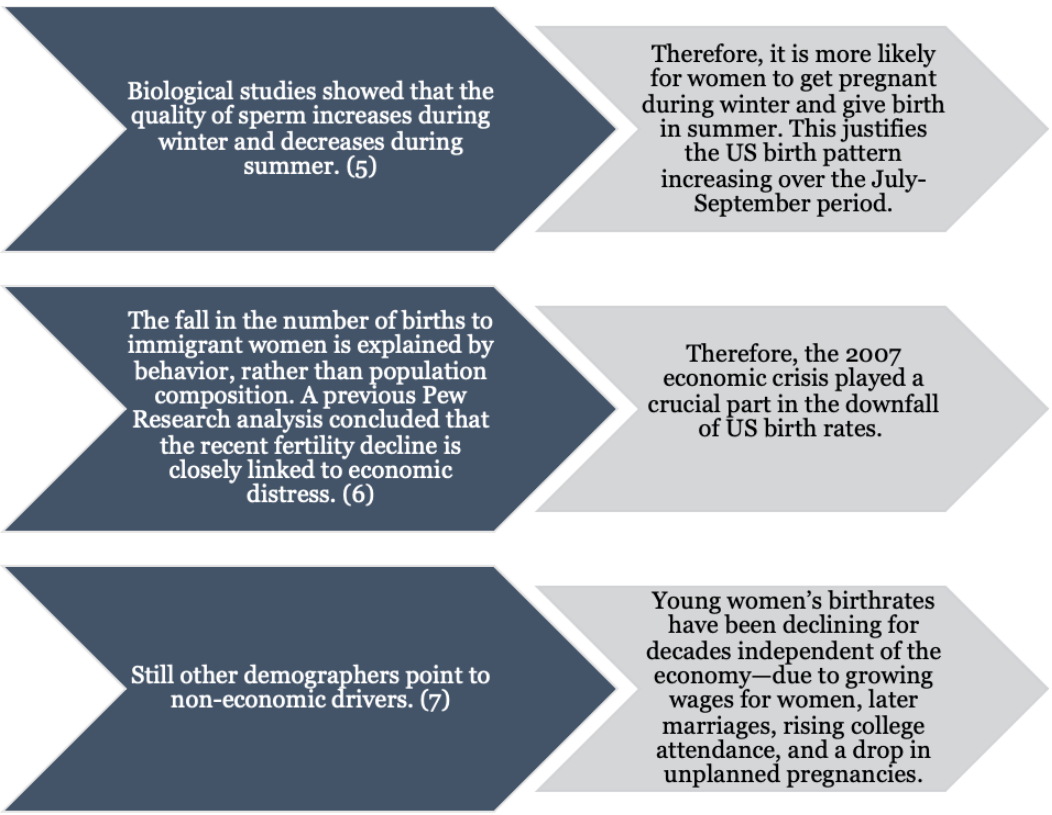
(Tableau)

These time series graphs show the number of live births between 2007 and 2012. Overall, there is a decrease in the number of births as the year progresses. A noticeable rise can be seen during July and August. However, there is a decrease in the beginning of the year and at the end of the year.



(Tableau)

For a better understanding of the situation, we aggregated the births-per-month and added a three month moving average. As expected, it confirms the increase in births over the July-September period, with the start and end of the years being the lowest.



Section 3: AAN and AAA Model

3.1 Best model selection

To forecast the number of live births in the US up to February 2013, AAN model and AAA model have been used.

The AAN model

```
RMSE =   15812.79
```
(R)

The AAA model

```
RMSE =   6307.65
```
(R)

Since the AAA model has a lower RMSE error rate than the AAN model, the AAA model is the better fit for the US Births data.

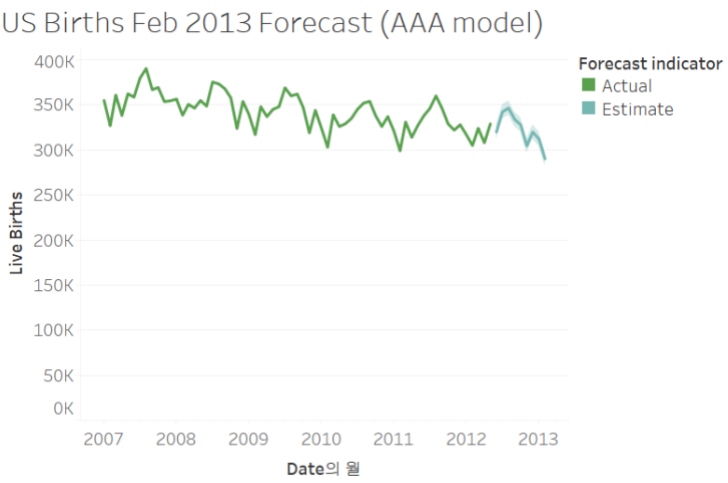
3.2 Forecasts using AAA model

The number of live US births up to February 2013 has been predicted using the AAA model as a result and we generated forecast plots in both Tableau and R with 80% confidence levels.

Tableau:

Forecast indicator: **Estimate**
Month of Date: **2013년 2월**
Live Births: **290,128**

(Tableau)



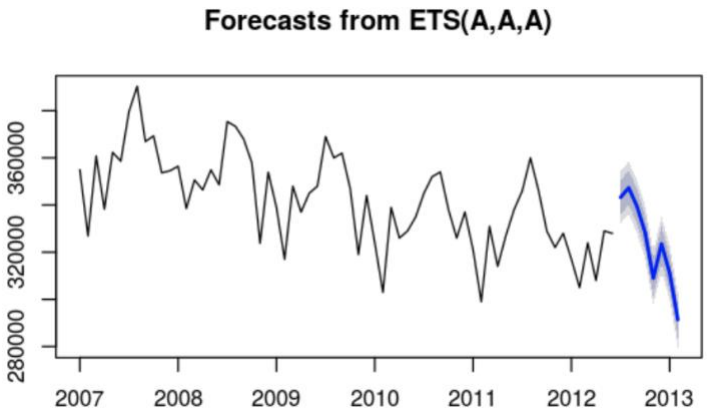
(Tableau)

Following the same seasonality pattern, there is a slight increase in the number of live births during the summer season. However, the overall trend shows a decrease in the number of births as the year progresses and the predicted number of births is to decline further in February 2013, with the value 290,128.

R:

```
February 2013: mean births =  291394.2
          upper 80% confid. births =  299042.4
```

(R)

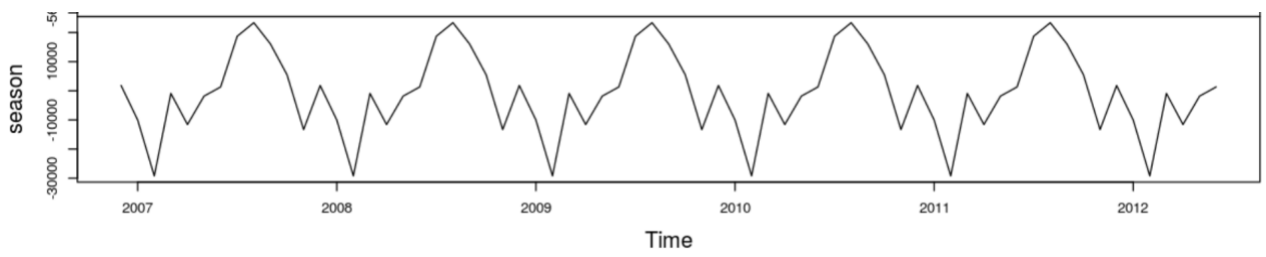


(R)

A decline can be seen in the number of US Births from early 2008, where it peaked, to late 2012. Although there was a slight rise at the end of 2012, the US births are predicted to fall. Analysing the forecast, February 2013 would see 291.394 births.

Section 4: Seasonality Patterns Comparison

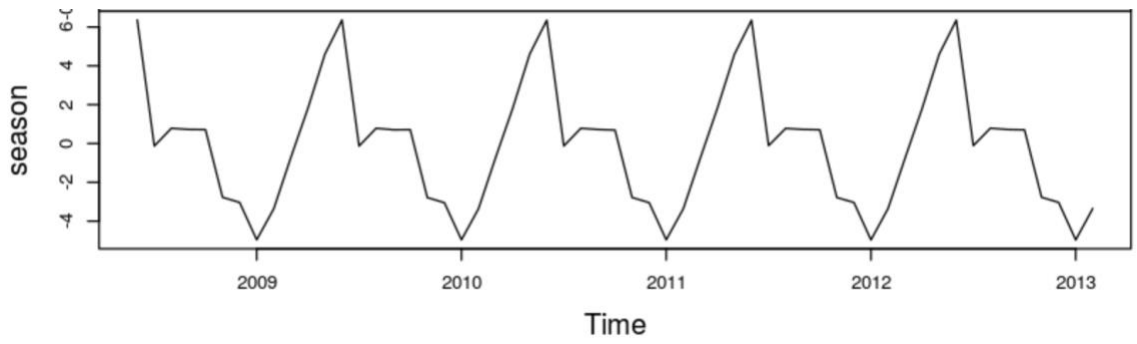
Seasonality US Births



(R)

The US Birth rate peaks in July of each year, while it falls to a low point during the end of the winter.

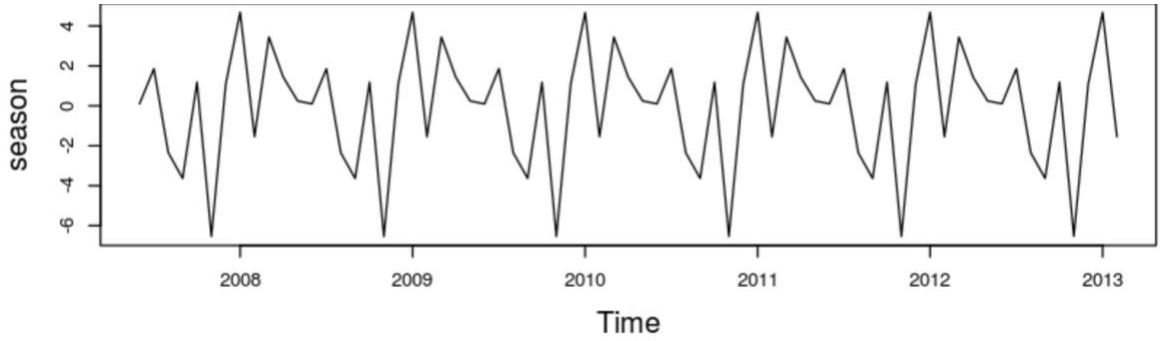
Seasonality NICU ALOS



(R)

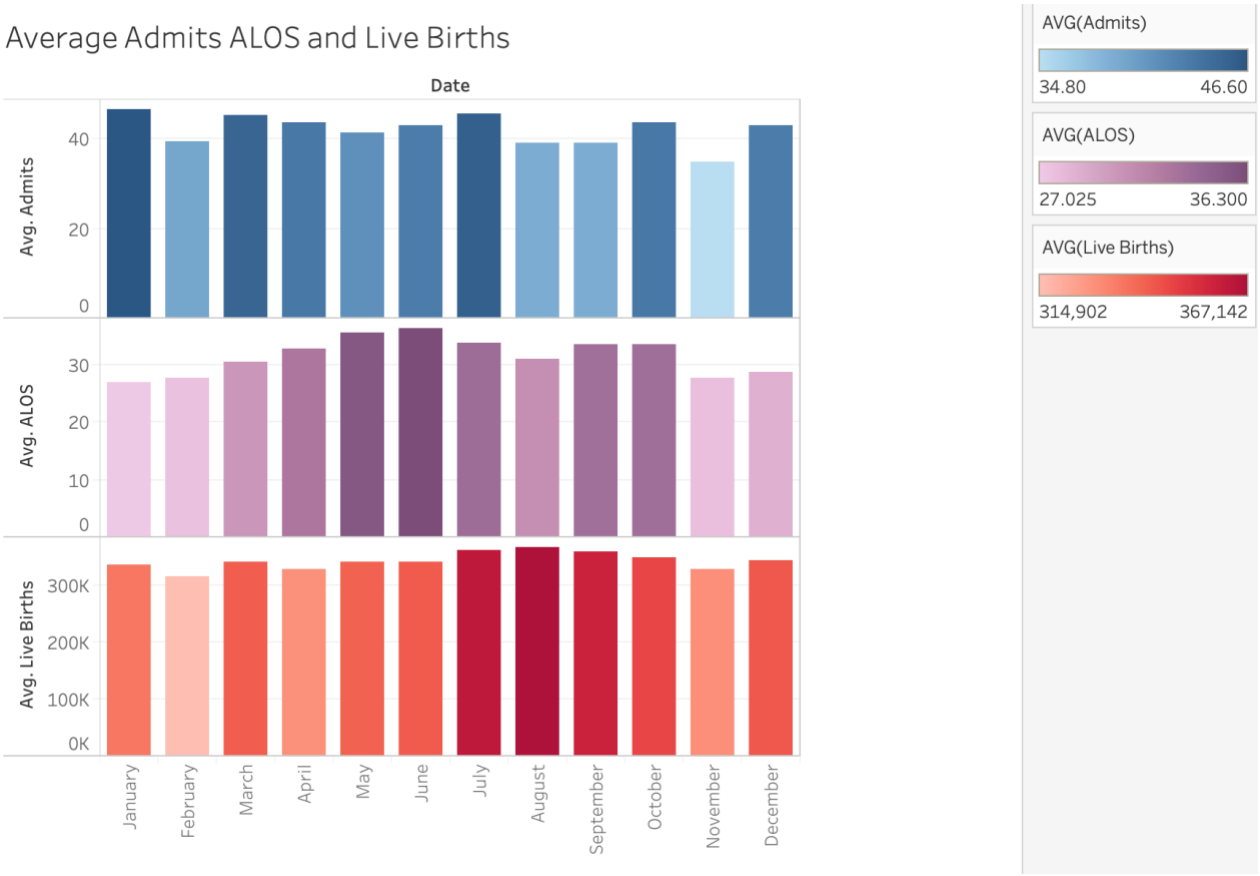
The NICU average length of stay follows the same pattern as the US birth rates, peaking in summer and having its lowest point in late December, early January.

Seasonality NICU Admissions



(R)

The NICU Admissions rate has its lowest point in autumn, October-November and sees a quick rise, peaking in late December, followed by a decrease throughout the new year until it reaches the low point again. It does not follow the trend of the other two, both because of the number of births and the cold weather which impact citizens' health, leading to more admissions.

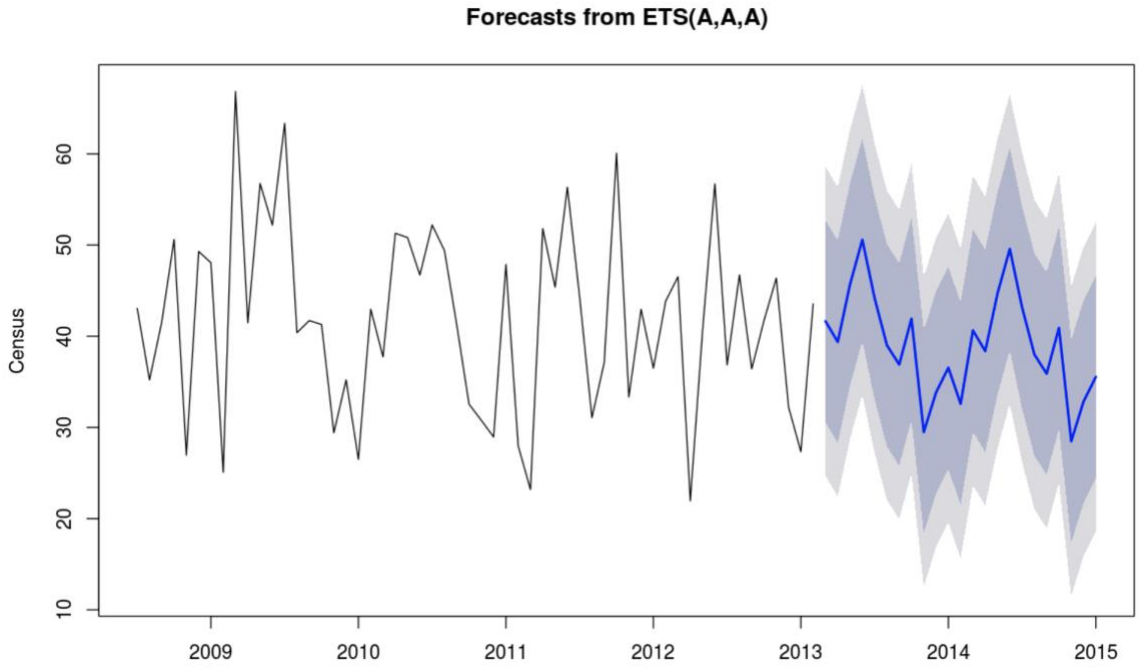


(Tableau)

By joining the dataset, we can see that the spike in US births are preceded by the rise in NICU admissions and length of stay.
We can explain it by the fact that the main reason NICU's receive babies is for prematurity.
Hence, admissions and length of stay increase during the period preceding the rise of US birth lives.

Section 5: Our Recommendation

Bed Occupancy Forecast 2015 (R)



Although NICU is the most important part of the hospital and the number of births tend to increase between July and September, our analysis and forecasts showed that the number of births is going to continue to fall at least upto January 2014.

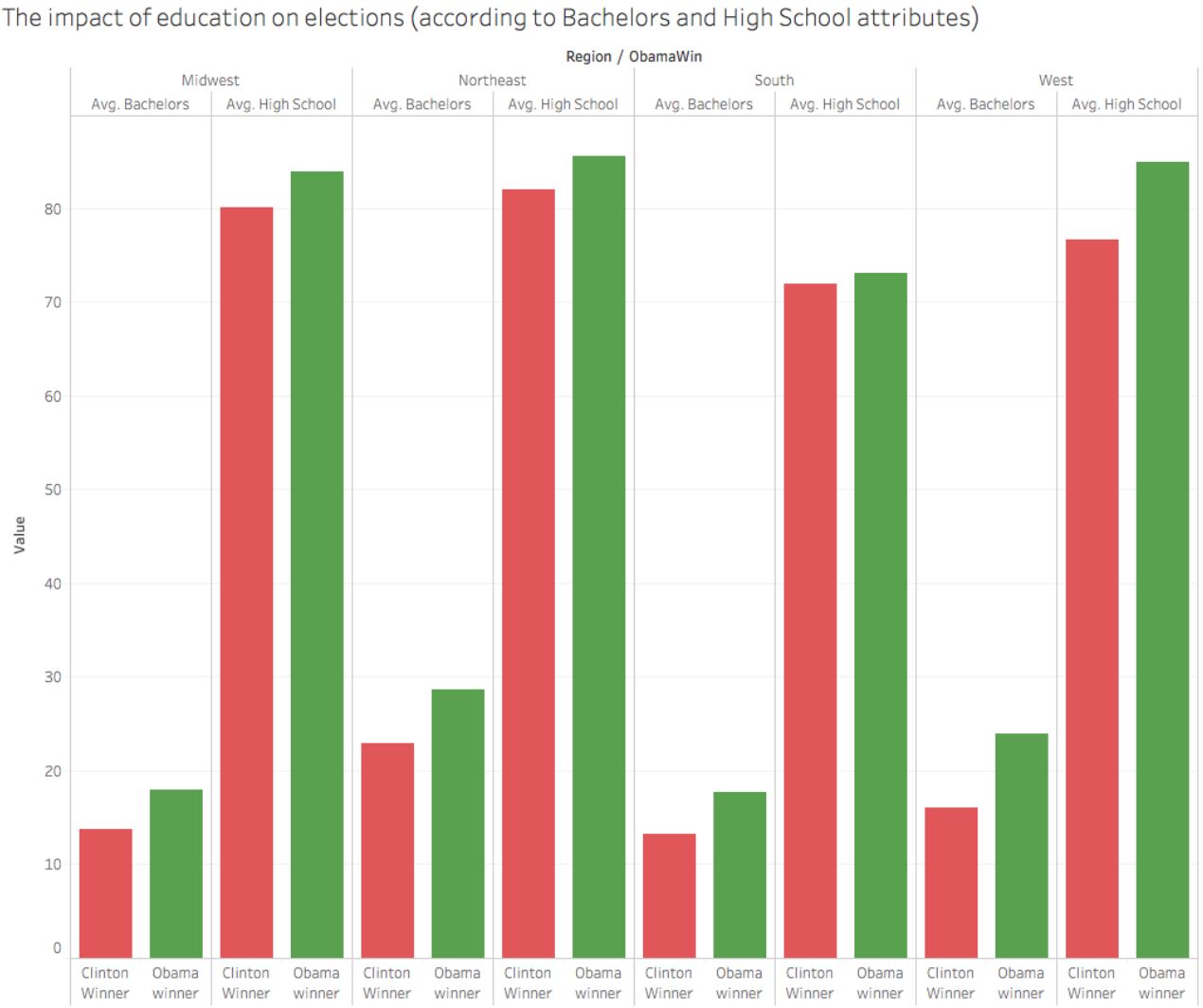
Furthermore, the bed occupancy forecast illustrates a decrease in number of beds until 2015. Therefore, we would recommend the COO of Garfield Children's Hospital to invest the money in other parts of the children's hospital rather than increasing the number of beds.

Appendix:

The most Obama correlated attributes are ethnicities, age, income and education. We decided to focus on the three last attributes. However, before doing that we've also studied the education attribute.

2.2.2.0. Education

Education is represented in the dataset by 2 attributes - percentage of people that graduated highschool and bachelors for each county. Therefore, the type is numerical. To demonstrate the positive correlation between votes for Obama and the high level of education, we studied the average of high school and bachelors



(Tableau)

All regions follow the same trend: the more educated people are, the more likely they are to vote for Obama.
On average, the level of education is always higher for people that voted for Obama compared to those that voted for Clinton.

We can also see this by regios:

<Education>

	Region	ObamaWin 0	1	na
Avg. Bachelors	Midwest	13.76	17.99	15.77
	Northeast	22.90	28.66	19.22
	South	13.21	17.65	14.17
	West	16.09	23.91	19.56
Avg. High School	Midwest	80.09	83.92	81.51
	Northeast	81.98	85.65	81.61
	South	71.94	73.11	70.67
	West	76.72	84.92	83.94

2.2.2.1. Income

ObamaWin	IncomeAbove75K
Lose	12.82738
Win	16.50535

(R)

It can be seen that voters with income over $75,000 voted for Obama.

2.2.2.2. Ethnicity

1 more aggregated table was created in tableau with different regions and 1 demographic attribute (race).

<Race>

	Region	ObamaWin 0	1	na
Avg. American Indian	Midwest	0.58	1.09	3.64
	Northeast	0.45	0.48	0.23
	South	2.19	0.36	0.72
	West	5.15	2.35	4.64
Avg. Asian	Midwest	0.47	1.04	0.69
	Northeast	2.75	2.22	1.00
	South	0.63	1.25	0.67
	West	2.60	1.85	1.81
Avg. Black	Midwest	1.73	2.93	2.34
	Northeast	6.52	5.84	3.42
	South	9.20	33.18	12.90
	West	2.28	1.35	0.50
Avg. Hawaiian	Midwest	0.03	0.03	0.02
	Northeast	0.04	0.04	0.03
	South	0.04	0.04	0.07
	West	0.19	0.15	0.58
Avg. Hispanic	Midwest	2.83	3.11	2.00
	Northeast	6.71	4.86	2.33
	South	4.61	3.40	13.71
	West	28.40	12.49	4.98
Avg. White	Midwest	96.79	94.38	92.79
	Northeast	89.18	90.48	94.56
	South	86.80	64.50	85.33
	West	88.53	93.14	90.95

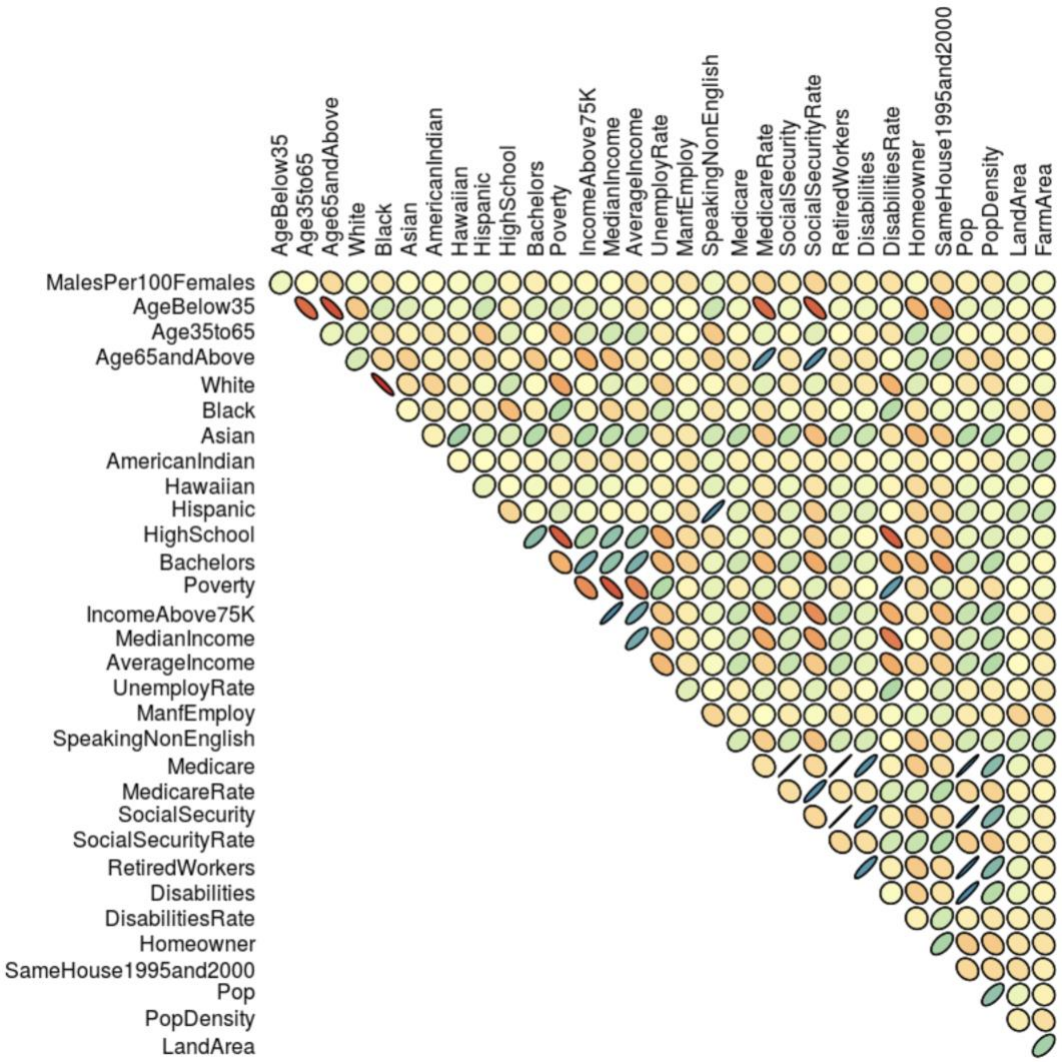
2.2.2.3. Age

Winner	AgeBelow35	Age35to65	Age65andAbove
Clinton	44.78817	39.54294	15.67415
Obama	46.82309	39.42418	13.75529

(R)

It can be seen that Obama and Clinton were drawing the same mean number of voters aged 35 to 65, however Obama edged Clinton in drawing voters under 35.

2.3. Correlations between variables



Code Appendix

Data analytics 2

Contents page

Part 1 - Obama Clinton

In [133]:
```
#Read the file
elect.df <- read.csv('Obama.csv')
```

Section 3 : Data pre-processing

3.1.2. Data manipulation: R

In [134]:
```
#Creating the ObamaWin columns both numerical and categorical


for (row in 1:nrow(elect.df)) {
     Obamav <- elect.df[row, "Obama"]
     Clintonv<- elect.df[row, "Clinton"]
    if(is.na(Obamav))
        {elect.df[row,"ObamaWin"]<-NA
         elect.df[row,"ObamaWinNr"]<-NA
        }
    else if(Obamav>Clintonv)
        {elect.df[row,"ObamaWin"]<-"Win"
        elect.df[row,"ObamaWinNr"]<-1}
    else
        {elect.df[row,"ObamaWinNr"]<-0
        elect.df[row,"ObamaWin"]<-"Lose"}
        }
```

In [135]:
```
elect.df
```

County	State	Region	FIPS	ElectionDate	ElectionType	TotalVote	Clinton	Obama	MalesPer100Females	⋯	Disabilities	DisabilitiesRat
Adair	IA	Midwest	19001	1/3/2008	Caucuses	75	22	24	96.7	⋯	90	1145
Adams	IA	Midwest	19003	1/3/2008	Caucuses	50	18	7	96.8	⋯	78	1829
Allamakee	IA	Midwest	19005	1/3/2008	Caucuses	80	25	33	104.5	⋯	186	1265
Appanoose	IA	Midwest	19007	1/3/2008	Caucuses	60	17	10	94.0	⋯	453	3315

2.1. Nature, size, source of the data

In [136]:
```
summary(elect.df[,3:42])
```

```
      Region          FIPS          ElectionDate    ElectionType
Midwest  : 814   Min.   : 1001   2/5/2008 :1128   Caucuses: 310
Northeast: 217   1st Qu.:18102   3/4/2008 : 358   Primary :2558
South    :1419   Median :30110   2/9/2008 : 196
West     : 418   Mean   :31029   5/6/2008 : 192
                 3rd Qu.:46124   2/12/2008: 158
                 Max.   :56045   5/20/2008: 156
                                 (Other)  : 680
  TotalVote          Clinton           Obama        MalesPer100Females
Min.   :     13   Min.   :     4   Min.   :     4   Min.   : 76.20
1st Qu.:    732   1st Qu.:   329   1st Qu.:   254   1st Qu.: 94.90
Median :   2330   Median :  1106   Median :   878   Median : 97.60
Mean   :  12864   Mean   :  5974   Mean   :  6178   Mean   : 99.08
3rd Qu.:   6418   3rd Qu.:  3200   3rd Qu.:  2749   3rd Qu.:100.50
Max.   :1413869   Max.   :771700   Max.   :743686   Max.   :200.90
NA's   :1131      NA's   :1131     NA's   :1131
  AgeBelow35      Age35to65     Age65andAbove       White
Min.   : 4.50   Min.   :18.60   Min.   : 2.20   Min.   :  6.70
1st Qu.:42.20   1st Qu.:38.00   1st Qu.:12.20   1st Qu.: 81.90
Median :45.40   Median :39.80   Median :14.40   Median : 93.70
Mean   :45.66   Mean   :39.59   Mean   :14.76   Mean   : 86.76
3rd Qu.:48.70   3rd Qu.:41.40   3rd Qu.:17.00   3rd Qu.: 97.50
Max.   :75.80   Max.   :79.20   Max.   :33.70   Max.   :100.00

    Black           Asian        AmericanIndian      Hawaiian
Min.   : 0.10   Min.   : 0.100   Min.   : 0.100   Min.   : 0.00000
1st Qu.: 0.60   1st Qu.: 0.300   1st Qu.: 0.200   1st Qu.: 0.00000
Median : 2.60   Median : 0.500   Median : 0.400   Median : 0.00000
Mean   : 9.90   Mean   : 1.065   Mean   : 1.689   Mean   : 0.07706
3rd Qu.:11.93   3rd Qu.: 0.900   3rd Qu.: 0.800   3rd Qu.: 0.10000
Max.   :86.00   Max.   :46.600   Max.   :91.800   Max.   :30.60000
NA's   :80      NA's   :94       NA's   :99
   Hispanic      HighSchool     Bachelors        Poverty
Min.   : 0.100   Min.   :34.7   Min.   : 4.90   Min.   : 2.60
1st Qu.: 1.200   1st Qu.:70.6   1st Qu.:11.10   1st Qu.:10.10
Median : 2.400   Median :78.7   Median :14.30   Median :13.20
Mean   : 7.349   Mean   :76.9   Mean   :16.45   Mean   :13.95
3rd Qu.: 6.800   3rd Qu.:83.6   3rd Qu.:19.30   3rd Qu.:16.90
Max.   :97.600   Max.   :97.0   Max.   :63.70   Max.   :39.40
                 NA's   :1      NA's   :1       NA's   :1
IncomeAbove75K   MedianIncome   AverageIncome    UnemployRate
Min.   : 1.50   Min.   :16868   Min.   : 5148   Min.   : 1.500
1st Qu.: 9.20   1st Qu.:32314   1st Qu.:23250   1st Qu.: 3.800
Median :11.90   Median :37272   Median :26222   Median : 4.700
Mean   :14.02   Mean   :39107   Mean   :27281   Mean   : 4.899
3rd Qu.:16.40   3rd Qu.:43430   3rd Qu.:29840   3rd Qu.: 5.700
Max.   :57.30   Max.   :98245   Max.   :93377   Max.   :15.300
NA's   :2       NA's   :1       NA's   :30      NA's   :1
  ManfEmploy      SpeakingNonEnglish    Medicare        MedicareRate
Min.   : 0.2024   Min.   : 0.400     Min.   :      7   Min.   :   60
1st Qu.: 4.7194   1st Qu.: 3.000     1st Qu.:   2162   1st Qu.:14108
Median : 8.9509   Median : 4.600     Median :   4514   Median :16863
Mean   :10.6564   Mean   : 8.617     Mean   :  13724   Mean   :17088
3rd Qu.:14.9613   3rd Qu.: 8.600     3rd Qu.:  10613   3rd Qu.:19854
Max.   :57.2291   Max.   :92.100     Max.   :1059297   Max.   :60845
NA's   :293       NA's   :1          NA's   :1         NA's   :1
SocialSecurity    SocialSecurityRate RetiredWorkers    Disabilities
Min.   :     20   Min.   : 1518      Min.   :    15   Min.   :     4.0
1st Qu.:   2548   1st Qu.:16827      1st Qu.:  1490   1st Qu.:   274.8
Median :   5360   Median :20040      Median :  3095   Median :   698.5
Mean   :  15616   Mean   :19939      Mean   :  9869   Mean   :  2381.4
3rd Qu.:  12845   3rd Qu.:23032      3rd Qu.:  7722   3rd Qu.:  1671.2
Max.   :1047590   Max.   :41943      Max.   :681035   Max.   :400125.0
NA's   :1         NA's   :1          NA's   :1        NA's   :8
DisabilitiesRate   Homeowner     SameHouse1995and2000      Pop
Min.   :  125    Min.   :19.60   Min.   :15.40        Min.   :     60
1st Qu.: 1380    1st Qu.:70.50   1st Qu.:54.30        1st Qu.:  12335
Median : 2192    Median :75.20   Median :59.40        Median :  26968
Mean   : 2693    Mean   :73.89   Mean   :58.74        Mean   :  99226
3rd Qu.: 3436    3rd Qu.:78.90   3rd Qu.:64.00        3rd Qu.:  68372
Max.   :20185    Max.   :89.60   Max.   :90.50        Max.   :9948081
NA's   :8        NA's   :2       NA's   :1
  PopDensity          LandArea        FarmArea        ObamaWin
Min.   :    0.10   Min.   :    2   Min.   :   1.0   Length:2868
1st Qu.:   19.80   1st Qu.:  434   1st Qu.:  84.0   Class :character
Median :   47.25   Median :  626   Median : 182.0   Mode  :character
Mean   :  265.18   Mean   : 1004   Mean   : 293.7
3rd Qu.:  118.03   3rd Qu.:  950   3rd Qu.: 348.0
Max.   :70190.80   Max.   :20105   Max.   :4595.0
                   NA's   :1       NA's   :87
```

```
Max.   :70190.80   Max.   :20105   Max.   :4595.0
                   NA's   :1       NA's   :87
```

In [137]:
```
unique(elect.df[c("Region")])
```

	Region
1	Midwest
100	Northeast
110	West
127	South

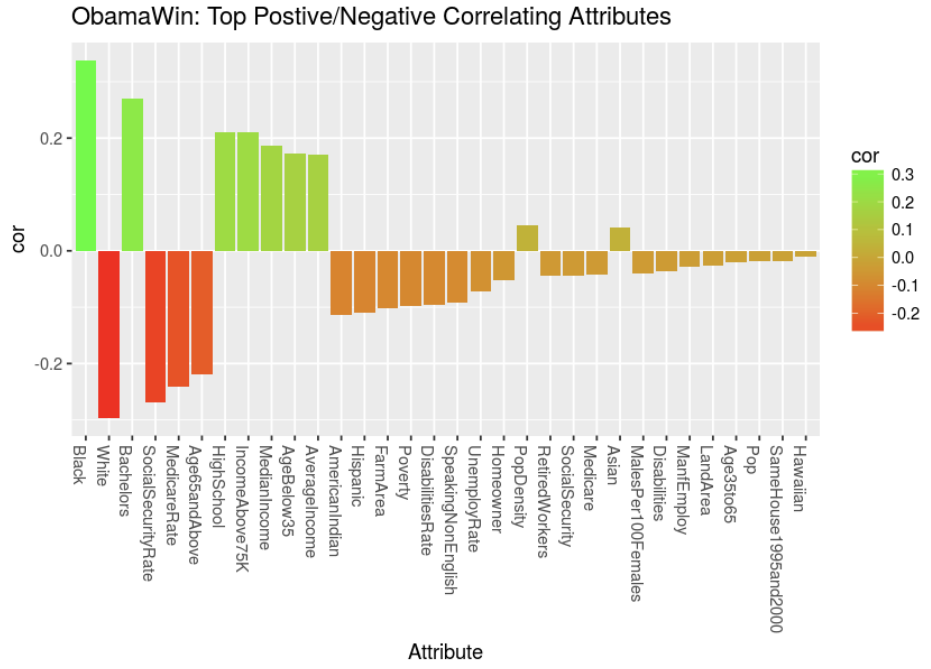
1.2. Subproblem: Electorate segmentation reserach

Correlations with ObamaWin

In [138]:
```
#top 10 correlated attributes with Obama's victory


targetCol <- which(names(elect.df)=="ObamaWinNr")
startCol <- which(names(elect.df)=="MalesPer100Females")
endCol <- which(names(elect.df)=="FarmArea")

cor.ObamaWin <- cor(elect.df[,c(targetCol, startCol:endCol)]
                    ,use="complete.obs")[-1,"ObamaWinNr"]

cor.ObamaWin.df <- data.frame(cor=cor.ObamaWin, abs.cor=abs(cor.ObamaWin),
                              row.names=names(cor.ObamaWin))

cor.ObamaWin.df <- cor.ObamaWin.df[order(-cor.ObamaWin.df$abs.cor),]

cor.ObamaWin.df[1:10,]
```

	cor	abs.cor
Black	0.3376574	0.3376574
White	-0.2958959	0.2958959
Bachelors	0.2696747	0.2696747
SocialSecurityRate	-0.2692112	0.2692112
MedicareRate	-0.2411597	0.2411597
Age65andAbove	-0.2182512	0.2182512
HighSchool	0.2105106	0.2105106
IncomeAbove75K	0.2103824	0.2103824
MedianIncome	0.1871499	0.1871499
AgeBelow35	0.1735508	0.1735508

In [139]:
```
#Top correlated attributes with Obama's victory


options(repr.plot.height=5)

library(ggplot2)

ggplot(cor.ObamaWin.df, aes(x=reorder(row.names(cor.ObamaWin.df),-abs.cor), y=cor, fill=cor)) +
    geom_col() + ggtitle("ObamaWin: Top Postive/Negative Correlating Attributes") + xlab("Attribute") +
    scale_fill_gradient(low="red", high="green") +
    theme(axis.text.x=element_text(angle=-90, hjust=0))
```

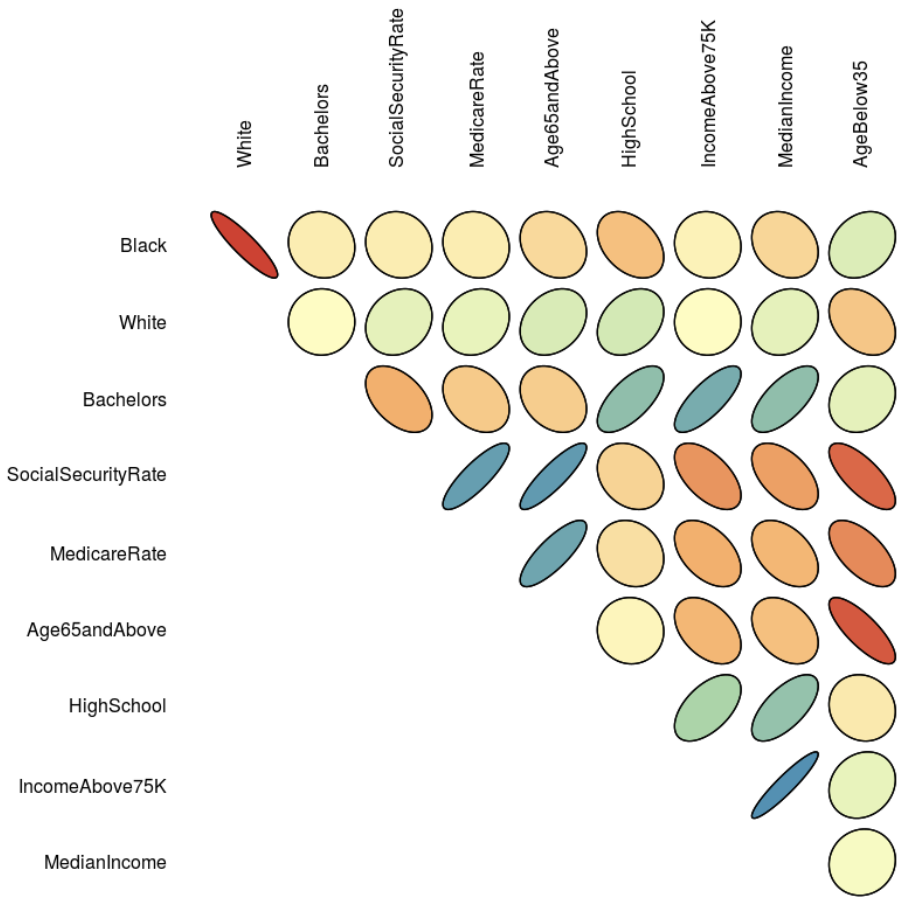


2.3. Correlations between variables

Correlations between top correlated attributes with ObamaWin

In [140]:

```
library (ellipse); library (RColorBrewer); options(repr.plot.height=10)
my_colors=colorRampPalette(brewer.pal(5, "Spectral"))(100)
data=cor(elect.df[,c("Black", "White", "Bachelors", "SocialSecurityRate",
                "MedicareRate", "Age65andAbove", "HighSchool", "IncomeAbove75K",
                "MedianIncome", "AgeBelow35")], use="complete.obs")

plotcorr(data, col=my_colors[data*50+50], mar=c(0,0,0,0),
        cex.lab=0.8, type="upper", diag=FALSE)
```



3.2.1.Data manipulation: R

Imputting missing values

In [141]:
```
#check missing values
countNAs <- function (v) sum(ifelse(is.na(v),1,0))

elect.countNAs <- sapply(elect.df, countNAs)

elect.countNAs[elect.countNAs != 0]
```

TotalVote	1131
Clinton	1131
Obama	1131
Black	80
Asian	94
AmericanIndian	99
HighSchool	1
Bachelors	1
Poverty	1
IncomeAbove75K	2
MedianIncome	1
AverageIncome	30
UnemployRate	1
ManfEmploy	293
SpeakingNonEnglish	1
Medicare	1
MedicareRate	1
SocialSecurity	1
SocialSecurityRate	1
RetiredWorkers	1
Disabilities	8
DisabilitiesRate	8
Homeowner	2
SameHouse1995and2...	1
LandArea	1
FarmArea	87
ObamaWin	1131
ObamaWinNr	1131

In [142]:
```
# 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)
```

In [143]:
```
# 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]])}
```

In [144]:
```
countNAs <- function (v) sum(ifelse(is.na(v),1,0))

elect.countNAs <- sapply(elect.df, countNAs)

elect.countNAs[elect.countNAs != 0]
```

TotalVote	1131
Clinton	1131
Obama	1131
HighSchool	1
Bachelors	1
Poverty	1
IncomeAbove75K	2
MedianIncome	1
AverageIncome	1
UnemployRate	1
SpeakingNonEnglish	1

ObamaWinNr	1131

In [145]:
```
# 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,]
```

In [146]:
```
countNAs <- function (v) sum(ifelse(is.na(v),1,0))

elect.countNAs <- sapply(elect.df, countNAs)

elect.countNAs[elect.countNAs != 0]

# We now see that all the missing data has addressed.
```

TotalVote	1130
Clinton	1130
Obama	1130
ObamaWin	1130
ObamaWinNr	1130

Covert data type

In [147]:
```
elect.df$ElectionDate <- as.Date(elect.df$ElectionDate,
                                 format="%m/%d/%Y")
elect.df$ElectionDate
```

2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03
2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-03	2008-01-08
2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-08	2008-01-19
2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19
2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-19	2008-01-26	2008-01-26	2008-01-26	2008-01-26
2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26
2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26
2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26	2008-01-26

Create known and unknown vote datasets

In [148]:
```
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"), ]
```

In [149]:
```
# We can now see now many rows there are in our known
# and unknown datasets

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

1736

1130

Section 2

2.2.2. Studying the relevant data attributes

2.2.2.2. Ethnicity

In [150]:
```
data1<-elect.df.known[elect.df.known$ObamaWin=="Win",]

data2<-elect.df.known[elect.df.known$ObamaWin=="Lose",]
```

In [151]:
```
d1<-aggregate(data.frame(Nr_Obama_Wins = data1$ElectionDate), list(ElectionDate = data1$ElectionDate), length)
```

In [152]:
```
d2<-aggregate(data.frame(Nr_Clinton_Wins = data2$ElectionDate), list(ElectionDate = data2$ElectionDate), length)
```

In [153]:
```
#created a table that counts each candidate's number of wins in each election date

d3<-merge(d1,d2,by="ElectionDate")
d3
```

ElectionDate	Nr_Obama_Wins	Nr_Clinton_Wins
2008-01-03	51	48
2008-01-08	5	5
2008-01-19	11	6
2008-01-26	44	2
2008-01-29	9	58
2008-02-05	478	649
2008-02-09	97	99
2008-02-10	12	4
2008-02-12	116	42

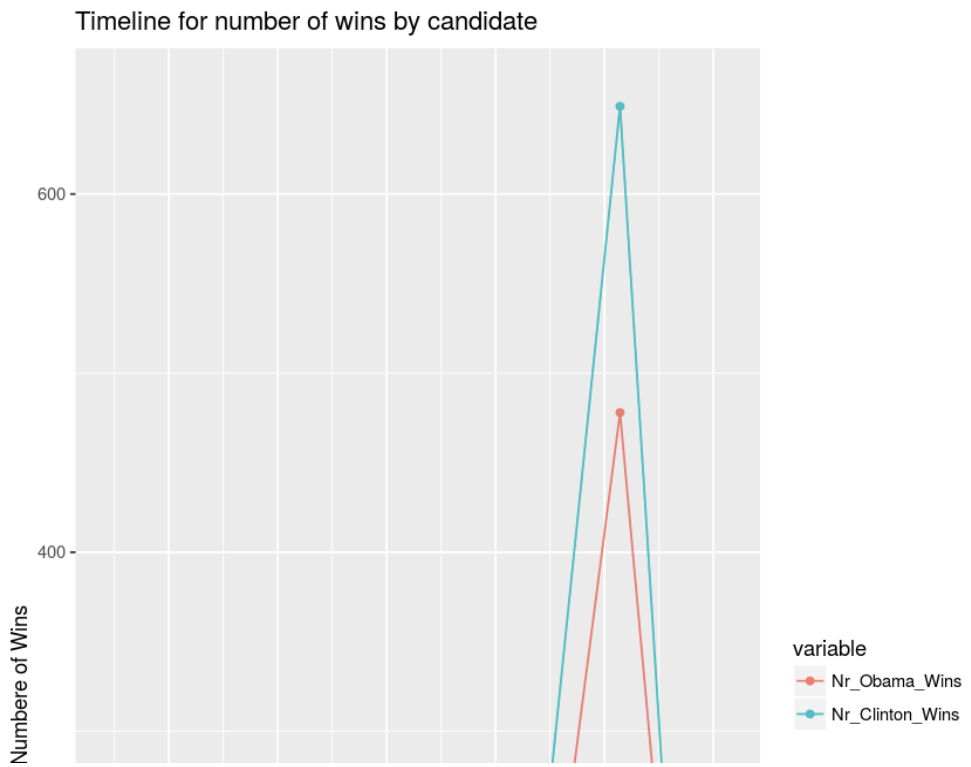
In [154]:

```
d3m <- reshape2::melt(d3, id.var='ElectionDate')
d3m
```

ElectionDate	variable	value
2008-01-03	Nr_Obama_Wins	51
2008-01-08	Nr_Obama_Wins	5
2008-01-19	Nr_Obama_Wins	11
2008-01-26	Nr_Obama_Wins	44
2008-01-29	Nr_Obama_Wins	9
2008-02-05	Nr_Obama_Wins	478
2008-02-09	Nr_Obama_Wins	97
2008-02-10	Nr_Obama_Wins	12
2008-02-12	Nr_Obama_Wins	116
2008-01-03	Nr_Clinton_Wins	48
2008-01-08	Nr_Clinton_Wins	5
2008-01-19	Nr_Clinton_Wins	6
2008-01-26	Nr_Clinton_Wins	2
2008-01-29	Nr_Clinton_Wins	58
2008-02-05	Nr_Clinton_Wins	649
2008-02-09	Nr_Clinton_Wins	99
2008-02-10	Nr_Clinton_Wins	4
2008-02-12	Nr_Clinton_Wins	42

In [155]:

```
#Timelime for each cadidate's number of wins

ggplot(d3m, aes(x=ElectionDate, y=value, col=variable)) + geom_line()+
  ylab('Numbere of Wins')+

  geom_point() +
ggtitle("Timeline for number of wins by candidate")
```



ElectionDate

In [157]:
```
da1<-elect.df.known %>%
   group_by(ElectionDate) %>%
   summarise(AvgBlack = mean(Black))
da1
```

ElectionDate	AvgBlack
2008-01-03	0.8444444
2008-01-08	0.6800000
2008-01-19	2.3235294
2008-01-26	37.1195652
2008-01-29	14.3746269
2008-02-05	9.5952085
2008-02-09	10.8397959
2008-02-10	0.5500000
2008-02-12	19.3911392

In [158]:
```
da2<-elect.df.known %>%
   group_by(ElectionDate) %>%
   summarise(AvgWhite = mean(White))
da2
```

ElectionDate	AvgWhite
2008-01-03	97.67172
2008-01-08	96.96000
2008-01-19	90.67059
2008-01-26	61.21304
2008-01-29	82.71791
2008-02-05	86.20248
2008-02-09	86.08112
2008-02-10	97.33125
2008-02-12	77.72089

In [159]:
```
da3<-elect.df.known %>%
   group_by(ElectionDate) %>%
   summarise(AvgAmericanIndian = mean(AmericanIndian))
da3
```

ElectionDate	AvgAmericanIndian
2008-01-03	0.2898990
2008-01-08	0.2600000
2008-01-19	3.7529412
2008-01-26	0.4086957
2008-01-29	0.6641791
2008-02-05	1.8477374
2008-02-09	1.3459184
2008-02-10	0.7125000
2008-02-12	0.3145570

In [160]:
```
da4<-elect.df.known %>%
   group_by(ElectionDate) %>%
   summarise(AvgHispanic = mean(Hispanic))
da4
```

ElectionDate	AvgHispanic
2008-01-03	2.835354
2008-01-08	1.450000
2008-01-19	14.294118

In [161]:
```
da5<-elect.df.known %>%
   group_by(ElectionDate) %>%
   summarise(AvgAsian = mean(Asian))
da5
```

ElectionDate	AvgAsian
2008-01-03	0.6959596
2008-01-08	1.2200000
2008-01-19	1.7058824
2008-01-26	0.6456522
2008-01-29	1.2373134
2008-02-05	1.2307010
2008-02-09	0.9020408
2008-02-10	0.5875000
2008-02-12	1.6658228

In [162]:
```
da6<-elect.df.known %>%
   group_by(ElectionDate) %>%
   summarise(AvgHawaiian = mean(Hawaiian))
da6
```

ElectionDate	AvgHawaiian
2008-01-03	0.02626263
2008-01-08	0.03000000
2008-01-19	0.19411765
2008-01-26	0.03913043
2008-01-29	0.05671642
2008-02-05	0.06521739
2008-02-09	0.06887755
2008-02-10	0.01875000
2008-02-12	0.03417722

In [163]:
```
#created a table with average of each ethicity for each election date


da<-merge(da1,da2,by="ElectionDate")
da<-merge(da,da3,by="ElectionDate")
da<-merge(da,da4,by="ElectionDate")
da<-merge(da,da5,by="ElectionDate")
da<-merge(da,da6,by="ElectionDate")

da
```

ElectionDate	AvgBlack	AvgWhite	AvgAmericanIndian	AvgHispanic	AvgAsian	AvgHawaiian
2008-01-03	0.8444444	97.67172	0.2898990	2.835354	0.6959596	0.02626263
2008-01-08	0.6800000	96.96000	0.2600000	1.450000	1.2200000	0.03000000
2008-01-19	2.3235294	90.67059	3.7529412	14.294118	1.7058824	0.19411765
2008-01-26	37.1195652	61.21304	0.4086957	2.947826	0.6456522	0.03913043
2008-01-29	14.3746269	82.71791	0.6641791	10.549254	1.2373134	0.05671642
2008-02-05	9.5952085	86.20248	1.8477374	7.510027	1.2307010	0.06521739
2008-02-09	10.8397959	86.08112	1.3459184	4.727041	0.9020408	0.06887755
2008-02-10	0.5500000	97.33125	0.7125000	0.887500	0.5875000	0.01875000
2008-02-12	19.3911392	77.72089	0.3145570	3.298101	1.6658228	0.03417722

2008-02-12	19.3911392	77.72089	0.3145570	3.298101	1.6658228	0.03417722

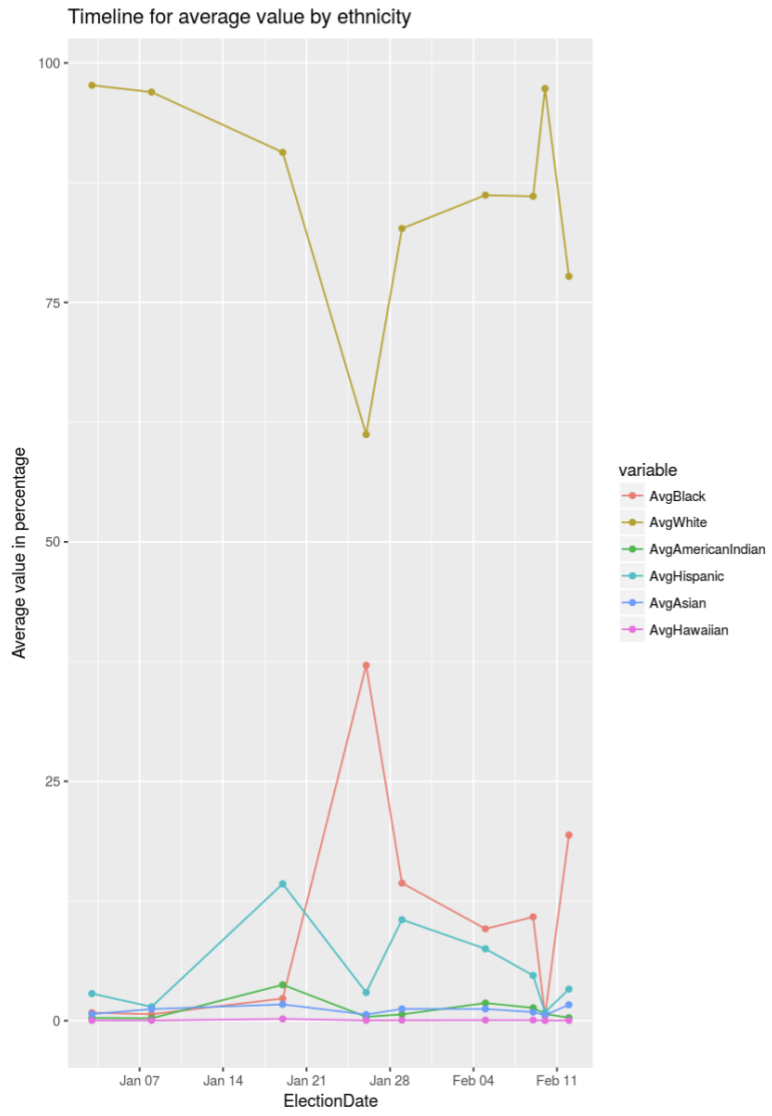
In [164]:

```
dam <- reshape2::melt(da, id.var='ElectionDate')
dam
```

ElectionDate	variable	value
2008-01-03	AvgBlack	0.84444444
2008-01-08	AvgBlack	0.68000000
2008-01-19	AvgBlack	2.32352941
2008-01-26	AvgBlack	37.11956522
2008-01-29	AvgBlack	14.37462687
2008-02-05	AvgBlack	9.59520852
2008-02-09	AvgBlack	10.83979592
2008-02-10	AvgBlack	0.55000000
2008-02-12	AvgBlack	19.39113924
2008-01-03	AvgWhite	97.67171717
2008-01-08	AvgWhite	96.96000000
2008-01-19	AvgWhite	90.67058824
2008-01-26	AvgWhite	61.21304348
2008-01-29	AvgWhite	82.71791045
2008-02-05	AvgWhite	86.20248447
2008-02-09	AvgWhite	86.08112245

2008-02-12	AvgHawaiian	0.03417722

In [165]:
```
#Timeline for average number of ethinicities in the counties that voted in different dates

ggplot(dam, aes(x=ElectionDate, y=value, col=variable)) + geom_line()+
  ylab('Average value in percentage')+

  geom_point()  +
ggtitle("Timeline for average value by ethnicity")
```

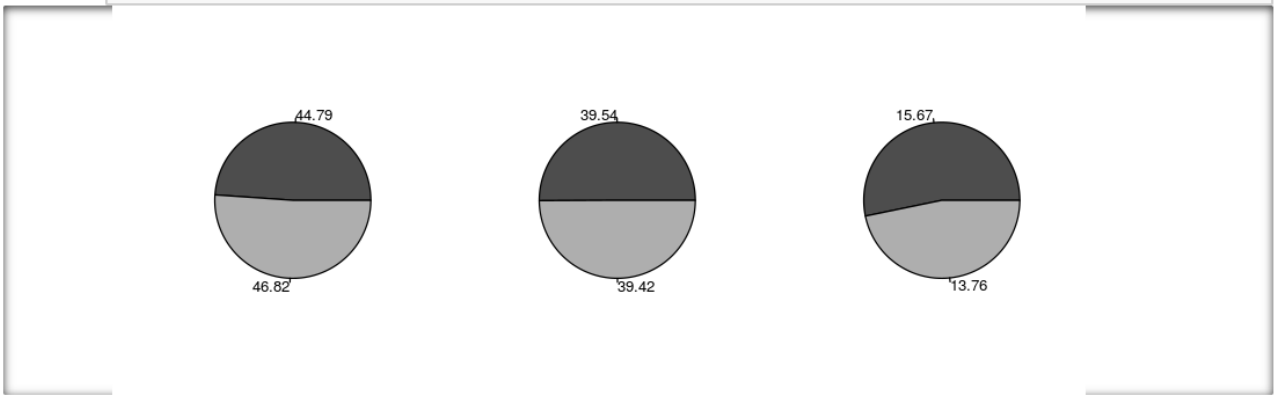


2.2.2.3. Age

In [166]:
```
# Libraries
library(ggplot2)
library(dplyr)
```

In [167]:
```
#created variables for the averages for each candidate and the age groups


a1=as.numeric(format(round(mean(elect.df.known$AgeBelow35[elect.df.known$ObamaWin=="Lose"]),2)))
b1=as.numeric(format(round(mean(elect.df.known$AgeBelow35[elect.df.known$ObamaWin=="Win"]),2)))

a2=as.numeric(format(round(mean(elect.df.known$Age35to65[elect.df.known$ObamaWin=="Lose"]),2)))
b2=as.numeric(format(round(mean(elect.df.known$Age35to65[elect.df.known$ObamaWin=="Win"]),2)))


a3=as.numeric(format(round(mean(elect.df.known$Age65andAbove[elect.df.known$ObamaWin=="Lose"]),2)))
b3=as.numeric(format(round(mean(elect.df.known$Age65andAbove[elect.df.known$ObamaWin=="Win"]),2)))
```

In [168]:
```
#  Pie Chart - winner by age groups
par(mfrow=c(1,3) )

library(plotrix)
slices1 <- c(a1,b1)
slices2 <- c(a2,b2)
slices3 <- c(a3,b3)


lbls <- c("Clinton","Obama")



pie(slices1,labels=slices1 ,
   xlab="Age Below 35",col=grey.colors(3))
pie(slices2,labels=slices2 ,
   xlab="Age 35 to 65",col=grey.colors(3))
pie(slices3,labels=slices3 ,
   xlab="Age 65 and Above",col=grey.colors(3))

mtext(side=3, text="Winner by age group")


plot.new()
    legend("left",legend=lbls, fill=grey.colors(3), box.lty=0, title="Winner")
```



Aggregate table

In [169]:
```
#winner by age group and region

roundmean <- function(x) round(mean(x),0)

(ag <- aggregate(cbind(AgeBelow35,Age35to65,Age65andAbove) ~ ObamaWin + Region,
          data=elect.df,
          FUN=roundmean))
```

ObamaWin	Region	AgeBelow35	Age35to65	Age65andAbove
Lose	Midwest	43	40	18
Win	Midwest	45	39	16
Lose	Northeast	45	41	14
Win	Northeast	43	43	14
Lose	South	45	40	15
Win	South	48	39	13
Lose	West	48	38	14
Win	West	47	40	13

Win	West	47	40	13

3.2. Split the known data into training/testing datasets

In [170]:
```
# 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 75% of the row indices in the known dataset
rowIndicesTrain <- sample(1:nKnown,
                          size = round(nKnown*0.75),
                          replace = FALSE)
```

In [171]:
```
# 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, ]
```

Section 4 - generate and test prediction models

4.1. Three prediction models

4.1.1.Linear regression model

In [172]:
```
#build the model

lm1 <- lm(ObamaWinNr ~Black+White+Bachelors+SocialSecurityRate+MedicareRate
          +Age65andAbove+ HighSchool+IncomeAbove75K+MedianIncome+AgeBelow35
          ,data = elect.df.training)
summary(lm1)
```

```
Call:
lm(formula = ObamaWinNr ~ Black + White + Bachelors + SocialSecurityRate +
    MedicareRate + Age65andAbove + HighSchool + IncomeAbove75K +
    MedianIncome + AgeBelow35, data = elect.df.training)

Residuals:
     Min       1Q   Median       3Q      Max
-1.18494 -0.34364 -0.05188  0.38025  0.98090

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.864e+00  4.080e-01  -4.569 5.36e-06 ***
Black               2.303e-02  2.148e-03  10.723  < 2e-16 ***
White               6.850e-03  2.192e-03   3.125  0.00182 **
Bachelors           1.352e-02  2.909e-03   4.649 3.68e-06 ***
SocialSecurityRate -6.806e-06  7.124e-06  -0.955  0.33959
MedicareRate       -1.371e-07  4.018e-06  -0.034  0.97279
Age65andAbove      -8.789e-03  7.855e-03  -1.119  0.26340
HighSchool          1.618e-02  2.520e-03   6.420 1.91e-10 ***
IncomeAbove75K     -2.915e-02  5.793e-03  -5.032 5.54e-07 ***
```

In [173]:
```
#summarize results
lm1.pred<- predict(lm1, elect.df.test)
```

In [174]:
```
lm1.pred
```

```
1  0.480379533960963
6  0.570385151037576
7  0.641335803020555
11 0.441244315584941
27 0.363624111606389
31 0.559627628773246
35 0.408615482321577
38 0.536368567025936
41 0.564054668698223
43 0.413942536633769
44 0.538457017322972
```

```
64  0.441023651311812
```

In [175]:
```
elect.df$ObamaWinNr
```

```
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
<NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
```

In [176]:
```
#round the predictions to the appropriate class

class_prediction <-
    ifelse(lm1.pred>0.50,
          "1",
          "0")
```

In [177]:
```
.libPaths("/usr/local/lib/R/site-library")
library(caret)
```

In [178]:
```
#see model accuracy

confusionMatrix(class_prediction, elect.df.test$ObamaWinNr)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 174  87
         1  46 127

               Accuracy : 0.6935
                 95% CI : (0.6478, 0.7366)
    No Information Rate : 0.5069
    P-Value [Acc > NIR] : 2.346e-15

                  Kappa : 0.3854
 Mcnemar's Test P-Value : 0.0005235

            Sensitivity : 0.7909
            Specificity : 0.5935
         Pos Pred Value : 0.6667
         Neg Pred Value : 0.7341
             Prevalence : 0.5069
         Detection Rate : 0.4009
   Detection Prevalence : 0.6014
      Balanced Accuracy : 0.6922

       'Positive' Class : 0
```

In [179]:
```
#create results table
model.results <- data.frame(Accuracy=0.69,  Model="linear regression")
```

4.1.2. Backward stepwise with most correlated variables

In [180]:
```
# stepwise
lm.step.backward <- step(lm1, direction = "backward")
```

```
Start:  AIC=-2282.12
ObamaWinNr ~ Black + White + Bachelors + SocialSecurityRate + 
    MedicareRate + Age65andAbove + HighSchool + IncomeAbove75K + 
    MedianIncome + AgeBelow35

                      Df Sum of Sq    RSS     AIC
- MedicareRate         1    0.0002 221.84 -2284.1
- AgeBelow35           1    0.0003 221.84 -2284.1
- SocialSecurityRate   1    0.1568 222.00 -2283.2
- Age65andAbove        1    0.2151 222.06 -2282.9
<none>                             221.84 -2282.1
- White                1    1.6776 223.52 -2274.3
- MedianIncome         1    3.1448 224.99 -2265.8
- Bachelors            1    3.7139 225.56 -2262.5
- IncomeAbove75K       1    4.3511 226.20 -2258.8
- HighSchool           1    7.0829 228.93 -2243.2
- Black                1   19.7584 241.60 -2173.0

Step:  AIC=-2284.12
ObamaWinNr ~ Black + White + Bachelors + SocialSecurityRate + 
    Age65andAbove + HighSchool + IncomeAbove75K + MedianIncome + 
    AgeBelow35

                      Df Sum of Sq    RSS     AIC
- AgeBelow35           1    0.0004 221.85 -2286.1
- SocialSecurityRate   1    0.1915 222.04 -2285.0
- Age65andAbove        1    0.2475 222.09 -2284.7
<none>                             221.84 -2284.1
- White                1    1.6778 223.52 -2276.3
- MedianIncome         1    3.1542 225.00 -2267.7
- Bachelors            1    3.8283 225.67 -2263.8
- IncomeAbove75K       1    4.3591 226.20 -2260.8
- HighSchool           1    7.1451 228.99 -2244.8
- Black                1   19.7642 241.61 -2175.0

Step:  AIC=-2286.12
ObamaWinNr ~ Black + White + Bachelors + SocialSecurityRate + 
    Age65andAbove + HighSchool + IncomeAbove75K + MedianIncome

                      Df Sum of Sq    RSS     AIC
- SocialSecurityRate   1    0.2085 222.05 -2286.9
- Age65andAbove        1    0.2771 222.12 -2286.5
<none>                             221.85 -2286.1
- White                1    1.6874 223.53 -2278.2
- MedianIncome         1    3.2885 225.13 -2269.0
- Bachelors            1    3.8309 225.68 -2265.8
- IncomeAbove75K       1    4.3606 226.21 -2262.8
- HighSchool           1    7.1585 229.00 -2246.8
- Black                1   19.8786 241.72 -2176.4

Step:  AIC=-2286.89
ObamaWinNr ~ Black + White + Bachelors + Age65andAbove + HighSchool + 
    IncomeAbove75K + MedianIncome

                  Df Sum of Sq    RSS     AIC
<none>                        222.05 -2286.9
- White            1    1.5439 223.60 -2279.9
- Age65andAbove    1    3.3334 225.39 -2269.5
- MedianIncome     1    3.5686 225.62 -2268.1
- Bachelors        1    4.0496 226.10 -2265.4
- IncomeAbove75K   1    4.4720 226.53 -2262.9
- HighSchool       1    7.9607 230.01 -2243.0
- Black            1   19.6710 241.72 -2178.4
```

In [181]:
```
summary(lm.step.backward)
```

```

Call:
lm(formula = ObamaWinNr ~ Black + White + Bachelors + Age65andAbove + 
    HighSchool + IncomeAbove75K + MedianIncome, data = elect.df.training)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.17344 -0.34302 -0.05882  0.38173  1.00453 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -1.947e+00  2.353e-01  -8.276 3.14e-16 ***
Black          2.278e-02  2.128e-03  10.707  < 2e-16 ***
```

```
Residual standard error: 0.4142 on 1294 degrees of freedom
Multiple R-squared:  0.315,     Adjusted R-squared:  0.3112
F-statistic: 84.99 on 7 and 1294 DF,  p-value: < 2.2e-16
```

In [182]:
```
#summarize results
lm.step.backward.pred<- predict(lm.step.backward, elect.df.test)
lm.step.backward.pred
```

1	0.45915101818686
6	0.566826391544139
7	0.640563630203552
11	0.429771420033102
27	0.354209666838106
31	0.558431695259838
35	0.40370380263824
38	0.538182904488333
41	0.55646051317139
43	0.407766977496232
44	0.538461545705053
46	0.482549328614769
49	0.365737240478119
50	0.546691505257223
51	0.61578590608276
64	0.442581577254267
66	0.428637219171519

In [183]:
```
#round results to the appropriate class

class_prediction <-
    ifelse(lm.step.backward.pred>0.50,
           "1",
           "0")
```

In [184]:
```
#see model accuracy

confusionMatrix(class_prediction, elect.df.test$ObamaWinNr)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 175  87
         1  45 127

               Accuracy : 0.6959
                 95% CI : (0.6502, 0.7388)
    No Information Rate : 0.5069
    P-Value [Acc > NIR] : 1.053e-15

                  Kappa : 0.39
 Mcnemar's Test P-Value : 0.0003589

            Sensitivity : 0.7955
            Specificity : 0.5935
         Pos Pred Value : 0.6679
         Neg Pred Value : 0.7384
             Prevalence : 0.5069
         Detection Rate : 0.4032
   Detection Prevalence : 0.6037
      Balanced Accuracy : 0.6945

       'Positive' Class : 0
```

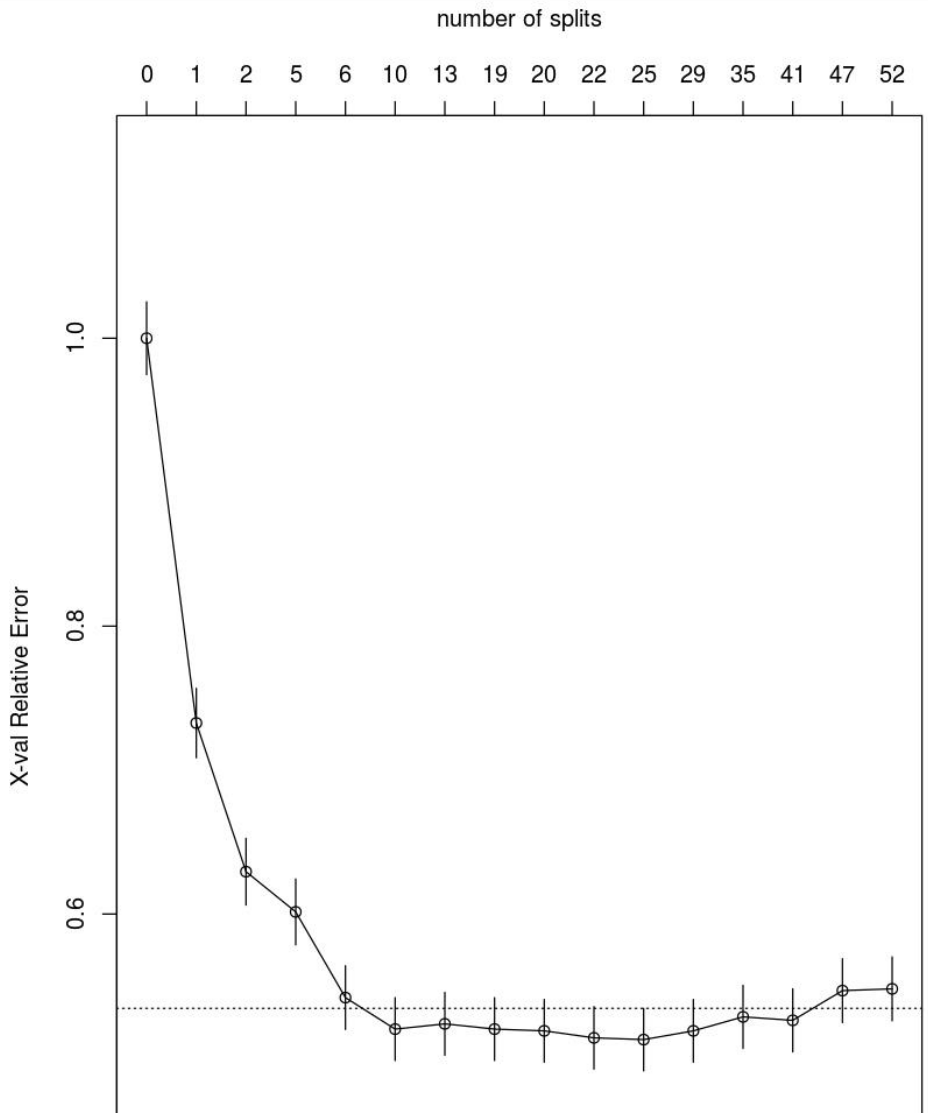
In [185]:
```
# Save the results for later ...

model.results <- rbind(model.results, data.frame(Accuracy=0.69,  Model="Stepwise with most correlated variables"))
```

4.1.3. Classification tree

In [186]:
```
library(rpart)
library(rpart.plot)    # install.packages("rpart.plot") if needed
```

In [187]:
```
set.seed(999)  # optional: as rpart randomly applies cross-validation setting seed ensures repeatability of outcomes

rt.all <- rpart(ObamaWin ~
          Black+HighSchool+
           MedianIncome+
        MedicareRate+
            AgeBelow35,



         data = elect.df.known,  # as rpart uses cross-validation can use entire known dataset rather than training dat
         xval=5,  # this is the number of cross-validations
         cp = 0.001)
```

In [188]:
```
plotcp(rt.all,upper = "splits")
```



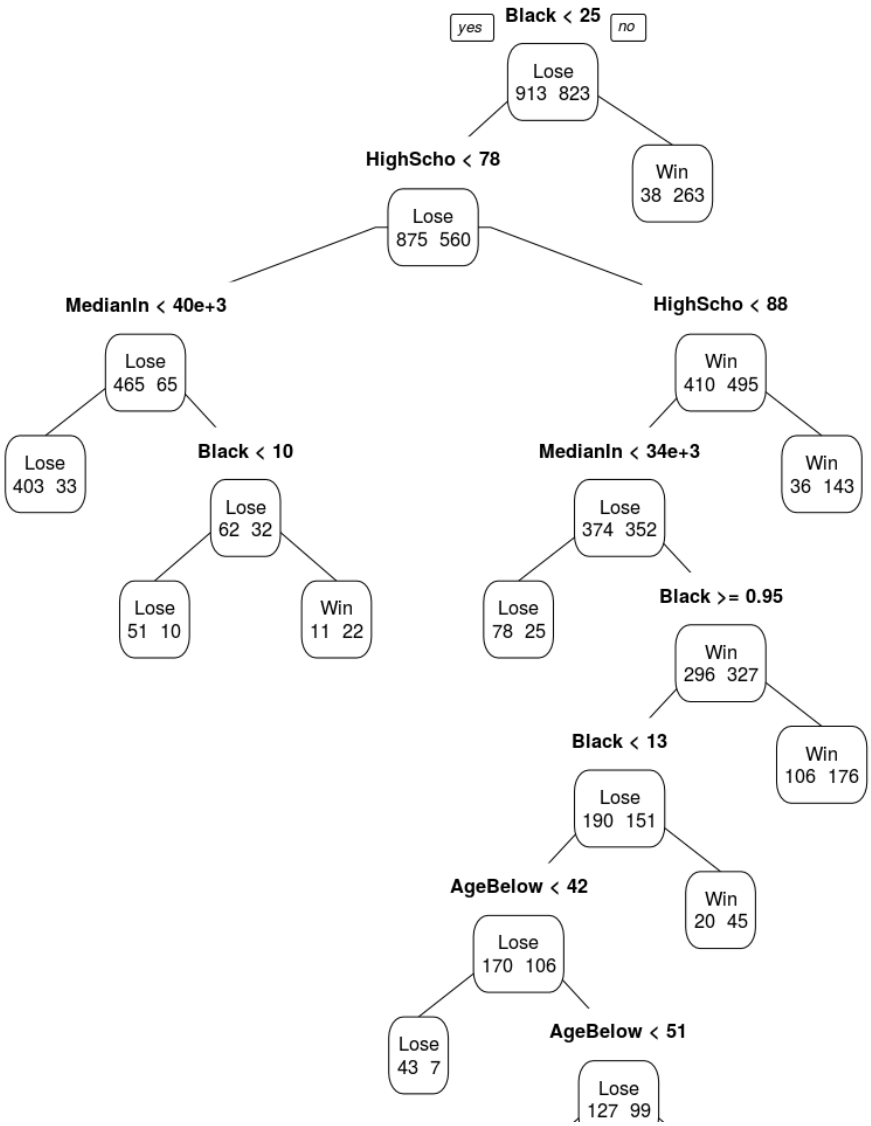
In [189]:
```
# This function determines the optimal cp corresponding to the tree with the smallest number of splits that has
# a xerror value less than the tree with the best (minimum) xerror value plus its standard error (xstd)

optimalCP <- function(rt.model){
    df<-as.data.frame(rt.model$cptable)
    minerr <- min(df[,"xerror"])
    minerr.xstd <- df[df$xerror==minerr,"xstd"]
    df[df$xerror<minerr+minerr.xstd,][1,"CP"]}

optimalCP(rt.all)
```

0.00607533414337789

In [190]:
```
# We can now use this function to "prune" the rpart model back to an optimal number of splits

rt.all.opt <- prune(rt.all, cp=optimalCP(rt.all))
```

In [191]:
```
options(scipen = 999)
```

In [192]:
```
prp(rt.all.opt, type = 1, extra = 1)
```



In [193]:
```
rt.all.opt.pred <- predict(rt.all.opt, elect.df.test)
```

In [194]:
```
rt.all.opt.pred
```

	Lose	Win
1	0.2011173	0.7988827
6	0.2011173	0.7988827
7	0.6010363	0.3989637
11	0.6010363	0.3989637
27	0.7572816	0.2427184
31	0.6010363	0.3989637
35	0.3758865	0.6241135
38	0.3758865	0.6241135
41	0.3758865	0.6241135
43	0.3758865	0.6241135
44	0.6010363	0.3989637
46	0.3758865	0.6241135
49	0.3758865	0.6241135
50	0.6010363	0.3989637
51	0.2011173	0.7988827
64	0.6010363	0.3989637
66	0.3758865	0.6241135
72	0.3758865	0.6241135
74	0.3758865	0.6241135
76	0.3758865	0.6241135
77	0.2011173	0.7988827
80	0.7572816	0.2427184
83	0.3758865	0.6241135
85	0.2011173	0.7988827
88	0.3758865	0.6241135
91	0.2011173	0.7988827
93	0.7572816	0.2427184
94	0.6010363	0.3989637
95	0.3758865	0.6241135
106	0.2011173	0.7988827
⋮	⋮	⋮
1591	0.2011173	0.79888268
1601	0.3076923	0.69230769
1613	0.8600000	0.14000000
1617	0.9243119	0.07568807

1737	0.3333333	0.66666667

In [195]:
```
#assign the predictions to the right class

class_prediction <-
  ifelse(rt.all.opt.pred[,1] > 0.50,
         "Lose",
         "Win"
  )
```

In [196]:
```
#see model accuracy

confusionMatrix( class_prediction, elect.df.test$ObamaWin)
```

```
Confusion Matrix and Statistics

          Reference
Prediction Lose Win
      Lose  167  46
      Win    53 168

               Accuracy : 0.7719
                 95% CI : (0.7295, 0.8106)
    No Information Rate : 0.5069
    P-Value [Acc > NIR] : <0.0000000000000002

                  Kappa : 0.5439
 Mcnemar's Test P-Value : 0.5465

            Sensitivity : 0.7591
            Specificity : 0.7850
         Pos Pred Value : 0.7840
         Neg Pred Value : 0.7602
             Prevalence : 0.5069
         Detection Rate : 0.3848
   Detection Prevalence : 0.4908
      Balanced Accuracy : 0.7721

       'Positive' Class : Lose
```

In [197]:
```
#save the results
model.results <- rbind(model.results, data.frame(Accuracy=0.77,  Model="Classification tree"))
```

In [198]:
```
model.results
```

Accuracy	Model
0.69	linear regression
0.69	Stepwise with most correlated variables
0.77	Classification tree

4.2. Use best model to predict on unkown dataset

In [199]:
```
unknown.pred <- predict(rt.all.opt, elect.df.unknown)
```

In [200]:
```
unknown.pred
```

	Lose	Win
1738	0.3758865	0.62411348
1739	0.6010363	0.39896373
1741	0.3758865	0.62411348
1742	0.9243119	0.07568807
1743	0.7572816	0.24271845
1744	0.3758865	0.62411348
1745	0.3758865	0.62411348
1746	0.6010363	0.39896373
1747	0.3758865	0.62411348
1748	0.3758865	0.62411348
1749	0.3758865	0.62411348
1750	0.3758865	0.62411348
1751	0.9243119	0.07568807
1752	0.6010363	0.39896373
1753	0.8600000	0.14000000
1754	0.2011173	0.79888268
1755	0.6010363	0.39896373

In [201]:
```
#assign the prediction probabilities to the right class (class with higher probability)

class_prediction <-
  ifelse(unknown.pred[,1] > 0.50,
         "Lose",
         "Win"
  )
```

In [202]:
```
#predict vote winner and save as csv

predictions.df <- elect.df.unknown

predictions.df$PredictedObamaWin <-class_prediction

write.csv(predictions.df, file="PredictedObamaWin.csv")
```

In [203]:
```
predictions.df
```

	County	State	Region	FIPS	ElectionDate	ElectionType	TotalVote	Clinton	Obama	MalesPer100Females	⋯	DisabilitiesRate	Hor
1738	Hawaii	HI	West	15001	2008-02-19	Caucuses	NA	NA	NA	100.0	⋯	2269	64.5
1739	Honolulu	HI	West	15003	2008-02-19	Caucuses	NA	NA	NA	99.1	⋯	1825	54.6
1741	Maui	HI	West	15009	2008-02-19	Caucuses	NA	NA	NA	100.9	⋯	1122	57.6
1742	Adams	WI	Midwest	55001	2008-02-19	Primary	NA	NA	NA	116.2	⋯	2113	85.3
1743	Ashland	WI	Midwest	55003	2008-02-19	Primary	NA	NA	NA	97.5	⋯	2243	70.7
1744	Barron	WI	Midwest	55005	2008-02-19	Primary	NA	NA	NA	97.8	⋯	1791	75.8
1745	Bayfield	WI	Midwest	55007	2008-02-19	Primary	NA	NA	NA	103.0	⋯	1697	82.6
1746	Brown	WI	Midwest	55009	2008-02-19	Primary	NA	NA	NA	98.8	⋯	1432	65.4
1747	Buffalo	WI	Midwest	55011	2008-02-19	Primary	NA	NA	NA	101.4	⋯	1460	76.5
1748	Burnett	WI	Midwest	55013	2008-02-19	Primary	NA	NA	NA	100.2	⋯	1585	84.5
1749	Calumet	WI	Midwest	55015	2008-02-19	Primary	NA	NA	NA	100.4	⋯	802	80.4
1750	Chippewa	WI	Midwest	55017	2008-02-19	Primary	NA	NA	NA	103.8	⋯	1650	75.7
1751	Clark	WI	Midwest	55019	2008-02-19	Primary	NA	NA	NA	101.1	⋯	1513	81.2
1752	Columbia	WI	Midwest	55021	2008-02-19	Primary	NA	NA	NA	101.9	⋯	865	74.8

Section 5 - visuals for conclusion

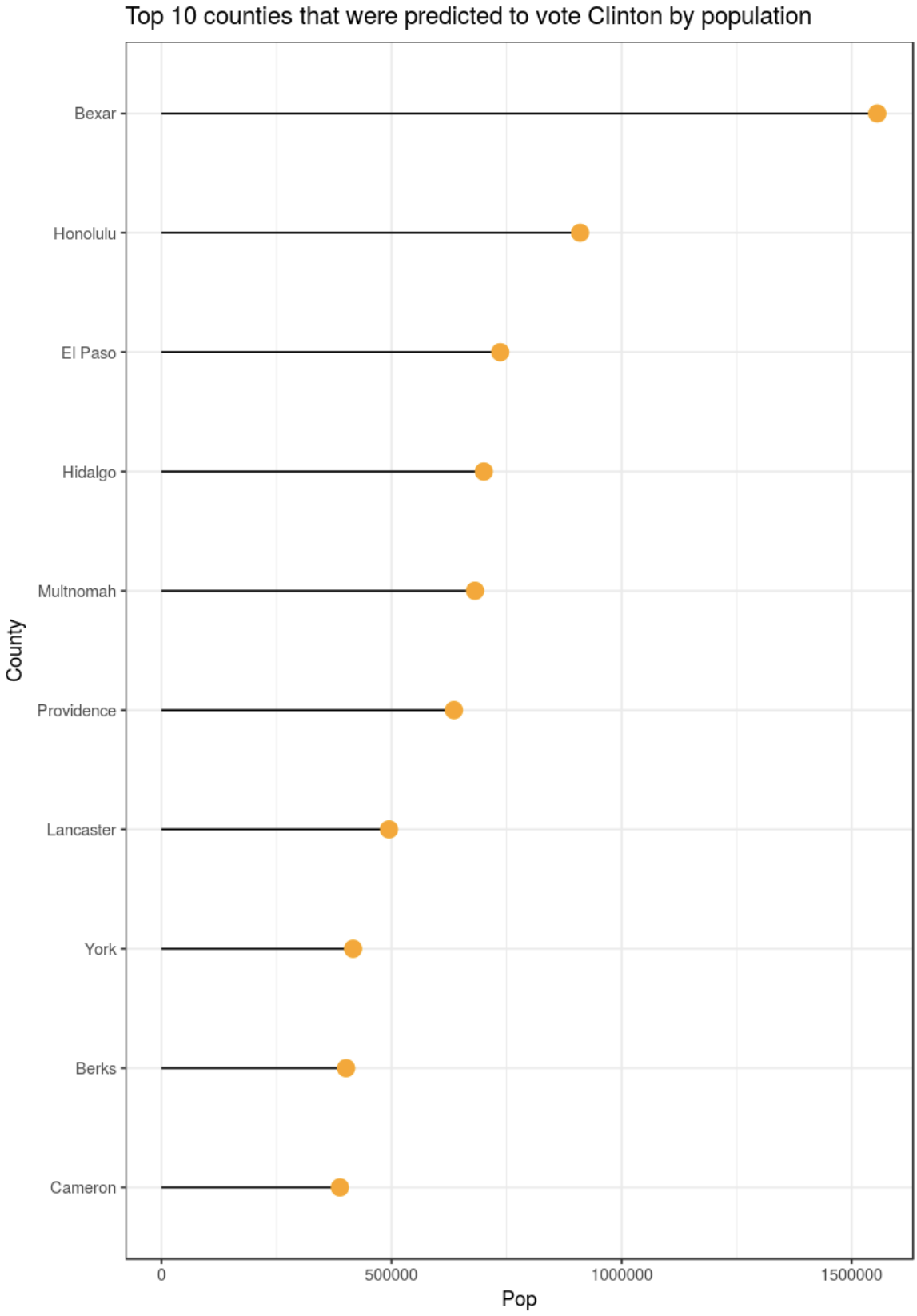
Top 10 counties by population in the counties that did not vote and were predicted Cliton.

In [204]:
```
.libPaths("/usr/local/lib/R/site-library")

library(dplyr)
```

In [205]:
```
#dataset with top 10 counties by population in which Cliton was predicted to win

data<-predictions.df[predictions.df$PredictedObamaWin=="Lose",]

data <- data[with(data,order(-Pop)),]
data <- data[1:10,]


data
```

	County	State	Region	FIPS	ElectionDate	ElectionType	TotalVote	Clinton	Obama	MalesPer100Females	⋯	DisabilitiesRate	Hc
1920	Bexar	TX	South	48029	2008-03-04	Primary	NA	NA	NA	95.3	⋯	2823	61
1739	Honolulu	HI	West	15003	2008-02-19	Caucuses	NA	NA	NA	99.1	⋯	1825	54
1976	El Paso	TX	South	48141	2008-03-04	Primary	NA	NA	NA	92.5	⋯	3424	63
2012	Hidalgo	TX	South	48215	2008-03-04	Primary	NA	NA	NA	95.1	⋯	4651	73
2736	Multnomah	OR	West	41051	2008-05-20	Primary	NA	NA	NA	98.6	⋯	2320	56
1905	Providence	RI	Northeast	44007	2008-03-04	Primary	NA	NA	NA	93.1	⋯	3708	53
2312	Lancaster	PA	Northeast	42071	2008-04-22	Primary	NA	NA	NA	95.7	⋯	1557	70
2343	York	PA	Northeast	42133	2008-04-22	Primary	NA	NA	NA	97.2	⋯	1531	76
2282	Berks	PA	Northeast	42011	2008-04-22	Primary	NA	NA	NA	96.6	⋯	1864	74
1936	Cameron	TX	South	48061	2008-03-04	Primary	NA	NA	NA	92.6	⋯	4750	67

In [206]:
```


# Barplot
ggplot(data, aes(x=reorder(County,Pop), y=Pop)) +
   geom_segment( aes(xend=County, yend=0)) +
geom_point( size=4, color="orange")+
xlab("County")+
coord_flip() +
theme_bw()+
ggtitle("Top 10 counties that were predicted to vote Clinton by population")
```

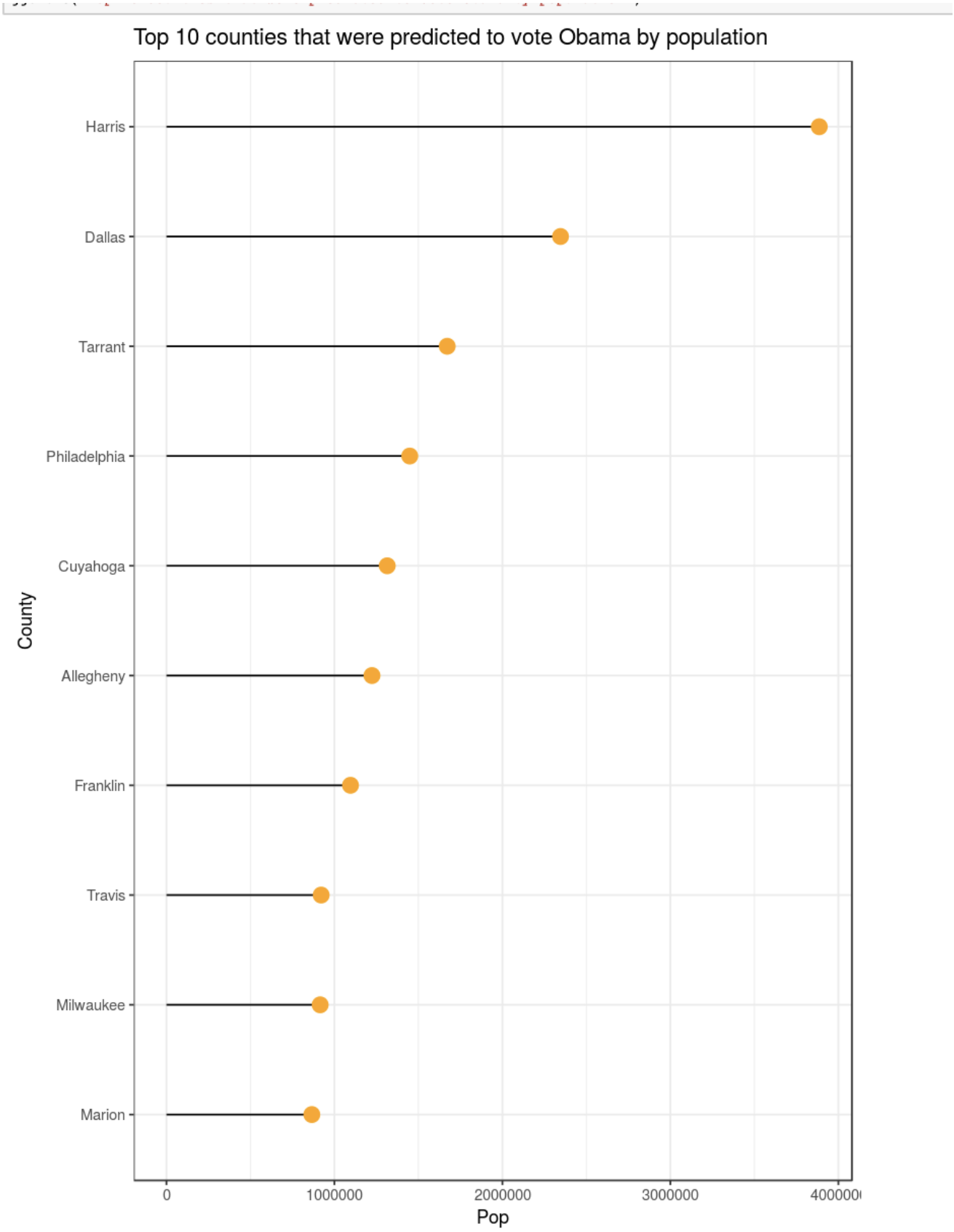

Top 10 counties that were predicted to vote Clinton by population
Bexar
Honolulu
El Paso
Hidalgo
Multnomah
Providence
Lancaster
York
Berks
Cameron
County
0
500000
1000000
1500000
Pop

Pop

In [207]:
```
#dataset with top 10 counties by population in which Obama was predicted to win

data<-predictions.df[predictions.df$PredictedObamaWin=="Win",]

data <- data[with(data,order(-Pop)),]
data <- data[1:10,]


data
```

	County	State	Region	FIPS	ElectionDate	ElectionType	TotalVote	Clinton	Obama	MalesPer100Females	⋯	DisabilitiesRate	H
2005	Harris	TX	South	48201	2008-03-04	Primary	NA	NA	NA	100.0	⋯	2078	5
1962	Dallas	TX	South	48113	2008-03-04	Primary	NA	NA	NA	101.8	⋯	1823	5
2123	Tarrant	TX	South	48439	2008-03-04	Primary	NA	NA	NA	99.3	⋯	1381	6
2327	Philadelphia	PA	Northeast	42101	2008-04-22	Primary	NA	NA	NA	87.1	⋯	6461	5
1831	Cuyahoga	OH	Midwest	39035	2008-03-04	Primary	NA	NA	NA	90.0	⋯	3104	6
2278	Allegheny	PA	Northeast	42003	2008-04-22	Primary	NA	NA	NA	90.7	⋯	2556	6
1838	Franklin	OH	Midwest	39049	2008-03-04	Primary	NA	NA	NA	95.9	⋯	2235	5
2130	Travis	TX	South	48453	2008-03-04	Primary	NA	NA	NA	106.4	⋯	1375	5
1782	Milwaukee	WI	Midwest	55079	2008-02-19	Primary	NA	NA	NA	92.8	⋯	3615	5
2392	Marion	IN	Midwest	18097	2008-05-06	Primary	NA	NA	NA	94.4	⋯	2101	5

In [208]:
```
# Barplot
ggplot(data, aes(x=reorder(County,Pop), y=Pop)) +
   geom_segment( aes(xend=County, yend=0)) +
geom_point( size=4, color="orange")+
xlab("County")+
coord_flip() +
theme_bw()+
ggtitle("Top 10 counties that were predicted to vote Obama by population")
```


Top 10 counties that were predicted to vote Obama by population
Harris
Dallas
Tarrant
Philadelphia
Cuyahoga
Allegheny
Franklin
Travis
Milwaukee
Marion
County
0
1000000
2000000
3000000
Pop

In [3]:
```
#NICU
#Load libraries
.libPaths("/usr/local/lib/R/site-library")
library(ggplot2)
library(caret)
library("forecast")
library(tidyr)
library(lubridate)
library(repr)
```

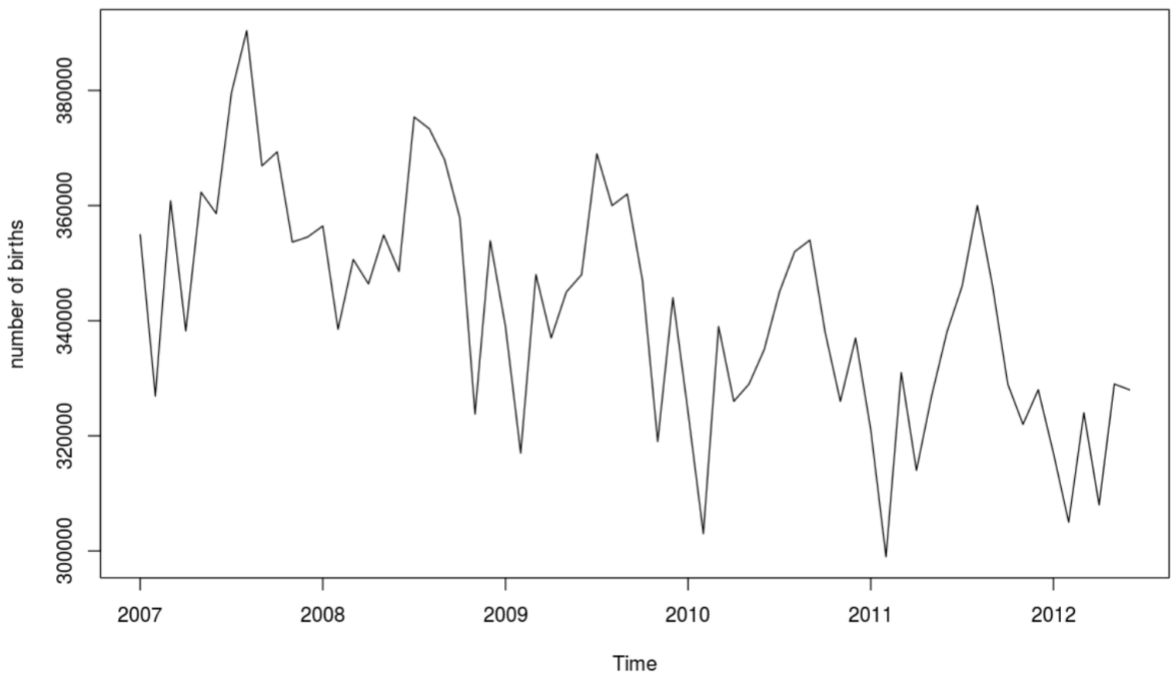
```
Warning message:
“replacing previous import ‘stats::sigma’ by ‘robustbase::sigma’ when loading ‘ddalpha’”
Attaching package: ‘lubridate’

The following object is masked from ‘package:base’:

    date
```

In [4]:
```
#Read US births data and create Year, Month, Day columns
baby.df <- read.csv("US_Births.csv")
baby.df$Date <- as.Date(paste(as.character(baby.df$Yr_Mo), "1", sep=""), format = "%Y%m%d")
baby.df_2 <- separate(baby.df, Date, c("Year", "Month", "Day"))
baby.df_2
```

Yr_Mo	Live.Births	Year	Month	Day
200701	354943	2007	01	01
200702	326891	2007	02	01
200703	360828	2007	03	01
200704	338224	2007	04	01
200705	362319	2007	05	01
200706	358606	2007	06	01
200707	379616	2007	07	01
200708	390378	2007	08	01
200709	366904	2007	09	01
200710	369324	2007	10	01

In [5]:
```
#Storing Births Data in a Time Series Object
Live.Births.ts <- ts(baby.df_2$Live.Births,
                     start = c(2007, 01),
                     end = c(2012, 06),
                     freq = 12)
```

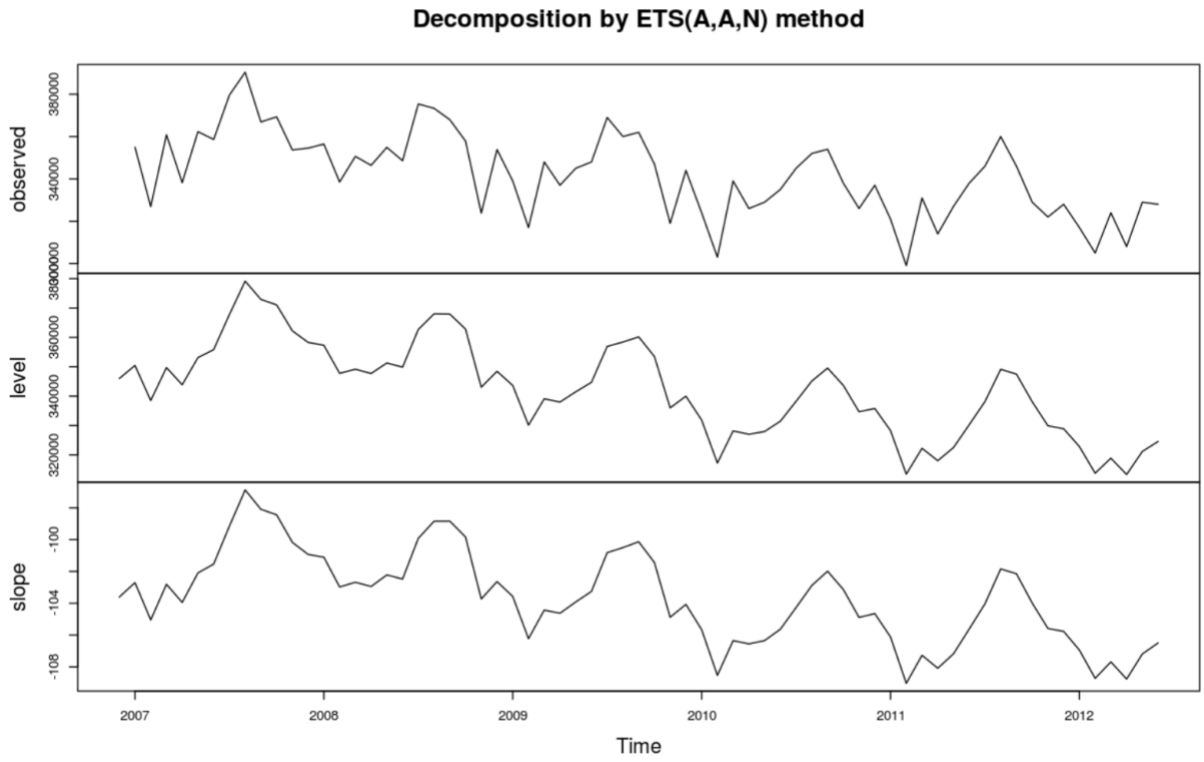
In [6]:
```
#Adjusting and plotting Number of Births
options(repr.plot.width=10, repr.plot.height=6.5)

plot(Live.Births.ts, ylab = "number of births")
```



In [7]:
```
#AAN model + RMSE
(Live.Births.ets.AAN <- ets(Live.Births.ts, model = "AAN"))

rmse.ets <- function (etsmodel) cat("RMSE = ",
                                    sqrt(etsmodel$mse))

rmse.ets(Live.Births.ets.AAN)
```

```
ETS(A,A,N)

Call:
 ets(y = Live.Births.ts, model = "AAN")

  Smoothing parameters:
    alpha = 0.505
    beta  = 1e-04

  Initial states:
    l = 345984.1013
    b = -103.612

  sigma:  15584.28

     AIC     AICc      BIC
1560.848 1561.848 1571.796

RMSE =  15584.28
```

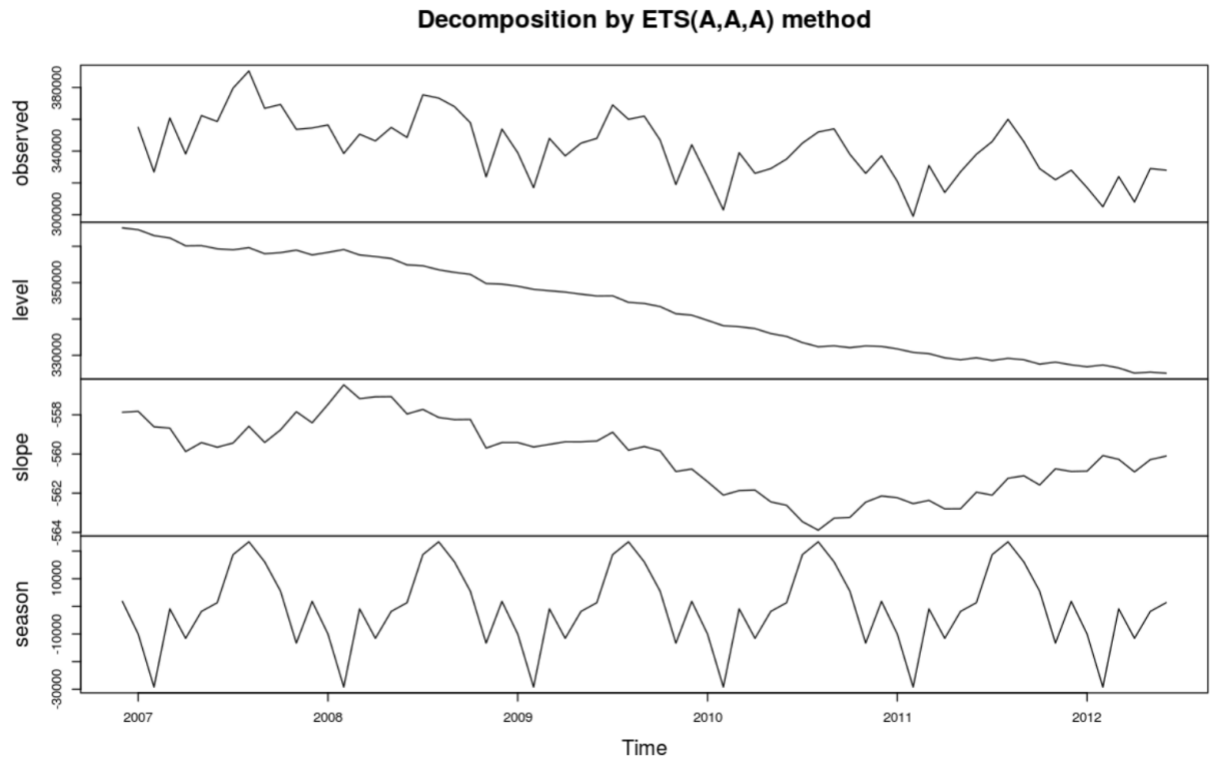
In [8]:
```
#Plotting AAN
plot(Live.Births.ets.AAN)
```



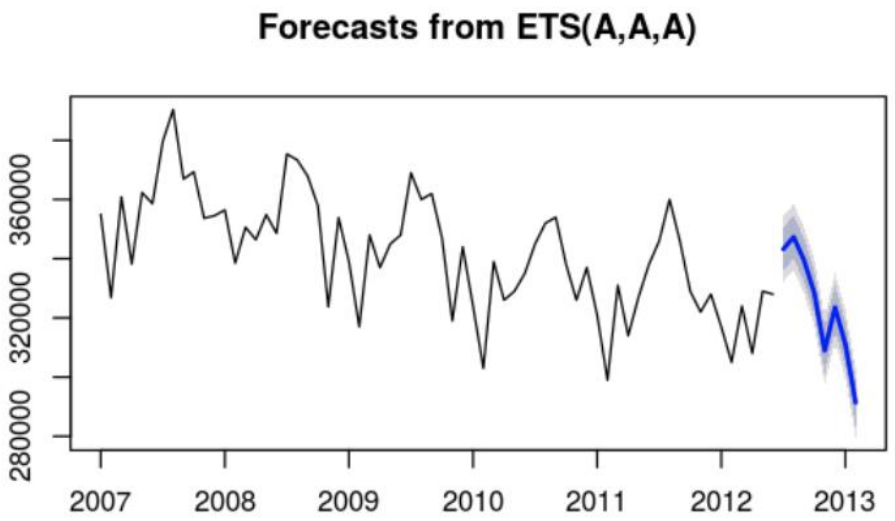
In [9]:
```
#AAA model + RMSE
(Live.Births.ets.AAA <- ets(Live.Births.ts, model = "AAA"))

rmse.ets <- function (etsmodel) cat("RMSE = ",
                                    sqrt(etsmodel$mse))

rmse.ets(Live.Births.ets.AAA)
```

```
ETS(A,A,A)

Call:
 ets(y = Live.Births.ts, model = "AAA")

  Smoothing parameters:
    alpha = 0.1342
    beta  = 1e-04
    gamma = 1e-04

  Initial states:
    l = 365057.0079
    b = -557.8703
    s=1815.176 -13310.1 5510.562 16085.76 23364.83 18724.14
           1333.278 -1820.319 -11570.28 -899.8827 -29211.26 -10021.9

  sigma:  5622.293

     AIC     AICc      BIC
1450.271 1463.021 1487.495

RMSE =  5622.293
```

In [10]:
```
#Plotting AAA (Seasonality)
plot(Live.Births.ets.AAA)
```



In [11]:
```
#Forecasting AAA Model to Feb 2013
Live.Births.ets.AAA.pred <- forecast(Live.Births.ets.AAA, h = 8)


#Plot the forecast for the AAA model
par(mfrow = c(2, 2))
plot(Live.Births.ets.AAA.pred)
```



In [12]:
```
#Number of US Births in Feb 2013 with 80% conf level
forecast <- forecast(Live.Births.ets.AAA, h = 8, level = c(80, 95))

cat('February 2013: mean births = ',round(forecast$mean[8],1),"\n")
cat('           upper 80% confid. births = ',round(forecast$upper[8,1],1),"\n")
```

```
February 2013: mean births =  291394.2
           upper 80% confid. births =  299042.4
```

In [12]:
```
#Number of US Births in Feb 2013 with 80% conf level
forecast <- forecast(Live.Births.ets.AAA, h = 8, level = c(80, 95))

cat('February 2013: mean births = ',round(forecast$mean[8],1),"\n")
cat('           upper 80% confid. births = ',round(forecast$upper[8,1],1),"\n")
```

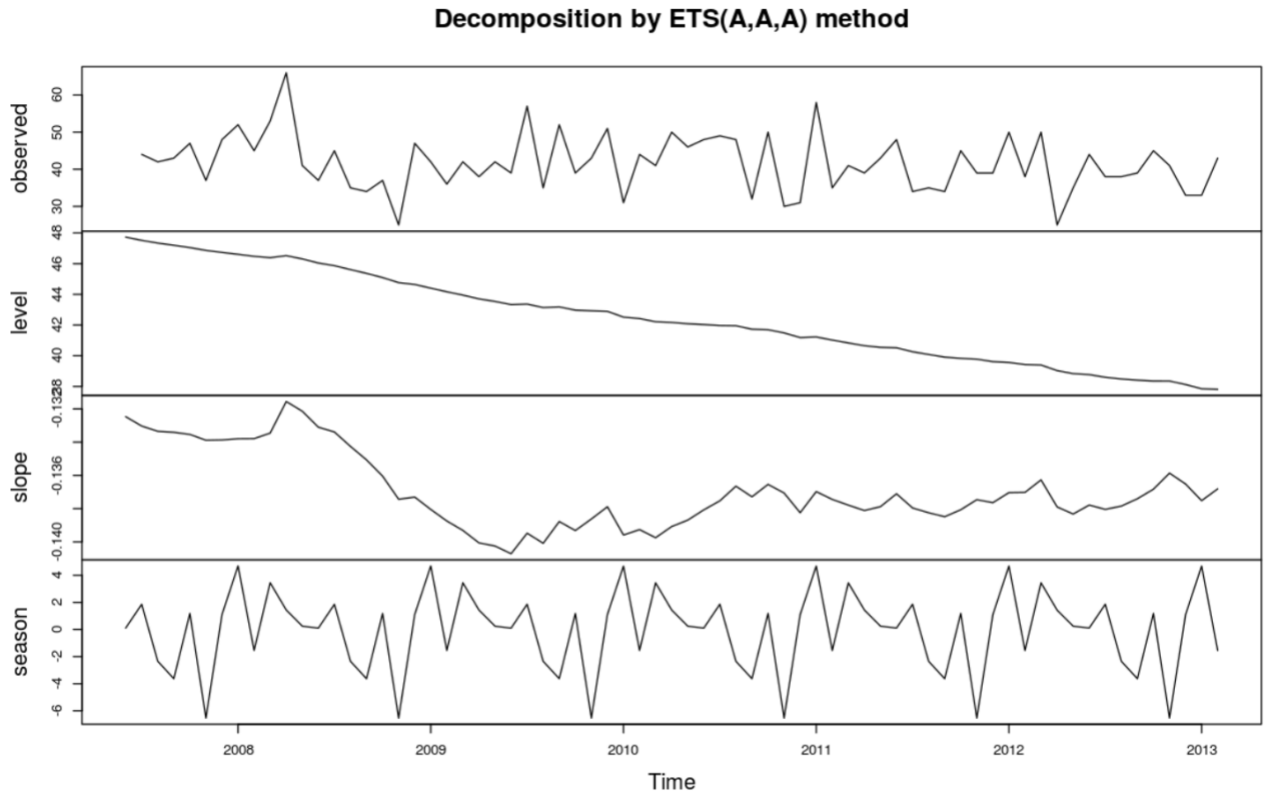
```
February 2013: mean births =  291394.2
           upper 80% confid. births =  299042.4
```

In [13]:
```
#Load the NICU dataset
NICU.df <- read.csv("NICU.csv")
```

In [12]:
```
#Store the admits data in a Time Series Object
admits.ts <- ts(NICU.df$Admits,
                start = c(2007, 7),
                end = c(2013, 2),
                freq = 12)
```

In [13]:
```
#Plotting AAA Model Admits
(admits.ets.AAA <- ets(admits.ts, model = "AAA"))
plot(admits.ets.AAA)
```

```
ETS(A,A,A)

Call:
 ets(y = admits.ts, model = "AAA")

  Smoothing parameters:
    alpha = 0.0145
    beta  = 1e-04
    gamma = 5e-04

  Initial states:
    l = 47.7283
    b = -0.1325
    s=0.1023 0.2482 1.4273 3.4425 -1.5452 4.6862
           1.098 -6.5329 1.1865 -3.6319 -2.3422 1.8612

  sigma:  7.0941

     AIC     AICc      BIC
587.3863 599.6263 625.1179
```

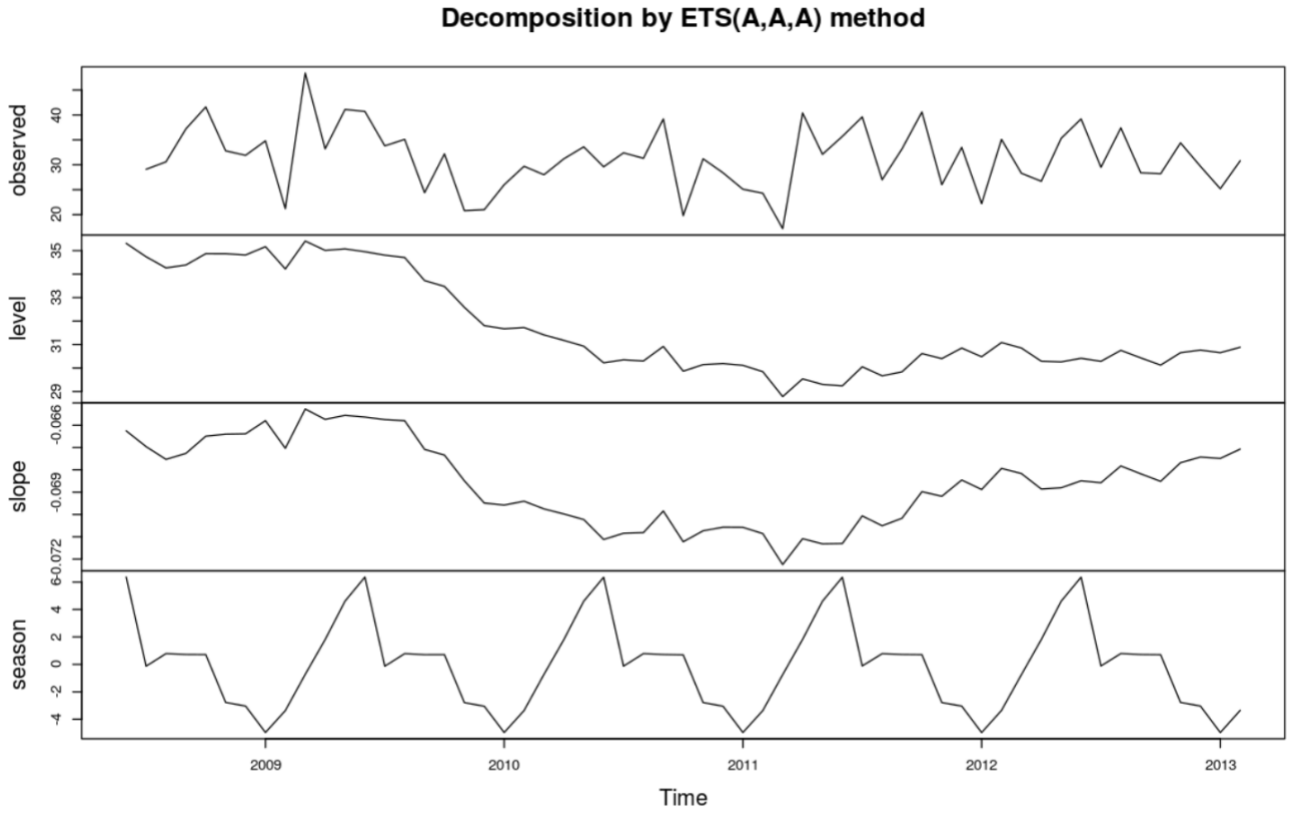


In [14]:
```
#Store the ALOS data in a Time Series Object
alos.ts <- ts(NICU.df$ALOS,
              start = c(2007, 7),
              end = c(2013, 2),
              freq = 12)
```

In [15]:
```
#Plotting AAA Model ALOS
(alos.ets.AAA <- ets(alos.ts, model = "AAA"))
plot(alos.ets.AAA)
```

```
Warning message in ets(alos.ts, model = "AAA"):
“Missing values encountered. Using longest contiguous portion of time series”
```

```
ETS(A,A,A)

Call:
 ets(y = alos.ts, model = "AAA")

  Smoothing parameters:
    alpha = 0.0837
    beta  = 1e-04
    gamma = 0.0015

  Initial states:
    l = 35.3086
    b = -0.0663
    s=6.3646 4.605 1.8114 -0.7489 -3.3493 -4.9687
           -3.0369 -2.7734 0.7035 0.7197 0.7928 -0.1198

  sigma:  5.7794

     AIC     AICc      BIC
455.9014 472.0066 490.3323
```

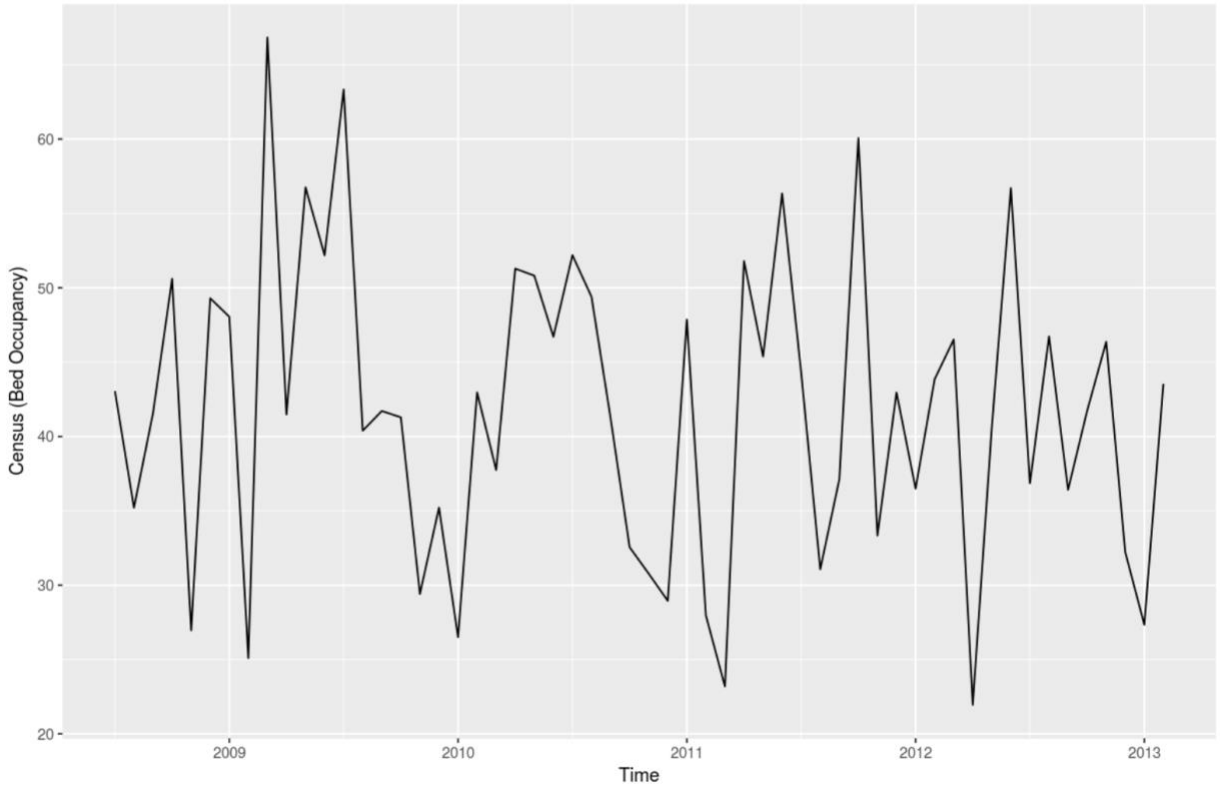

Decomposition by ETS(A,A,A) method
observed
level
slope
season
Time
2009
2010
2011
2012
2013

In [12]:
```
#Number of US Births in Feb 2013 with 80% conf level
forecast <- forecast(Live.Births.ets.AAA, h = 8, level = c(80, 95))

cat('February 2013: mean births = ',round(forecast$mean[8],1),"\n")
cat('           upper 80% confid. births = ',round(forecast$upper[8,1],1),"\n")
```

```
February 2013: mean births =  291394.2
           upper 80% confid. births =  299042.4
```

In [14]:
```
NICU.df$Census <- with(NICU.df, Admits*ALOS*12/365)
```

In [16]:
```
census.ts <- ts(NICU.df$Census[-(1:12)],     # remove first 12 months that are missing ALOS data
                start = c(2008,7),
                end = c(2013,2),
                freq = 12)

autoplot(census.ts, ylab = "Census (Bed Occupancy)")
```



In [17]:
```
census.ets.ann <- ets(census.ts, model = "ANN")
rmse.ets(census.ets.ann)
```

```
RMSE =  10.21951
```

In [18]:
```
census.ets.aan <- ets(census.ts, model = "AAN")
rmse.ets(census.ets.aan)
```

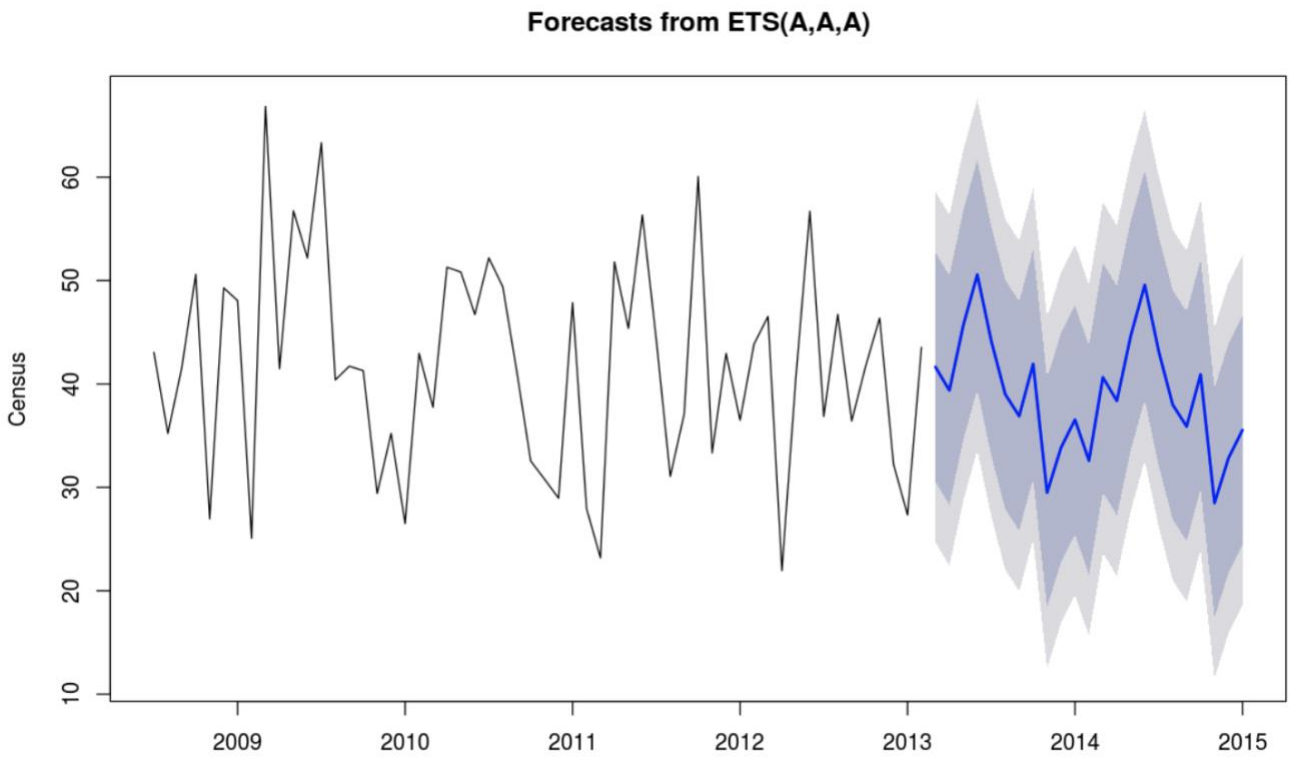
```
RMSE =  10.12035
```

In [20]:
```
census.ets.aaa <- ets(census.ts, model = "AAA")
rmse.ets(census.ets.aaa)
```

```
RMSE =  8.64606
```

In [21]:
```
forecast <- forecast(census.ets.aaa, h = 23, level = c(80, 95))

cat('December 2014: mean beds = ',round(forecast$mean[23],1),"\n")
cat('           upper 80% confid. beds = ',round(forecast$upper[23,1],1),"\n")

plot(forecast, ylab = "Census")
```

```
December 2014: mean beds =  35.5
           upper 80% confid. beds =  46.6
```



Appendix

2.2.2.1. Income

In [123]:
```
#Aggregate table 1 - Mean IncomeAbove75K by ObamaWin
aggregate(IncomeAbove75K ~ ObamaWin,
                    data=elect.df,
                    FUN=mean)
```

ObamaWin	IncomeAbove75K
Lose	12.82738
Win	16.50535

2.2.2.3. Age

In [129]:
```
#Create a "Winner" attribute with either Obama or Clinton as possible values
elect.df$Winner <- ifelse(elect.df$Obama>elect.df$Clinton,
                                "Obama",
                                "Clinton")
```

In [128]:
```
#Aggregate table 2 - Mean AgeBelow35, Age35to65 and Age65andAbove where Obama and Clinton win

aggregate(cbind(AgeBelow35, Age35to65,Age65andAbove) ~ Winner,
                    data=elect.df,
                    FUN=mean)
```

Winner	AgeBelow35	Age35to65	Age65andAbove
Clinton	44.78817	39.54294	15.67415
Obama	46.82309	39.42418	13.75529

Appendix 2.3.

Correlations between all attributes

In [209]:
```
library (ellipse); library (RColorBrewer); options(repr.plot.height=10)
my_colors=colorRampPalette(brewer.pal(5, "Spectral"))(100)
data=cor(elect.df[,c(10:41)], use="complete.obs")

plotcorr(data, col=my_colors[data*50+50], mar=c(0,0,0,0),
        cex.lab=0.8, type="upper", diag=FALSE)
```

MalesPer100Females
AgeBelow35
Age35to65
Age65andAbove
White
Black
Asian
AmericanIndian
Hawaiian
Hispanic
HighSchool
Bachelors
Poverty
IncomeAbove75K
MedianIncome
AverageIncome
UnemployRate
ManfEmploy
SpeakingNonEnglish
Medicare
MedicareRate
SocialSecurity
SocialSecurityRate
RetiredWorkers
Disabilities
DisabilitiesRate
Homeowner
SameHouse1995and2000
Pop
PopDensity
LandArea
FarmArea

References

(1) “Election Center 2008: Results,” CNNPolitics.com, http://www.cnn.com/ELECTION/2008/primaries/results/scorecard/#D (accessed 2021 Feb 3).

(2) Rosentiel T. Inside Obama’s Sweeping Victory [Internet]. Pew Research Center. Pew Research Center; 2008 [Accessed 2021 Feb 3]. Available from: https://www.pewresearch.org/2008/11/05/inside-obamas-sweeping-victory/

(3) Khun D. Exit polls: How Obama won [Internet]. POLITICO. 2008 [Accessed 2020 Feb 5]. Available from: https://www.politico.com/story/2008/11/exit-polls-how-obama-won-015297

(4) Whiter A. MSIN0025 Week 4: Joining Datasets Using Dates [Internet]. login.echo360.org.uk. [Accessed 2021 Feb 12]. Available from: https://echo360.org.uk/lesson/dcd0689f-a08b-4839-a357-d89b6bb469bd/classroom

(5) Hazelwood-Smith S. Sperm quality rises in winter - BioNews [Internet]. www.bionews.org.uk. 2013 [Accessed 2021 Feb 7]. Available from: https://www.bionews.org.uk/page_94037

(6) Livingston G, Cohn D. U.S. Birth Rate Falls to a Record Low; Decline Is Greatest Among Immigrants [Internet]. Pew Research Center’s Social & Demographic Trends Project. 2012 [Accessed 2021 Feb 10]. Available from: https://www.pewresearch.org/social-trends/2012/11/29/u-s-birth-rate-falls-to-a-record-low-decline-is-greatest-among-immigrants/

(7) Howe N. U.S. Birthrate Falls -- Again [Internet]. Forbes. 2015 [cited 2021 Feb 11]. Available from: https://www.forbes.com/sites/neilhowe/2015/01/28/u-s-birthrate-falls-again/?sh=71efbbb83bfe