Chatzimoschou Angeliki

bapt 1534

**STATISTICS FOR BUSINESS ANALYTICS II, PROJECT 1 Professor: Karlis Dimitrios Assistant: Rui Miguel Forte**

Logistic Regression US Elections dataset

Chatzimoschou Angeliki (BAPT 1534)

18/04/2016

The dataset contains the results from the presidential elections in 1992 for 3141 counties in US with demographic characteristics of the counties. The specific data extract doesn’t contain the voting results of 27 counties, thus we removed the respective missing values.   
 Our goal is to create a model predicting the win of the Republican candidate over the Democrat candidate and define which variables contribute to this win. This is a binary classification project ( 1 0: whether the Republican won or lost the county respectively), that’s why we will build a logistic regression classifier.

install.packages("psych") install.packages("Amelia") install.packages("ggplot2") install.packages("maps") install.packages("corrplot") install.packages("caret") install.packages("ResourceSelection") install.packages("pROC")

At first, we replace the abbreviation of each state’s name with its whole name   
(ie. ‘AL’ -> ‘ alabama ’ using Excel), so as to create afterwards a US map showing the Republican voting frequency of each state separately. Afterwards, we load the ‘counties’ dataset and draw the first 6 rows, in order to observe the values and have a first look at the type of our variables.

counties <- read.csv2("C:/Users/Angelica/Desktop/project I/counties.xls - Copy.csv")  
head(counties)

county state pop.density pop pop.change age6574 age75 crime college  
## 1 Autauga alabama 61 34222 11.9 5.7 4.1 4996 14.5  
## 2 Baldwin alabama 67 98280 35.4 9.2 6.0 3329 16.8  
## 3 Barbour alabama 29 25417 2.0 8.2 6.4 3192 11.8  
## 4 Bibb alabama 28 16576 9.2 6.7 6.0 0 4.7  
## 5 Blount alabama 62 39248 10.6 7.4 5.6 2052 7.0  
## 6 Bullock alabama 18 11042 3.7 8.5 7.6 3630 10.0  
   
 income farm democrat republican Perot white black turnout  
## 1 32240 1.8 30.9 55.9 12.3 79.31740 20.001753 45.54088  
## 2 30199 1.7 26.2 56.5 16.5 86.04498 12.861213 47.28938  
## 3 23838 2.4 46.4 42.9 9.8 55.54550 44.041389 41.03946  
## 4 23714 0.9 43.2 46.5 10.2 78.74035 20.982143 40.54054  
## 5 26323 4.7 32.9 53.8 11.8 97.83173 1.327456 42.05310  
## 6 17796 2.6 67.7 26.0 5.5 27.49502 72.323853 43.61529

*By observing the dataset, we have decided to recode the 49 US states in 5 greater areas (Southeast, Northeast, Southwest, West and Midwest) depending the geographical position of each state and set it as factor, because we don’t want R to use it as a numerical and affect the dataset distribution.*  
  
#create regions by grouping states  
for (i in 1: nrow(counties)) {  
 if (counties$state [i] == "louisiana" | counties$state [i] == "arkansas" | counties$state[i]=="mississippi" | counties$state [i] == "tennessee" | counties$state [i] == "alabama" | counties$state [i] == "kentucky" | counties$state [i] == "georgia" | counties$state [i] == "florida" | counties$state [i] == "south carolina" | counties$state [i] == "north carolina" | counties$state[i]=="virginia" | counties$state [i] == "west virginia" | counties$state[i]=="district of columbia" ){  
 counties$area [i] <- 1 #Southeast  
 } else if (counties$state [i] == "maryland" | counties$state [i] =="delaware" | counties$state [i] =="new jersey" | counties$state [i] =="connecticut" | counties$state [i] =="pennsylvania" | counties$state [i] =="new york" | counties$state [i] =="massachusetts" | counties$state [i] =="new hampshire" | counties$state [i] =="vermont" | counties$state [i] =="maine" | counties$state [i] =="rhode island"){  
 counties$area [i] <- 2 #Northeast  
 } else if ( counties$state [i] =="new mexico" | counties$state [i] =="oklahoma" | counties$state [i] =="texas" ){  
 counties$area [i] <- 3 #Southwest  
 } else if (counties$state [i] == "california" | counties$state [i] =="utah" | counties$state [i] =="alaska" | counties$state [i] =="colorado" | counties$state [i] =="wyoming" | counties$state [i] =="idaho" | counties$state [i] =="montana" | counties$state [i] =="washington" | counties$state [i] =="oregon" | counties$state [i] =="hawaii" ){  
 counties$area [i] <- 4 #West  
 } else {  
 counties$area [i] <- 5 #Midwest  
 }  
}  
counties$area<-as.factor(counties$area)  
View(counties)

After that, we count the NA observations, then we spot the variables: democrat, republican and Perot, containing NAs with the ‘missmap’ function of the library (Amelia) and remove the NAs using the function ‘na.omit’. Then we print some descriptive statistics by using the function ‘str’ and ‘describe’ from the library (psych). We extract information concerning the number of the observations, the type, mean, median, standard deviation, minimum/maximum values, skewness, kyrtosis and standard error of each variable.   
  
Furthermore, we use the libraries ‘ggplot2’ and ‘maps’ in order to generate a Republican Voting Frequency map, filled by the variable ‘republican’, in order to obtain a picture about the voting frequency of each US state concerning the Republican candidate.  
In order to achieve this, we load the dataset ‘states’ and we merge it with the ‘counties’ dataset (by the newly created variable ‘region’ which is identical to the ‘state’ variable) into the new dataset named ‘Total’.  
   
As we observe the map, we realize that the frequency ratio of the republican is rather lowin the majority of the US states (most of the state are of red color) and this is also confirmed by the actual outcome of this year’s elections (democrats’ won).   
  
#remove rows with NAs  
sum(is.na(counties))

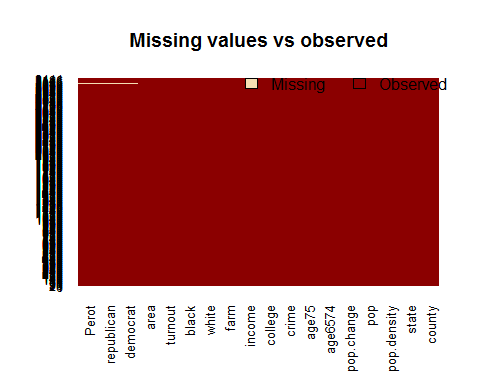
## [1] 81

library(Amelia)

## Loading required package: Rcpp

missmap(counties, main = "Missing values vs observed")  
  
counties<-na.omit(counties)  
  
sum(is.na(counties))

## [1] 0



#descriptives  
  
str(counties)

## 'data.frame': 3141 obs. of 18 variables:  
## $ county : Factor w/ 1895 levels "Abbeville","Acadia",..: 83 90 101 153 168 229 239 250 300 323 ...  
## $ state : Factor w/ 49 levels "alabama","alaska",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ pop.density: int 61 67 29 28 62 18 28 191 62 36 ...  
## $ pop : int 34222 98280 25417 16576 39248 11042 21892 116034 36876 19543 ...  
## $ pop.change : num 11.9 35.4 2 9.2 10.6 ...  
## $ age6574 : num 5.7 9.2 8.2 6.7 7.4 ...  
## $ age75 : num 4.1 6 6.4 6 5.6 ...  
## $ crime : int 4996 3329 3192 0 2052 3630 3530 4174 3173 3223 ...  
## $ college : num 14.5 16.8 11.8 4.7 7 ...  
## $ income : int 32240 30199 23838 23714 26323 17796 21499 28340 26331 24932 ...  
## $ farm : num 1.8 1.7 2.4 0.9 4.7 ...  
## $ democrat : num 30.9 26.2 46.4 43.2 32.9 ...  
## $ republican : num 55.9 56.5 42.9 46.5 53.8 ...  
## $ Perot : num 12.3 16.5 9.8 10.2 11.8 ...  
## $ white : num 79.3 86 55.5 78.7 97.8 ...  
## $ black : num 20 12.86 44.04 20.98 1.33 ...  
## $ turnout : num 45.5 47.3 41 40.5 42.1 ...  
## $ area : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...

library(psych)  
describe(counties)

## vars n mean sd median trimmed mad  
## county\* 1 3141 949.97 530.99 948.00 947.60 659.76  
## state\* 2 3141 25.91 13.76 25.00 26.18 17.79  
## pop.density 3 3141 222.89 1428.46 39.00 60.39 44.48  
## pop 4 3141 79181.75 263812.66 22085.00 33769.84 22181.18  
## pop.change 5 3141 6.50 20.63 2.70 3.82 14.23  
## age6574 6 3141 8.29 2.22 8.20 8.22 1.93  
## age75 7 3141 6.58 2.40 6.30 6.42 2.08  
## crime 8 3141 3007.76 2326.23 2629.00 2759.28 2088.98  
## college 9 3141 13.51 6.58 11.80 12.49 4.45  
## income 10 3141 28476.08 7164.25 27361.00 27814.36 5794.00  
## farm 11 3141 6.44 7.34 3.90 5.06 4.45  
## democrat 12 3114 39.73 10.78 39.00 39.34 9.79  
## republican 13 3114 39.79 8.59 39.20 39.58 8.60  
## Perot 14 3114 19.81 6.88 20.30 19.85 7.71  
## white 15 3141 87.11 15.88 94.14 90.24 7.49  
## black 16 3141 8.59 14.34 1.50 5.13 2.17  
## turnout 17 3141 43.71 8.60 44.09 44.00 7.48  
## area\* 18 3141 3.02 1.71 3.00 3.03 2.97  
## min max range skew kurtosis se  
## county\* 1.00 1895.00 1894.00 0.04 -1.12 9.47  
## state\* 1.00 49.00 48.00 -0.07 -1.20 0.25  
## pop.density 0.00 52432.00 52432.00 23.55 713.18 25.49  
## pop 52.00 8863164.00 8863112.00 16.55 448.14 4707.19  
## pop.change -34.40 207.70 242.10 2.44 12.05 0.37  
## age6574 0.60 21.10 20.50 0.59 2.43 0.04  
## age75 0.00 15.90 15.90 0.64 0.63 0.04  
## crime 0.00 20179.00 20179.00 1.18 2.29 41.51  
## college 0.00 53.40 53.40 1.88 4.87 0.12  
## income 10903.00 65201.00 54298.00 1.20 2.73 127.83  
## farm 0.00 67.60 67.60 2.23 7.50 0.13  
## democrat 6.80 84.60 77.80 0.43 0.60 0.19  
## republican 9.10 75.00 65.90 0.23 0.18 0.15  
## Perot 3.20 46.90 43.70 -0.04 -0.67 0.12  
## white 5.04 100.00 94.96 -1.78 3.22 0.28  
## black 0.00 86.24 86.24 2.17 4.51 0.26  
## turnout 0.00 101.93 101.93 -0.89 5.80 0.15  
## area\* 1.00 5.00 4.00 -0.04 -1.70 0.03

# US MAP elections: republican frequency by state  
library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

library(maps)

##   
## # maps v3.1: updated 'world': all lakes moved to separate new #  
## # 'lakes' database. Type '?world' or 'news(package="maps")'. #

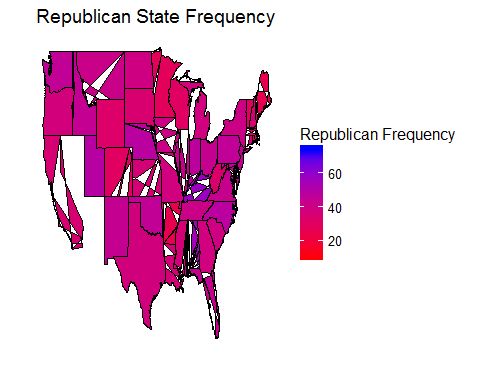
all\_states<-map\_data("state")  
head(all\_states)

## long lat group order region subregion  
## 1 -87.46201 30.38968 1 1 alabama <NA>  
## 2 -87.48493 30.37249 1 2 alabama <NA>  
## 3 -87.52503 30.37249 1 3 alabama <NA>  
## 4 -87.53076 30.33239 1 4 alabama <NA>  
## 5 -87.57087 30.32665 1 5 alabama <NA>  
## 6 -87.58806 30.32665 1 6 alabama <NA>

counties$region <- counties$state  
Total <- merge(all\_states, counties, by="region")  
head(Total)

## region long lat group order subregion county state  
## 1 alabama -87.94329 30.52719 1 30 <NA> Autauga alabama  
## 2 alabama -87.94329 30.52719 1 30 <NA> Baldwin alabama  
## 3 alabama -87.94329 30.52719 1 30 <NA> Barbour alabama  
## 4 alabama -87.94329 30.52719 1 30 <NA> Bibb alabama  
## 5 alabama -87.94329 30.52719 1 30 <NA> Blount alabama  
## 6 alabama -87.94329 30.52719 1 30 <NA> Cherokee alabama  
## pop.density pop pop.change age6574 age75 crime college income farm  
## 1 61 34222 11.9 5.7 4.1 4996 14.5 32240 1.8  
## 2 67 98280 35.4 9.2 6.0 3329 16.8 30199 1.7  
## 3 29 25417 2.0 8.2 6.4 3192 11.8 23838 2.4  
## 4 28 16576 9.2 6.7 6.0 0 4.7 23714 0.9  
## 5 62 39248 10.6 7.4 5.6 2052 7.0 26323 4.7  
## 6 36 19543 5.9 8.9 5.8 3223 6.7 24932 3.9  
## democrat republican Perot white black turnout area  
## 1 30.9 55.9 12.3 79.31740 20.001753 45.54088 1  
## 2 26.2 56.5 16.5 86.04498 12.861213 47.28938 1  
## 3 46.4 42.9 9.8 55.54550 44.041389 41.03946 1  
## 4 43.2 46.5 10.2 78.74035 20.982143 40.54054 1  
## 5 32.9 53.8 11.8 97.83173 1.327456 42.05310 1  
## 6 53.6 34.9 10.7 92.89259 6.605946 40.30088 1

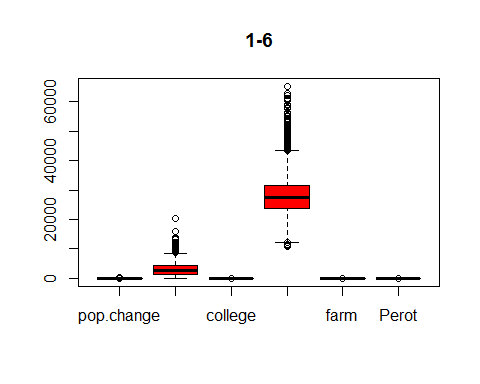
g <- ggplot()  
g <- g + geom\_polygon(data=Total, aes(x=long, y=lat, group = group, fill=Total$republican),colour="black") + scale\_fill\_continuous(low = "red", high = "blue", guide="colorbar")  
g1 <- g + theme\_bw() + labs(fill = "Republican Frequency",title = "Republican State Frequency", x="", y="")  
g1 + scale\_y\_continuous(breaks=c()) + scale\_x\_continuous(breaks=c()) + theme(panel.border = element\_blank())

  
*\*\* The white blanks in the ‘states’ map is an R bug. \*\**

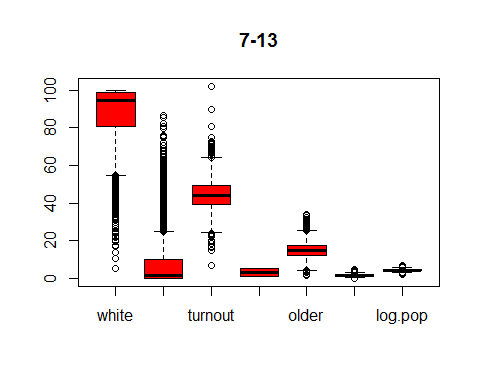
As a next step, we remove the 3 categorical variables that do not serve our predicting goal: ‘county’ (because all its values are unique and thus contain no statistical information), ‘state’ and its identical ‘region’ (because we have replaced it with the grouping ‘area’).   
  
In addition, we observe that we have 2 variables describing 2 close age groups: age6574 and age75+, thus we decide to recode them into one variable named ‘older’ by adding them up. Furthermore, the variables ‘pop’ and ‘pop density’ have really high skewness and kyrtosis values. In order to get them more normalized along with the rest of the variables we take the log (base=10) of the initial variables and create the ‘log. pop’ and ‘log. pop. density’ respectively.

Then we create our response variable ‘REP’ (factor)by recoding the variables ‘republican’   
and ‘democrat’. If the votes for the republican are more than the votes for the democrat   
then our REP variable takes the value 1 for this county, otherwise it takes the value 0.  
  
Finally, we remove from our dataset the initial variables: age6574, age75, pop.density, pop, democrat and republican, as we don’t need them anymore.  
  
#remove the 3 categoricals: county, state, region  
counties <- counties[,-c(1:2,19)]  
  
#create 'older' variable  
counties$older<-counties$age6574 + counties$age75  
  
# create log.pop.density & log.pop  
counties$log.pop.density<-log(counties$pop.density+1, 10)  
counties$log.pop<-log(counties$pop+1, 10)  
  
#create response var  
counties$REP <- ifelse((counties$republican > counties$democrat),1,0)  
counties$REP<-as.factor(counties$REP)  
View(counties)  
  
# omit vars age6574, age75, pop.density, dem,rep+perot  
  
counties$pop.density<-NULL  
counties$age6574<-NULL  
counties$age75<-NULL  
counties$democrat<-NULL  
counties$republican<-NULL  
counties$pop<-NULL  
View(counties)

# boxplots  
boxplot(counties[,1:6],main="1-6",col='red')

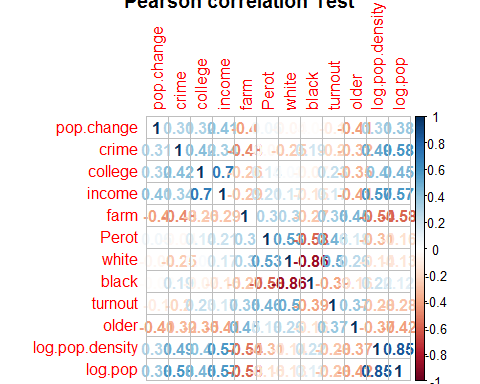


boxplot(counties[,7:13],main="7-13",col='red')



*Observing the boxplots, we see that the variables: crime, income, white, black, turnout and   
older contain a lot of outliers.*

#check correlation   
library(corrplot)  
  
corrplot(cor(counties[,-c(10,14)],method = "pearson"), method="num", main='Pearson correlation Test')



We check the correlation between the numerical independent variables and we observe that there is high negative correlation between the variables ‘white’ and ‘black’ ( -0.8 ) and   
high positive correlation between the variables ‘log.pop’ and ‘log.pop.density’ ( 0.86 ). This means that we can keep only one variable of each pair, in order to avoid the issue of multicollinearity (two variables containing the same statistical information).  
  
  
Now that we have made the necessary transformations to our dataset and have an idea about its descriptive characteristics, we are ready to divide our ‘counties’ dataset into training ( 80% ) and test ( 20% ) dataset, so as to continue with the modeling procedure.   
   
#create training-test set  
library(caret)

## Loading required package: lattice

counties\_sampling\_vector <- createDataPartition(counties$REP, p = 0.8, list = FALSE)  
  
counties\_train <- counties[counties\_sampling\_vector,]  
  
counties\_test <- counties[-counties\_sampling\_vector,]  
  
At first, we use the ‘step’ function, which helps us by auto-selecting the features which are statistically important for our predictive model. The stepwise feature selection works by using the AIC criterion, ignores possible collinearity and offers three methods of variable   
selection:  
i) the ‘forward’ method, which begins from the null model (containing only the constant   
term) and continues by adding the statistically important variables, achieving the best combination with the lowest AIC   
possible,  
ii) the ‘backward’ method, which begins from the full model (containing all the variables)  
and continues by taking out one by one the less important features, ending up again with   
the best combination with the lowest AIC,  
iii) the ‘both’ method, which uses both the ‘forward’ and ‘backward’ methods to suggest a   
model.  
All three methods conclude at the same combination of features and we usually use all of   
them in order to confirm our result.  
  
Another command that we use is the ‘glm’ function which adjusts the variables in a   
generalized linear model and as we want to use logistic regression, we set our model to   
belong to the Binomial distribution family (‘family’=binomial).   
For adjusting the logistic regression model we use the logit link. Glm (y ~ x1 + x2 + x3, family = binomial), where y represents the response variable (in this case is a categorical ( with values ​​of 1 0 )), and x1, x2, .. the independent variables. Since we are at the stage of   
choosing the important features we use the ‘counties\_train’ dataset.  
  
# modeling and step: auto variable selection  
options(scipen=999) #remove exponential annotation  
  
full<-glm(REP~., family = binomial, data = counties\_train)   
  
null<-glm(REP~1, family = binomial, data = counties\_train)  
  
step(null, scope=list(lower=null, upper=full), direction="forward")  
## Step: AIC=3072.79  
## REP ~ Perot + pop.change + farm + white + turnout + income +   
## log.pop.density + area + black  
##   
## Df Deviance AIC  
## <none> 3046.8 3072.8  
## + crime 1 3045.8 3073.8  
## + college 1 3046.5 3074.5  
## + log.pop 1 3046.7 3074.7  
## + older 1 3046.8 3074.8

##   
## Call: glm(formula = REP ~ Perot + pop.change + farm + white + turnout +   
## income + log.pop.density + area + black, family = binomial,   
## data = counties\_train)  
##   
## Coefficients:  
## (Intercept) Perot pop.change farm   
## -3.0974700 0.0370521 0.0125621 0.0394258   
## white turnout income log.pop.density   
## 0.0427171 -0.0688922 0.0001153 -1.0069867   
## area2 area3 area4 area5   
## -1.0569812 -0.2458444 -0.5924136 -0.6043237   
## black   
## 0.0150950   
##   
## Degrees of Freedom: 2491 Total (i.e. Null); 2479 Residual  
## Null Deviance: 3454   
## Residual Deviance: 3047 AIC: 3073

step(full, data=counties\_train, direction="backward")  
## Step: AIC=3072.79  
## REP ~ pop.change + income + farm + Perot + white + black + turnout +   
## area + log.pop.density  
##   
## Df Deviance AIC  
## <none> 3046.8 3072.8  
## - black 1 3050.0 3074.0  
## - Perot 1 3059.0 3083.0  
## - pop.change 1 3065.2 3089.2  
## - farm 1 3068.3 3092.3  
## - area 4 3074.7 3092.7  
## - white 1 3081.6 3105.6  
## - log.pop.density 1 3118.2 3142.2  
## - turnout 1 3118.3 3142.3  
## - income 1 3177.2 3201.2

##   
## Call: glm(formula = REP ~ pop.change + income + farm + Perot + white +   
## black + turnout + area + log.pop.density, family = binomial,   
## data = counties\_train)  
##   
## Coefficients:  
## (Intercept) pop.change income farm   
## -3.0974700 0.0125621 0.0001153 0.0394258   
## Perot white black turnout   
## 0.0370521 0.0427171 0.0150950 -0.0688922   
## area2 area3 area4 area5   
## -1.0569812 -0.2458444 -0.5924136 -0.6043237   
## log.pop.density   
## -1.0069867   
## Degrees of Freedom: 2491 Total (i.e. Null); 2479 Residual  
## Null Deviance: 3454   
## Residual Deviance: 3047 AIC: 3073

step(null, scope = list(upper=full), data=counties\_train, direction="both")  
## Step: AIC=3072.79  
## REP ~ Perot + pop.change + farm + white + turnout + income +   
## log.pop.density + area + black  
##   
## Df Deviance AIC  
## <none> 3046.8 3072.8  
## + crime 1 3045.8 3073.8  
## - black 1 3050.0 3074.0  
## + college 1 3046.5 3074.5  
## + log.pop 1 3046.7 3074.7  
## + older 1 3046.8 3074.8  
## - Perot 1 3059.0 3083.0  
## - pop.change 1 3065.2 3089.2  
## - farm 1 3068.3 3092.3  
## - area 4 3074.7 3092.7  
## - white 1 3081.6 3105.6  
## - log.pop.density 1 3118.2 3142.2  
## - turnout 1 3118.3 3142.3  
## - income 1 3177.2 3201.2

##   
## Call: glm(formula = REP ~ Perot + pop.change + farm + white + turnout +   
## income + log.pop.density + area + black, family = binomial,   
## data = counties\_train)  
##   
## Coefficients:  
## (Intercept) Perot pop.change farm   
## -3.0974700 0.0370521 0.0125621 0.0394258   
## white turnout income log.pop.density   
## 0.0427171 -0.0688922 0.0001153 -1.0069867   
## area2 area3 area4 area5   
## -1.0569812 -0.2458444 -0.5924136 -0.6043237   
## black   
## 0.0150950   
##   
## Degrees of Freedom: 2491 Total (i.e. Null); 2479 Residual  
## Null Deviance: 3454   
## Residual Deviance: 3047 AIC: 3073  
*\*\* For space economy I only include the final call for all three methods.\*\**RESULT: glm(formula = REP ~ Perot + pop.change + farm + white + turnout +   
 income + log.pop.density + area + black, family = binomial,   
 data = counties\_train)

All three methods end up proposing the same feature combination and now we   
are ready to train our logistic regression model and result to the best   
fitted one.  
Below, we use another command. The ‘summary’ function presents the results of the  
regression analysis of the model that has been adjusted. It gives the descriptive indexes of   
the residuals of the model fitted (deviance residuals), the coefficients of the estimators (Estimate) , the standard errors (standard errors), and the Wald control (z-value) of each of the p-values.  
 It also gives the Residual Deviance, which is the Deviance of the fitted model and   
the Null Deviance (the Deviance of the Null model), with each one’s degrees of freedom  
respectively. Finally, it also gives the value of AIC criterion.  
  
FIRST MODEL ATTEMPT  
best\_fit1 <- glm(REP ~ Perot + white + pop.change + farm + turnout + area +  
 income + log.pop.density + black,family = binomial,data = counties\_train)   
   
summary(best\_fit1)

## Call:  
## glm(formula = REP ~ Perot + white + pop.change + farm + turnout +   
## area + income + log.pop.density + black, family = binomial,   
## data = counties\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3724 -1.0741 0.4175 1.0196 2.4543   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.09747005 0.74962970 -4.132 0.000035962018072813 \*\*\*  
## Perot 0.03705211 0.01064608 3.480 0.000501 \*\*\*  
## white 0.04271714 0.00801837 5.327 0.000000099621998956 \*\*\*  
## pop.change 0.01256213 0.00302806 4.149 0.000033455826332965 \*\*\*  
## farm 0.03942577 0.00875144 4.505 0.000006635425867350 \*\*\*  
## turnout -0.06889216 0.00839250 -8.209 0.000000000000000223 \*\*\*  
## area2 -1.05698123 0.21264622 -4.971 0.000000667428876744 \*\*\*  
## area3 -0.24584444 0.19724935 -1.246 0.212631   
## area4 -0.59241357 0.22060930 -2.685 0.007245 \*\*   
## area5 -0.60432369 0.16774563 -3.603 0.000315 \*\*\*  
## income 0.00011534 0.00001064 10.840 < 0.0000000000000002 \*\*\*  
## log.pop.density -1.00698673 0.12426329 -8.104 0.000000000000000533 \*\*\*  
## black 0.01509505 0.00860827 1.754 0.079507 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3453.9 on 2491 degrees of freedom  
## Residual deviance: 3046.8 on 2479 degrees of freedom  
## AIC: 3072.8  
## Number of Fisher Scoring iterations: 4

Observing the summary results for best\_fit1, we conclude that we have to   
create a second by removing features with less than 3 \*\*\*.   
These features are indicated as non-statistically significant at all   
confidence intervals (p-value > 0.001 ) and thus less possible good predictors for our response variable.  
In our case, we remove the feature ‘black’ which is also correlated to ‘white’ feature.  
  
SECOND MODEL ATTEMPT  
best\_fit2 <- glm(REP ~ Perot + white + pop.change + farm + turnout + area +  
 income + log.pop.density,family = binomial,data = counties\_train)   
   
summary(best\_fit2)

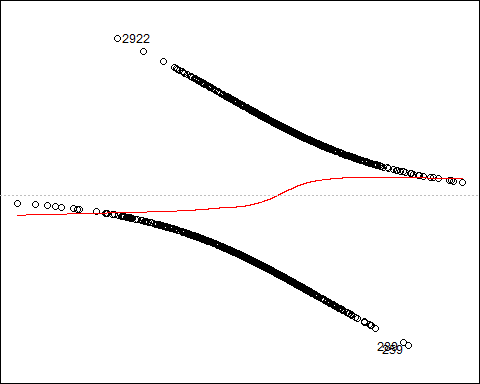
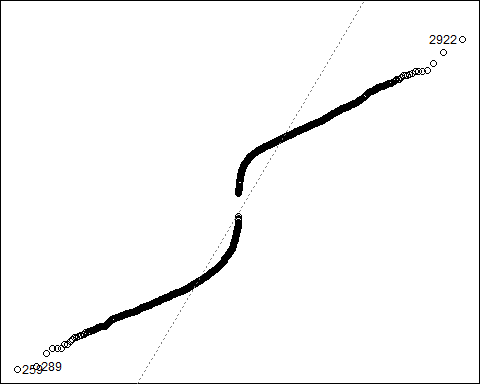
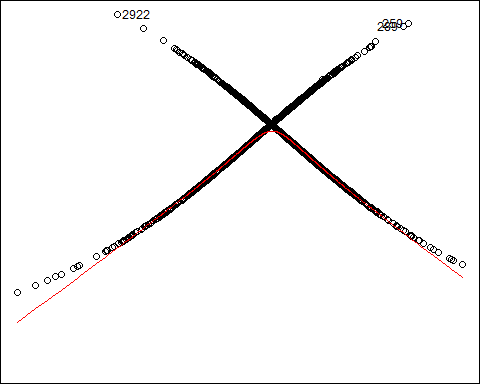
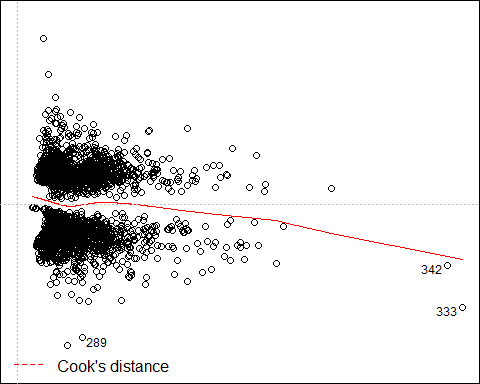
##   
## Call:  
## glm(formula = REP ~ Perot + white + pop.change + farm + turnout +   
## area + income + log.pop.density, family = binomial, data = counties\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3525 -1.0724 0.4103 1.0202 2.4772   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.05952561 0.42541033 -4.841 0.000001290129386107 \*\*\*  
## Perot 0.03853672 0.01058850 3.639 0.000273 \*\*\*  
## white 0.03101900 0.00411565 7.537 0.000000000000048146 \*\*\*  
## pop.change 0.01168970 0.00297186 3.933 0.000083729933047842 \*\*\*  
## farm 0.03932132 0.00873094 4.504 0.000006678837849058 \*\*\*  
## turnout -0.06574560 0.00818400 -8.033 0.000000000000000948 \*\*\*  
## area2 -1.14650784 0.20669384 -5.547 0.000000029079695463 \*\*\*  
## area3 -0.41154156 0.17265543 -2.384 0.017144 \*   
## area4 -0.75265469 0.20128299 -3.739 0.000185 \*\*\*  
## area5 -0.70748770 0.15745066 -4.493 0.000007009725323939 \*\*\*  
## income 0.00011486 0.00001063 10.809 < 0.0000000000000002 \*\*\*  
## log.pop.density -0.98024019 0.12302804 -7.968 0.000000000000001618 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3453.9 on 2491 degrees of freedom  
## Residual deviance: 3050.0 on 2480 degrees of freedom  
## AIC: 3074  
##   
## Number of Fisher Scoring iterations: 4

*Observing the summary of best\_fit2, we conclude that this is our best fitted   
model as we’ve reached a state where all our predictor variables are significantly important at all confidence intervals (plus, if we try removing any of   
the remaining features the AIC gets higher, something that we don’t want.)*  
glm(formula = REP ~ Perot + white + pop.change + farm + turnout +   
 area + income + log.pop.density, family = binomial, data = counties\_train)  
 *Important features:* Perot, white, pop.change, farm, turnout, area, income,   
 log.pop.density  
  
Number of Fisher Scoring iterations: 4, within the acceptable range ( 4 – 8) *As we see the AIC: 3074 takes a rather high value, which leads us to the   
conclusion that a logistic regression classifier is probably not a suited   
option for our project.*  
  
As we deploy a logistic regression classifier, it is more suitable to talk about odds ratio.   
Odds are determined from probabilities and range between 0 and infinity. They are defined as the ratio of the probability of success and the probability of failure.  
To receive the odds for our estimators we need to exponentiate their coefficients.  
  
# odds ratio and 95% CI  
exp(cbind(OR = coef(best\_fit2), confint(best\_fit2)))

## Waiting for profiling to be done...

## OR 2.5 % 97.5 %  
## (Intercept) 0.1275144 0.05490509 0.2912352  
## Perot 1.0392889 1.01800833 1.0611696  
## white 1.0315051 1.02333118 1.0399848  
## pop.change 1.0117583 1.00597513 1.0177656  
## farm 1.0401046 1.02268591 1.0583083  
## turnout 0.9363690 0.92133248 0.9513832  
## area2 0.3177444 0.21137969 0.4755106  
## area3 0.6626280 0.47183981 0.9286982  
## area4 0.4711142 0.31713783 0.6983917  
## area5 0.4928809 0.36137555 0.6700349  
## income 1.0001149 1.00009428 1.0001360  
## log.pop.density 0.3752210 0.29399942 0.4762952  
  
To begin with, all odds ratio belong in the range of their 95% confidence interval.  
Now we can say that for a one unit increase in voting for ‘Perot’, helps increase the odds of   
voting for the republican candidate (versus not voting for republicans) by a factor   
of 1.04. This is also logical, because this year’s winner was the democrat winner. So people voting for Perot were actually depriving votes from the democrat candidate, thus helping the republican one.  
Similarly, a one unit increase in ‘white’, ‘pop.change’, ‘farm’, ‘turnout’, ‘area 2’, ‘area3’, ‘area4’, ‘area5’, ‘income’, ‘log.pop.density’ helps increase the odds of voting for the republican   
candidate by a factor of 1.03, 1.01, 1.04, 0.93, 0.31, 0.66, 0.47, 0.49, 1, 0.37   
respectively. Comparing the odds ratio, we see that the ‘area’ and ‘log.pop.density’   
features are the less powerful predictors of our model.

Below we cite the diagnostic plots of our model.  
The diagnostic plots in logistic regression are the same as linear regression though now we   
draw conclusions by taking into consideration the assumptions of logistic regression.   
In general, we try to avoid using them as these plots are more intended for linear regression and sometimes could be misleading for logistic regression.

# log-reg plots  
plot(best\_fit2)  
  
  
1. Residuals vs Fitted  
  
  
The Residuals vs Fitted plot help you see, if there are curvilinear trends.  
But the fit of a logistic regression is curvilinear by nature, so you can have curvilinear trends in the residuals.  
  
  
  
  
  
2. Normal Q-Q  
  
  
The Normal Q-Q plot helps you see if your residuals are of normal distribution.   
But we no longer require a normality assumption for residuals in logistic regression.   
  
  
3. Scale Location  
  
  
The Scale Location plot can help you identify heteroscedasticity.   
However, in logistic regression we don’t need the homoscedasticity assumption.  
  
4. Residuals vs Leverage  


The Residuals vs Leverage can help you identify extreme outliers, known as leverage points.   
In this case, observations 289 333 and 342 that are below the thin red line are possible   
leverage points.  
  
Now, we continue with some model evaluation metrics.One measure of model fit is the significance of the overall model.   
This test asks whether the model with predictors fits better than a null model.   
The test statistic is the difference between the model deviance and the null model, which is of chi-square distribution with degrees of freedom equal to the difference in degrees of   
freedom between the current and the null model. In addition we also use the Hosmer   
Lemeshow goodness of fit test in order to double check the result.  
  
In our case, we see that the difference between null deviance and model deviance is 403.86 with 11 degrees of freedom. The deviance difference is rather high which explains away the variance observed by the response variable. The model p-value which is way less than 0.001 tells us that our model as a whole fits significantly better than an empty model. Our pseudo R square is equal to 0.12 which means that our model is said to explain only the 12% of the   
null deviance. This is not high enough, thus we assume that we don’t have an accurate   
enough logistic model. We cross-validate the poor fit of our model also by the result of the   
Hosmer – Lemeshow test, p – value < 0.00000000000000022, way smaller than 0.05.# difference between null deviance and model deviance  
with(best\_fit2, null.deviance - deviance)

## [1] 403.8559

# DoF (number of predictors)  
with(best\_fit2, df.null - df.residual)

## [1] 11

# model p-value  
with(best\_fit2, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))

## [1] 0.00000000000000000000000000000000000000000000000000000000000000000000000000000009291148  
  
#pseudo R\_squared  
(pseudo\_R\_squared <- with(best\_fit2, 1 - deviance / null.deviance))

## [1] 0.1169286

# Hosmer Lemeshow test  
library(ResourceSelection)

## Warning: package 'ResourceSelection' was built under R version 3.2.5

## ResourceSelection 0.2-6 2016-02-15

hoslem.test(counties\_train$REP, fitted(best\_fit2))

## Warning in Ops.factor(1, y): '-' not meaningful for factors

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: counties\_train$REP, fitted(best\_fit2)  
## X-squared = 2492, df = 8, p-value < 0.00000000000000022

Although not having a good fit model, now we can create the predicted probabilities.   
In order to generate the fitted results we use the ‘predict’ function. We take the predicted   
probabilities by applying the ‘predict’ function on the values of the ‘counties\_test’ dataset   
and setting the type=’response’.  
  
We can perform binary classification by applying a threshold of 0.5.   
If the predicted probability is above 0.5 then it belongs to class 1 (voter for republican),   
while if it is below 0.5, it belongs to class 0 (voter against republican).  
In addition, we’ll measure the classification accuracy by taking the mean of our expected   
values and actual values and also create a confusion matrix.   
In a confusion matrix, we can get the number of the correctly identified instances of class 1, which is also known as True Positive rate and the number of the correctly classified to class 0 (correctly rejected), which is also known as True Negative rate. There is also the False   
Positive rate(incorrectly identified - Type I error) and False Negative rate ( incorrectly   
rejected Type II error).  
  
In our case , we see that our model accuracy reaches the level of 0.69, which is not a good   
score. From the confusion matrix, we realize that 232 observations out of 622 ( counties\_test number of observations) were correctly identified as voters of the republican candidate,  
192 were correctly identified as voters of the republicans, 113 were incorrectly rejected as   
voters of the republicans and 85 were incorrectly rejected as voters of the republicans.  
  
  
# predictions, accuracy, confusion matrix  
fitted.results <- predict(best\_fit2,newdata=counties\_test,type='response')  
fitted.results <- ifelse(fitted.results > 0.5,1,0)  
mean <- mean(fitted.results == counties\_test$REP)  
paste('Accuracy',mean)

## [1] "Accuracy 0.681672025723473"

(confusion\_matrix<-table(predicted=fitted.results,observed=counties\_test$REP))

## observed  
## predicted 0 1  
## 0 192 85  
## 1 113 232

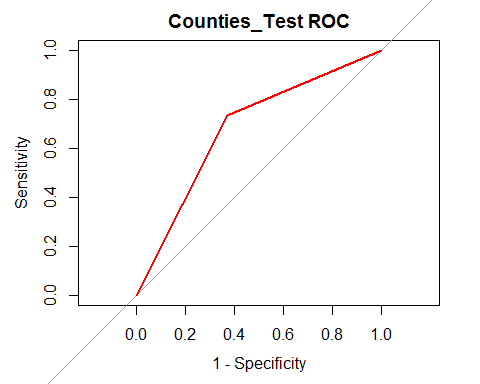
Finally another accuracy metric is the ROC Area Under Curve (AUC).The area under the classifier is just 0.5. The area under a perfect classifier is 1 as the curve   
passes through the point (0,1) and this is when we have to face the problem of overfitting.   
  
In our case, the AUC equals to 0.68 which indicates poor accuracy of our model to classify   
correctly the observations as voters or non- voters of the republicans.  
  
We can also plot the AUC with its ROC curve by using the library ‘pROC’ and the   
‘roc’ function. The ‘roc’ function uses the ‘Sensitivity’, which is defined as *TP/TP*+*FN*.   
Intuitively this metric corresponds to the proportion of positive data points that are   
correctly considered as positive, with respect to all positive data points and also uses the   
‘Specificity’, which is defined as *TN/TN*+*FP*. Intuitively this metric corresponds to the   
proportion of negative data points that are correctly considered as negative, with respect to   
all negative data points.  
# ROC curve kai AUC  
library(pROC)

## Type 'citation("pROC")' for a citation.  
## Attaching package: 'pROC'  
## The following objects are masked from 'package:stats':  
## cov, smooth, var

roc1 <- roc(counties\_test$REP,fitted.results)  
(auc <- roc1$auc)

## Area under the curve: 0.6807

plot(roc1,col="red",main="Counties\_Test ROC",legacy.axes=TRUE)



## Call:  
## roc.default(response = counties\_test$REP, predictor = fitted.results)  
## Data: fitted.results in 305 controls (counties\_test$REP 0) < 317 cases   
(counties\_test$REP 1).  
## Area under the curve: 0.6807