Stt864 Lab3

Nan Cao

March 15, 2016

```
setwd("C://Users//nan66//Google Drive//stt864//LAB3")
set.seed(52871775)
library(MASS)
```

Q1 Setting default path, loading need library and read data.

```
polls2008<-read.csv(file="2008-polls.csv",header=TRUE)
polls2012<-read.csv(file="2012-polls.csv",header=TRUE)
results2008<-read.csv(file="2008-results.csv",header=TRUE)
results2012<-read.csv(file="2012-results.csv",header=TRUE)
# select pollsters that conducted polls >5 states.
atleast5<-table(polls2008[,5])[table(polls2008[,5])>=5]
atleast5
```

```
##
##                 ARG       ArizonaStateU       CapitalSurvey
##                  78                   6                   7
##           ElwayPoll             EPICMRA FairleighDickinsonU
##                   5                   8                   8
##   FinancialDynamics            GfKRoper    InsiderAdvantage
##                  16                   8                  40
##          MaristColl          MasonDixon           MonmouthU
##                  17                  44                   5
##      MuhlenbergColl     OpinionResearch     PrincetonSurvey
##                  10                  51                   5
##         QuinnipiacU           Rasmussen              Selzer
##                  60                 359                   8
##           SienaColl            SuffolkU           SurveyUSA
##                  12                  15                 243
##       UofCincinnati     UofNewHampshire        UofWisconsin
##                   5                   6                  20
##               Zogby
##                  16
```

```
pollers<-c("ARG", "EPICMRA", "InsiderAdvantage",
           "MaristColl", "MasonDixon", "MuhlenbergColl",
           "QuinnipiacU", "Rasmussen", "SienaColl",
           "SuffolkU","SurveyUSA", "UofCincinnati",
           "UofNewHampshire","Zogby")
po08sub<-polls2008[polls2008[,5]%in%pollers,]
po12sub<-polls2012[polls2012[,5]%in%pollers,]
```

Q2

```r
#reformatting the poll and true results dataset as desired
#Dem win=1 Rep win=0
winers2008<-(results2008[,2]-results2008[,3]>0)+0
#name of 51 states
StateID2008<-results2008[,1]
Allresp<-NULL
for (sid in 1:51){
  ##operate on state=sid
    po08subID<-po08sub$State==StateID2008[sid]
    #polls (at least5),
    PoWin08SubSta<-po08sub[po08subID,]
    #Dem win=1 Rep win=0
    PoWin08Sta<-(PoWin08SubSta[,2]-PoWin08SubSta[,3]>0)+0
    #whether the polls is correct
    pollwinersIND<-(PoWin08Sta==winers2008[sid])+0
    #conbine it to "Allresp"
    Allresp<-c(Allresp,pollwinersIND)
}
#absolute difference between Supp rates of Dem and Rep.
margins2008<-abs(po08sub[,2]-po08sub[,3])
lagtime2008<-rep(0,dim(po08sub)[1])
electiondate2008<-c("Nov 04 2008")
EdDa08<-as.Date(electiondate2008, format="%b %d %Y")
for (i in 1:dim(po08sub)[1]){
    StDa08<-as.Date(as.character(po08sub[i,4]), format="%b %d %Y")
    lagtime2008[i]<-EdDa08-StDa08
}
data08<-cbind(Allresp,as.character(po08sub[,1]),margins2008,lagtime2008,
              as.character(po08sub[,5]))
```

Q3

```r
# select the states with at least one failure prediction
# aka data08$Allresp=0
stateslist<-unique(data08[which(data08[,1]=="0"),2])
subdata08<-data08[data08[,2]%in%stateslist,]
```

Q4 Q5

```r
# define new variables and fit a logistic regression model
resp<-as.integer(subdata08[,1])
statesFAC<-as.factor(subdata08[,2])
margins<-as.double(subdata08[,3])
lagtime<-as.double(subdata08[,4])
pollersFAC<-as.factor(subdata08[,5])
logitreg<-glm(resp~statesFAC+margins+lagtime+pollersFAC,family="binomial")
summary(logitreg)
```

```
##
## Call:
## glm(formula = resp ~ statesFAC + margins + lagtime + pollersFAC,
##     family = "binomial")
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5779  -0.5603   0.2084   0.5631   2.5537
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.477331   0.602829   0.792 0.428466
## statesFACFL                 -1.647375   0.547285  -3.010 0.002612 **
## statesFACGA                  1.354619   1.157192   1.171 0.241756
## statesFACIN                 -3.359969   0.926903  -3.625 0.000289 ***
## statesFACMA                  2.064918   1.233361   1.674 0.094087 .
## statesFACMI                 -0.109506   0.733656  -0.149 0.881348
## statesFACMN                  1.576421   0.909859   1.733 0.083167 .
## statesFACMO                 -0.427149   0.614079  -0.696 0.486683
## statesFACMT                  1.572071   1.165776   1.349 0.177491
## statesFACNC                 -2.289227   0.641511  -3.568 0.000359 ***
## statesFACND                  0.582515   1.411664   0.413 0.679867
## statesFACNH                  0.608812   0.770412   0.790 0.429386
## statesFACNJ                  0.562342   0.953698   0.590 0.555429
## statesFACNM                  0.115791   0.722887   0.160 0.872741
## statesFACNV                 -0.782439   0.620767  -1.260 0.207511
## statesFACNY                  1.106166   1.220608   0.906 0.364808
## statesFACOH                 -1.456813   0.554890  -2.625 0.008655 **
## statesFACOR                  2.466634   1.227231   2.010 0.044440 *
## statesFACPA                  0.999567   0.706504   1.415 0.157125
## statesFACVA                 -0.764514   0.578862  -1.321 0.186595
## statesFACWA                  2.049390   1.222229   1.677 0.093589 .
## statesFACWI                  1.724056   0.952639   1.810 0.070332 .
## statesFACWV                  0.176470   1.192351   0.148 0.882341
## margins                      0.243394   0.038387   6.341 2.29e-10 ***
## lagtime                     -0.010550   0.001722  -6.128 8.89e-10 ***
## pollersFACEPICMRA            1.884727   1.341388   1.405 0.160004
## pollersFACInsiderAdvantage   0.831820   0.586503   1.418 0.156112
## pollersFACMaristColl         1.899700   1.201596   1.581 0.113883
## pollersFACMasonDixon         0.368782   0.590033   0.625 0.531958
## pollersFACMuhlenbergColl    -0.107470   1.516623  -0.071 0.943508
## pollersFACQuinnipiacU        1.742448   0.629726   2.767 0.005658 **
## pollersFACRasmussen          0.273553   0.451894   0.605 0.544948
## pollersFACSienaColl         15.026258 542.747543   0.028 0.977913
## pollersFACSuffolkU           1.166058   0.920064   1.267 0.205024
## pollersFACSurveyUSA          0.831435   0.518039   1.605 0.108501
## pollersFACUofCincinnati      0.399582   1.113652   0.359 0.719742
## pollersFACUofNewHampshire   -1.361725   1.333940  -1.021 0.307335
## pollersFACZogby              0.501113   0.745531   0.672 0.501484
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 792.52  on 662  degrees of freedom
## Residual deviance: 492.68  on 625  degrees of freedom
## AIC: 568.68
##
```

```
## Number of Fisher Scoring iterations: 15
```

Based on the fitted modle, statesFACFL, statesFACIN, statesFACNC, statesFACOH, statesFACOR, statesFACWV, margins, lagtime, pollersFACQuinnipiacU are significantly (p-value<0.05) associated with Resp.

```
#Fit a simple logistic regression model without states as covariates
logitreg1<-glm(resp~margins+lagtime+pollersFAC,family="binomial")
summary(logitreg1)
```

```
##
## Call:
## glm(formula = resp ~ margins + lagtime + pollersFAC, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3334  -0.8649   0.3761   0.7965   2.1286
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -0.271191   0.355706  -0.762   0.4458
## margins                       0.284972   0.032126   8.870  < 2e-16 ***
## lagtime                      -0.005811   0.001352  -4.298 1.72e-05 ***
## pollersFACEPICMRA             1.801304   1.174831   1.533   0.1252
## pollersFACInsiderAdvantage    0.475944   0.498685   0.954   0.3399
## pollersFACMaristColl          1.824185   1.131871   1.612   0.1070
## pollersFACMasonDixon          0.149939   0.508331   0.295   0.7680
## pollersFACMuhlenbergColl      0.837409   1.265836   0.662   0.5083
## pollersFACQuinnipiacU         1.041485   0.521115   1.999   0.0457 *
## pollersFACRasmussen           0.028280   0.378125   0.075   0.9404
## pollersFACSienaColl          15.472666 531.264480   0.029   0.9768
## pollersFACSuffolkU            0.878430   0.877534   1.001   0.3168
## pollersFACSurveyUSA           0.454879   0.422271   1.077   0.2814
## pollersFACUofCincinnati      -0.693628   1.028374  -0.674   0.5000
## pollersFACUofNewHampshire    -0.685772   1.189587  -0.576   0.5643
## pollersFACZogby              -0.490406   0.655532  -0.748   0.4544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 792.52  on 662  degrees of freedom
## Residual deviance: 620.09  on 647  degrees of freedom
## AIC: 652.09
##
## Number of Fisher Scoring iterations: 15
```

```
anova(logitreg1,logitreg)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ margins + lagtime + pollersFAC
## Model 2: resp ~ statesFAC + margins + lagtime + pollersFAC
##   Resid. Df Resid. Dev Df Deviance
```

```
## 1       647     620.09
## 2       625     492.68 22   127.41
```

```
anova(logitreg1,logitreg,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ margins + lagtime + pollersFAC
## Model 2: resp ~ statesFAC + margins + lagtime + pollersFAC
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       647     620.09
## 2       625     492.68 22   127.41 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# reformating the 2012 poll data for prediction purpose
pollwiners2012<-(po12sub[,2]-po12sub[,3]>0)+0
margins2012<-abs(po12sub[,2]-po12sub[,3])
lagtime2012<-rep(0,dim(po12sub)[1])
electiondate2012<-c("Nov 06 2012")
EdDa12<-as.Date(electiondate2012, format="%b %d %Y")
for (i in 1:dim(po12sub)[1]){
  StDa12<-as.Date(as.character(po12sub[i,4]),format="%b %d %Y")
  lagtime2012[i]<-EdDa12-StDa12
}
data12<-cbind(pollwiners2012,as.character(po12sub[,1]),margins2012,lagtime2012,
              as.character(po12sub[,5]))
```

Based on the LRT test, P-value< 2.2e-16, the categorical variable SA is very significant, the first logistic model with SA is better. Q6 Q7 Q8 Q9

```
subdata12<-data12[data12[,2]%in%stateslist,]
# Predict with logistic and simple logistic model
margins2012<-as.double(subdata12[,3])
lagtime2012<-as.double(subdata12[,4])
pollersFAC2012<-as.factor(subdata12[,5])
StateName<-c("FL","MI","MO","CO")
n<-c(0,0,0,0)
SSE<-c(0,0,0,0)
SSE1<-c(0,0,0,0)
# Pred for Q6
Pred<-matrix(0,4,4)
rownames(Pred)<-StateName
colnames(Pred)<-c("LDemWin","LRepWin","sLDemWin","sLRepWin")
# Pred for Q7
Pred1<-matrix(0,4,4)
rownames(Pred1)<-StateName
colnames(Pred1)<-c("LDemWin","LRepWin","sLDemWin","sLRepWin")
# log regression CI for Q8
PredCI<-matrix(0,4,4)
rownames(PredCI)<-StateName
colnames(PredCI)<-c("DemWinLow","DemWinUp","RepWinLow","RepWinUp")
# sample log regression CI for Q8
```

```
PredCI1<-matrix(0,4,4)
rownames(PredCI1)<-StateName
colnames(PredCI1)<-c("DemWinLow","DemWinUp","RepWinLow","RepWinUp")
# bootstrap CI of log regression for Q8
BSCI<-matrix(0,4,4)
rownames(BSCI)<-StateName
colnames(BSCI)<-c("DemWinLow","DemWinUp","RepWinLow","RepWinUp")
# bootstrap CI of sample log regression for Q8
BSCI1<-matrix(0,4,4)
rownames(BSCI1)<-StateName
colnames(BSCI1)<-c("DemWinLow","DemWinUp","RepWinLow","RepWinUp")
# Weighted Pred of log regression for Q9
S.Pred<-matrix(0,4,4)
rownames(S.Pred)<-StateName
colnames(S.Pred)<-c("LDemWin","LRepWin","sLDemWin","sLRepWin")
# computation of weight of pollers for Q9
poNum<-length(pollers)
ErrRates<-rep(0,poNum)
for (poID in 1:poNum){
  PoPred08<-NULL
  PoPredRt<-NULL
  Po08Rt<-NULL
  StateID<-NULL
  PoPred08<-po08sub[po08sub[,5]==pollers[poID],]
  PoPredRt<-(PoPred08[,2]>PoPred08[,3])+0
  StateID<-PoPred08[,1]
  Po08Rt<-winers2008[StateID]
  ErrRates[poID]<-sum(1-(PoPredRt==winers2008[StateID]))/length(PoPredRt)
}
PoRank<-rank(ErrRates)
Weight<-1/(PoRank^2)
PollWeight<-data.frame(pollers,Weight)
#the loop
for (k in 1:4){
    # number of locations
    NOpolls<-sum(subdata12[,2]==StateName[k])
    locations<-which(subdata12[,2]==StateName[k])
    n[k]<-length(locations)
    # clearance of variables in the loop
    probDemwin<-NULL
    probGopwin<-NULL
    subWeight<-NULL
    X<-matrix(0,n[k],4)
    Xs<-matrix(0,n[k],3)
    X[,1]<-1
    Xs[,1]<-1
    X[,4]<-1
    Xs[,3]<-1
    # container of predictions
    LogPR<-cbind(as.double(subdata12[locations,1]),
        rep(0,n[k]),rep(0,n[k]),rep(0,n[k]))
    sLogPR<-cbind(as.double(subdata12[locations,1]),
        rep(0,n[k]),rep(0,n[k]),rep(0,n[k]))
```

```
DeLogPR<-cbind(as.double(subdata12[locations,1]),
    rep(0,n[k]),rep(0,n[k]),rep(0,n[k]),rep(0,n[k]))
DesLogPR<-cbind(as.double(subdata12[locations,1]),
    rep(0,n[k]),rep(0,n[k]),rep(0,n[k]))
counts<-0
for (i in locations){
    counts<-counts+1
    LogDPs<-data.frame(statesFAC=StateName[k],margins=margins2012[i],
                       lagtime=lagtime2012[i], pollersFAC=pollersFAC2012[i])
    sLogDPs<-data.frame(margins=margins2012[i],lagtime=lagtime2012[i],
                        pollersFAC=pollersFAC2012[i])
    X[counts,2:3]<-as.matrix(LogDPs[2:3])
    Xs[counts,1:2]<-as.matrix(sLogDPs[1:2])
    LogPR[counts,2:4]<-unlist(predict(logitreg,LogDPs,type="response",se.fit=TRUE))
    sLogPR[counts,2:4]<-unlist(predict(logitreg1,sLogDPs,type="response",se.fit=TRUE))
    #derivative
    DeLogPR[counts,2:5]<-(1-LogPR[counts,2])*LogPR[counts,2]*unlist(LogDPs)
    DesLogPR[counts,2:4]<-(1-sLogPR[counts,2])*sLogPR[counts,2]*unlist(sLogDPs)
}
SSE<-sum(LogPR[,3]^2)
SSE1<-sum(sLogPR[,3]^2)
P1<-LogPR[,1]*LogPR[,2]+(1-LogPR[,1])*(1-LogPR[,2])
P2<-sLogPR[,1]*sLogPR[,2]+(1-sLogPR[,1])*(1-sLogPR[,2])

# Predictions for Q6
Pred[k,1]<-mean(P1)
Pred[k,2]<-mean(1-P1)
Pred[k,3]<-mean(P2)
Pred[k,4]<-mean(1-P2)
# Predictions for Q7
Pred1[k,1]<-mean(P1>0.5+0)
Pred1[k,2]<-mean(P1<0.5+0)
Pred1[k,3]<-mean(P2>0.5+0)
Pred1[k,4]<-mean(P2<0.5+0)
# Q8.1
# clearance of containers
varbeta<-NULL
varbeta1<-NULL
# define the function of var of two models
VarGBeta<-function(x){
    x<-as.matrix(x[2:5])
    return(t(x)%*%ginv(t(X)%*%diag(P1*(1-P1))%*%X)%*%x)
}
VarGBeta1<-function(x){
    x<-as.matrix(x[2:4])
    return(t(x)%*%ginv(t(Xs)%*%diag(P2*(1-P2))%*%Xs)%*%x)
}
varbeta<-apply(DeLogPR,1,VarGBeta)
varbeta1<-apply(DesLogPR,1,VarGBeta1)
PredCI[k,1]<-mean(P1)-qnorm(0.975)*sqrt(mean(varbeta)/n[k])
PredCI[k,2]<-mean(P1)+qnorm(0.975)*sqrt(mean(varbeta)/n[k])
PredCI[k,3]<-1-PredCI[k,2]
PredCI[k,4]<-1-PredCI[k,1]
```

```
    PredCI1[k,1]<-mean(P2)-qnorm(0.975)*sqrt(mean(varbeta1)/n[k])
    PredCI1[k,2]<-mean(P2)+qnorm(0.975)*sqrt(mean(varbeta1)/n[k])
    PredCI1[k,3]<-1-PredCI1[k,2]
    PredCI1[k,4]<-1-PredCI1[k,1]
    # Q8.2 Bootstrap prediction CI
    xStar<-NULL
    xStar1<-NULL
    for (boot in 1:500){
        xStar[boot]<-mean(sample(P1,n[k],replace=TRUE))
        xStar1[boot]<-mean(sample(P2,n[k],replace=TRUE))
    }
    BSCI[k,1]<-mean(xStar)-qnorm(0.975)*sd(xStar)
    BSCI[k,2]<-mean(xStar)+qnorm(0.975)*sd(xStar)
    BSCI[k,3]<-mean(1-xStar)-qnorm(0.975)*sd(1-xStar)
    BSCI[k,4]<-mean(1-xStar)+qnorm(0.975)*sd(1-xStar)
    BSCI1[k,1]<-mean(xStar1)-qnorm(0.975)*sd(xStar1)
    BSCI1[k,2]<-mean(xStar1)+qnorm(0.975)*sd(xStar1)
    BSCI1[k,3]<-mean(1-xStar1)-qnorm(0.975)*sd(1-xStar1)
    BSCI1[k,4]<-mean(1-xStar1)+qnorm(0.975)*sd(1-xStar1)
    #Q9 Silver's approach
    subPoName<-subdata12[subdata12[,2]==StateName[k],5]
    subWeight<-rep(0,NOpolls)
    for (f in 1:NOpolls){
      subWeight[f]<-PollWeight[PollWeight[,1]==subPoName[f],2]
    }
    W.P1<-P1*subWeight/sum(subWeight)
    W.P2<-P2*subWeight/sum(subWeight)
    W.P1.C<-(1-P1)*subWeight/sum(subWeight)
    W.P2.C<-(1-P2)*subWeight/sum(subWeight)
    S.Pred[k,1]<-sum(W.P1)
    S.Pred[k,2]<-sum(W.P1.C)
    S.Pred[k,3]<-sum(W.P2)
    S.Pred[k,4]<-sum(W.P2.C)
}
```

outputs:

```
#Out put of Q6
Pred
```

```
##      LDemWin   LRepWin  sLDemWin  sLRepWin
## FL 0.5530233 0.4469767 0.5779907 0.4220093
## MI 0.8434990 0.1565010 0.8421328 0.1578672
## MO 0.3170803 0.6829197 0.2893665 0.7106335
## CO 0.4906724 0.5093276 0.5221877 0.4778123
```

```
#Out put of Q7s
Pred1
```

```
##       LDemWin   LRepWin  sLDemWin  sLRepWin
## FL 0.57777778 0.4222222 0.6000000 0.4000000
## MI 0.93750000 0.0625000 0.9375000 0.0625000
## MO 0.07692308 0.9230769 0.1538462 0.8461538
## CO 0.52631579 0.4736842 0.4736842 0.5263158
```

```r
#Actuall election results in 2012
results2012[results2012[,1]%in%StateName,]
```

```
##    State  Dem  Rep
## 6     CO 51.2 46.5
## 10    FL 50.0 49.1
## 23    MI 54.3 44.8
## 25    MO 44.3 53.9
```

The prediction in Q6 is much more accurate in FL, MI & MO, however, in CO, the prediction in Q7 is better.

```r
#Out put of Q8
PredCI
```

```
##     DemWinLow  DemWinUp   RepWinLow  RepWinUp
## FL  0.3958377 0.7102088  0.28979123 0.6041623
## MI  0.5423551 1.1446428 -0.14464283 0.4576449
## MO -0.2871695 0.9213302  0.07866983 1.2871695
## CO  0.2289955 0.7523494  0.24765059 0.7710045
```

```r
PredCI1
```

```
##      DemWinLow  DemWinUp   RepWinLow  RepWinUp
## FL  0.26281080 0.8931705  0.10682946 0.7371892
## MI  0.17111475 1.5131508 -0.51315081 0.8288853
## MO -0.87528353 1.4540165 -0.45401655 1.8752835
## CO -0.04291397 1.0872894 -0.08728944 1.0429140
```

```r
BSCI
```

```
##    DemWinLow  DemWinUp  RepWinLow  RepWinUp
## FL 0.4840409 0.6217617 0.37823827 0.5159591
## MI 0.7564992 0.9284138 0.07158616 0.2435008
## MO 0.2131528 0.4203838 0.57961618 0.7868472
## CO 0.3626699 0.6277566 0.37224340 0.6373301
```

```r
BSCI1
```

```
##    DemWinLow  DemWinUp  RepWinLow  RepWinUp
## FL 0.5093387 0.6489122 0.35108776 0.4906613
## MI 0.7625510 0.9216922 0.07830783 0.2374490
## MO 0.1913550 0.3759988 0.62400123 0.8086450
## CO 0.4241494 0.6211532 0.37884681 0.5758506
```

```r
#Out put of Q9
S.Pred
```

```
##      LDemWin   LRepWin  sLDemWin  sLRepWin
## FL 0.6099908 0.3900092 0.7777139 0.2222861
## MI 0.8713073 0.1286927 0.8814376 0.1185624
## MO 0.3189026 0.6810974 0.2979604 0.7020396
## CO 0.5571267 0.4428733 0.5809884 0.4190116
```

Intervals based Silver's approach cover all the 4 actual election results. All other intervals we gotten previously couldn't do this. However Intervals based Silver's approach are much wider than other intervals.