```
######################################################################
#################### Logit and Probit Regression ###################
######################################################################

set.seed(112460) ## Seed for replicability

require(plyr) ## Utility tools
require(dplyr)## Utility tools
require(lmtest) ## Supplemental and postestimation tests
require(sandwich) ## Specific for the sandwich version of robust SE
calculations
require(ggplot2) ## Graphical presentation
require(stargazer) ## Tables
require(car) # Companion to Applied Regression
require(carData) # Supplemental Data
require(aod) # For wald.test
require(stats4) # For BIC
library(tidyverse)


binarydata<-ANES_2000

      #### party ID ####
binarydata<- ANES_2000 %>% filter(VCF0303 %in% c("1","3"))
binarydata$VCF0303<-as.numeric(binarydata$VCF0303)
binarydata$VCF0303<-replace(binarydata$VCF0303, binarydata$VCF0303==3, 0)
names(binarydata) [names(binarydata)== 'VCF0303'] <- 'PartyID'
CrossTable(binarydata$PartyID)

      #### gender ####
binarydata <- binarydata %>%
  filter(VCF0104 %in% c("1", "2", "3"))
names(binarydata) [names(binarydata)== 'VCF0104'] <- 'Gender'

      #### race ####
binarydata <- binarydata %>%
  filter(VCF0105a %in% c("1", "2", "3", "4", "5", "6"))
names(binarydata) [names(binarydata)== 'VCF0105a'] <- 'Race'

binarydata2 <- binarydata %>%
  mutate(White = Race)
binarydata$White<-replace(binarydata2$White, binarydata2$White==1, 0)
binarydata$White<-replace(binarydata2$White, binarydata2$White==2, 1)
binarydata$White<-replace(binarydata2$White, binarydata2$White==3, 0)
binarydata$White<-replace(binarydata2$White, binarydata2$White==4, 0)
binarydata$White<-replace(binarydata2$White, binarydata2$White==5, 0)
binarydata$White<-replace(binarydata2$White, binarydata2$White==6, 0)
summary(binarydata2)

binarydata3 <- binarydata %>%
  mutate(Black = Race)
binarydata$Black<-replace(binarydata3$Black, binarydata3$Black==1, 0)
binarydata$Black<-replace(binarydata3$Black, binarydata3$Black==2, 1)
binarydata$Black<-replace(binarydata3$Black, binarydata3$Black==3, 0)
```

```
binarydata$Black<-replace(binarydata3$Black, binarydata3$Black==4, 0)
binarydata$Black<-replace(binarydata3$Black, binarydata3$Black==5, 0)
binarydata$Black<-replace(binarydata3$Black, binarydata3$Black==6, 0)
summary(binarydata3)

binarydata4 <- binarydata %>%
  mutate(Hispanic = Race)
binarydata$Hispanic<-replace(binarydata4$Hispanic, binarydata4$Hispanic==1,
0)
binarydata$Hispanic<-replace(binarydata4$Hispanic, binarydata4$Hispanic==5,
1)
binarydata$Hispanic<-replace(binarydata4$Hispanic, binarydata4$Hispanic==2,
0)
binarydata$Hispanic<-replace(binarydata4$Hispanic, binarydata4$Hispanic==3,
0)
binarydata$Hispanic<-replace(binarydata4$Hispanic, binarydata4$Hispanic==4,
0)
binarydata$Hispanic<-replace(binarydata4$Hispanic, binarydata4$Hispanic==6,
0)
summary(binarydata4)

binarydata5 <- binarydata %>%
  mutate(Asian = Race)
binarydata$Asian<-replace(binarydata5$Asian, binarydata5$Asian==1, 0)
binarydata$Asian<-replace(binarydata5$Asian, binarydata5$Asian==2, 1)
binarydata$Asian<-replace(binarydata5$Asian, binarydata5$Asian==3, 0)
binarydata$Asian<-replace(binarydata5$Asian, binarydata5$Asian==4, 0)
binarydata$Asian<-replace(binarydata5$Asian, binarydata5$Asian==5, 0)
binarydata$Asian<-replace(binarydata5$Asian, binarydata5$Asian==6, 0)
summary(binarydata5)

      #### education ####
binarydata <- binarydata %>%
  filter(VCF0110 %in% c("1", "2", "3", "4"))
names(binarydata) [names(binarydata)== 'VCF0110'] <- 'Education'

        #### age ####
binarydata <- binarydata %>%
  filter(VCF0102 %in% c("1", "2", "3"))
names(binarydata) [names(binarydata)== 'VCF0102'] <- 'Age'

        #### family income ####
binarydata <- binarydata %>%
  filter(VCF0114 %in% c("1", "2", "3", "4", "5"))
names(binarydata) [names(binarydata)== 'VCF0114'] <- 'Family_Income'
attach(binarydata)

      #### Regression Models ####
require(gmodels)
model1<-glm(as.factor(PartyID) ~ Gender + Education + Family_Income ,
family = binomial(link = "logit"), data=binarydata)
summary(model1)
model2<-glm(as.factor(PartyID) ~ Gender + Race, family = binomial(link =
"logit"), data=binarydata)
```

```
summary(model2)
model3<-glm(as.factor(PartyID) ~ Gender + Race + Education + Family_Income,
family = binomial(link = "logit"), data=binarydata)
summary(model3)
model4<-glm(as.factor(PartyID) ~ White + Race + Education + Family_Income,
family = binomial(link = "logit"), data=binarydata)
summary(model4)
model5<-glm(as.factor(PartyID) ~ Black + Race + Education + Family_Income,
family = binomial(link = "logit"), data=binarydata)
summary(model5)
model6<-glm(as.factor(PartyID) ~ Hispanic + Race + Education +
Family_Income, family = binomial(link = "logit"), data=binarydata)
summary(model6)
model7<-glm(as.factor(PartyID) ~ Asian + Race + Education + Family_Income,
family = binomial(link = "logit"), data=binarydata)
summary(model7)

      #### Crosstables ####
CrossTable(binarydata$PartyID)
CrossTable(binarydata$Race)
CrossTable(binarydata$White)
CrossTable(binarydata$Black)
CrossTable(binarydata$Hispanic)
CrossTable(binarydata$Asian)

    #### Summary of Models ####
summary(model1)
summary(model2)
summary(model3)
summary(model4)
summary(model5)
summary(model6)
summary(model7)

    #### Wald.Test ####
library(dplyr)
require(plyr)

wald.test.stars<- function(pvalue){
  if(pvalue<0.1 & pvalue>=0.05){return("*")
  } else if(pvalue<0.05 & pvalue>=0.01){return("**")
  } else if(pvalue<0.01){return("*")
  } else {return(" ")}
}

stargazer.wald.chi<- function(model){
  require(aod)
  w1<-wald.test(b = coef(model), Sigma = vcov(model), Terms =
2:length(model$coefficients))
  w1chi<-w1$result$chi2[1]
  return(format(round(w1chi, 3), nsmall=3))
}

stargazer.wald.sig<- function(model){
  require(aod)
  w1<-wald.test(b = coef(model), Sigma = vcov(model), Terms =
2:length(model$coefficients))
  w1p<-w1$result$chi2[3]
  starw1<-wald.test.stars(w1p)
  return(starw1)
}

stargazer.wald.output<-function(model){
  out<-paste(stargazer.wald.chi(model), stargazer.wald.sig(model))
  return(out)
}

wald.test(b = coef(model1), Sigma = vcov(model1), Terms =
2:length(model1$coefficients))
wald.test(b = coef(model2), Sigma = vcov(model2), Terms =
2:length(model2$coefficients))
wald.test(b = coef(model3), Sigma = vcov(model3), Terms =
2:length(model3$coefficients))
wald.test(b = coef(model4), Sigma = vcov(model4), Terms =
2:length(model4$coefficients))
wald.test(b = coef(model5), Sigma = vcov(model5), Terms =
2:length(model5$coefficients))
wald.test(b = coef(model6), Sigma = vcov(model6), Terms =
2:length(model6$coefficients))
wald.test(b = coef(model7), Sigma = vcov(model7), Terms =
2:length(model7$coefficients))

      #### PRE ###
mode_of_model <- function(model_name) {
  outcome<-model_name$y
  categories <- unique(outcome)
  categories[which.max(tabulate(match(outcome, categories)))]
}

pre(model1)
pre(model2)
pre(model3)
pre(model4)
pre(model5)
pre(model6)
pre(model7)

    #### Stargazer ####
stargazer.pre<-function(model){
  temp1<<-pre(model)
  temp2<<-format(round(temp1, 3), nsmall=3)
  out<<-paste(temp2)
  return(out)
}

stargazer(model3, model4, model5, model6, model7, type = "latex", keep.stat
= c("n", "ll", "aic"),
          add.lines=list(c("Wald $\\chi^{2}$",
  stargazer.wald.output(model3),
  stargazer.wald.output(model4),
  stargazer.wald.output(model5),
  stargazer.wald.output(model6),
  stargazer.wald.output(model5))),
c("P.R.E.",
  stargazer.pre(model3),
  stargazer.pre(model4),
  stargazer.pre(model5),
  stargazer.pre(model6),
  stargazer.pre(model7)),
table.layout = "mc-b-t-sa-n")
```