Exam Template

Nick Thompson

10/10/2016

Pull in Data

```
setwd("~/Documents/GitHubRepo/729_Reed_MLE_git/Exam")
data <- read.csv(file = "https://raw.githubusercontent.com/Neilblund/729A/master/data/voterid.csv", head
#or
# data <- read.dta(file = "hyperlink or filename goes here")
#or
# data <- read.dta13(file = "hyperlink or filename goes here")
save(data, file = "data.RData")
load("data.RData")
#View(data)
```

Introduction:

```
#descriptive statistics for all variables
#stargazer(voterid, type = 'text')
# run probit, show results
data <- na.omit(data)
#View(data) - here we put the point prediction that we're looking for.
# If you're using mean, then keep mean.
# If you're using a SD up and down from the mean, then use that.
# If you're using a different range, then use that.
data$mean_gopleg <- mean(data$gopleg)
data$sd_gopleg <- 0.5*sd(data$gopleg)
data$med_g = median(data$gopleg)
```

Descriptive statistics

```
stargazer(data, type = 'text', header = F)
```

== Statistic N Mean St. Dev. Min Max
———————————————— photo 48 0.375 0.489 0 1
fraud 48 1.066 1.817 0.000 7.833 election_margin 48 15.400 10.827 4.911 49.431 gopleg 48 49.088 14.693 13.900 79.991 mean_gopleg 48 49.088 0.000 49.088 49.088 sd_gopleg 48 7.347 0.000 7.347 7.347 med_g 48 49.502 0.000 49.502 49.502 ————————————————

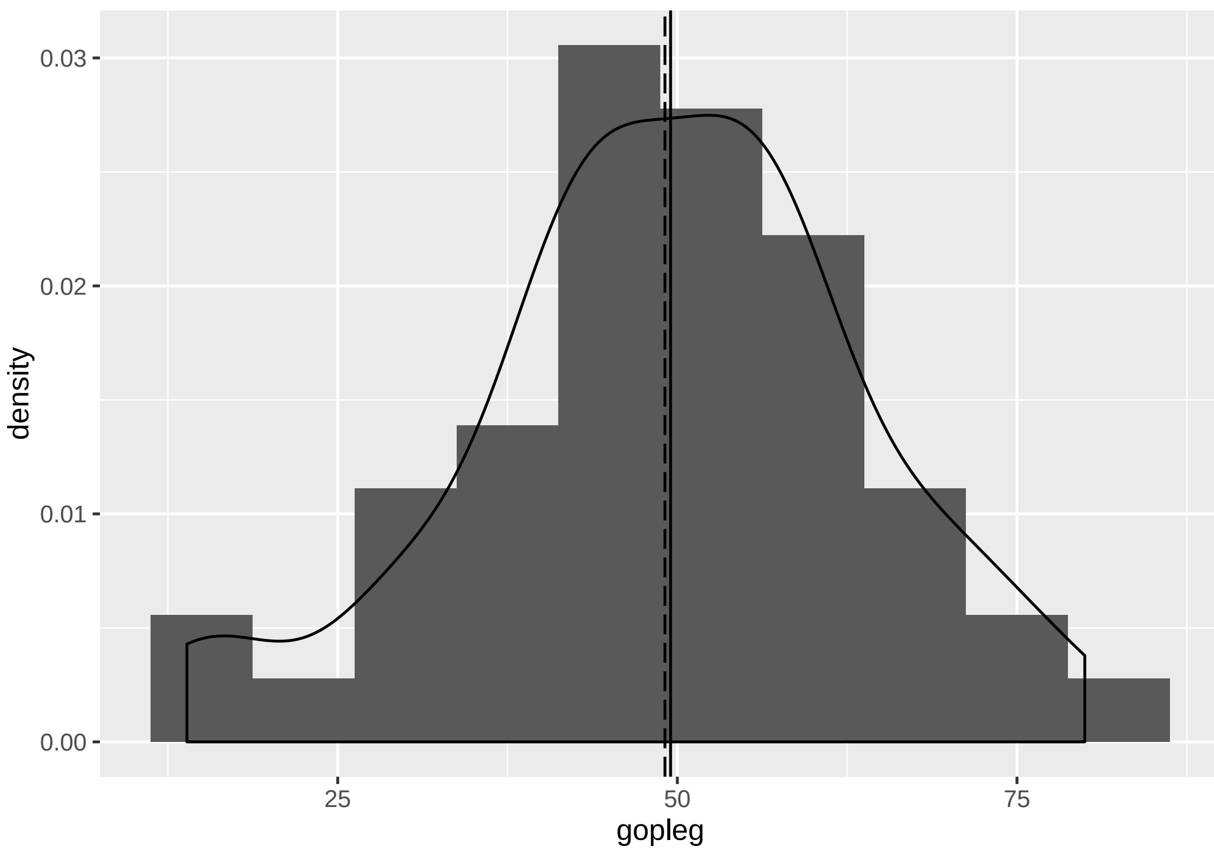
Plot a histogram to see what the data looks like. Identify skewness for determining if using mean, median, or tail.

```
g <- ggplot(data,aes(x=gopleg))
# adjust the binwidth to the desired width so the histogram tells you something.
# the geom_vlines give you lines to mark the mean and median.
# Consider using the median if the skewness > 0.5 or skewness < -0.5, but use your
```

```r
# own discretion.  If the -0.5 < skewness < 0.5 then use the mean.
skewness(data$gopleg)
```

```
## [1] -0.3384022
```

```r
# kurtosis(data$gopleg) # if you care about kurtosis
# anova(model_1p) # if for some reason you want ANOVA
g + geom_histogram(aes(y=..density..),binwidth = 7.5) +
  geom_density() +
  geom_vline(xintercept = data$mean_gopleg,linetype='longdash') +
  geom_vline(xintercept = data$med_g)
```




```r
gopleg_obs_low <- data$mean_gopleg - data$sd_gopleg
gopleg_obs_high <- data$mean_gopleg + data$sd_gopleg
```

Calculate the average effect of `variable name` using observed values.

The logit model.

```r
# run probit, show results
(model_1l <- glm('photo ~ fraud + election_margin + gopleg',
               family = binomial(link = "logit"),
               data = data))
```

```
##
```

```
## Call:  glm(formula = "photo ~ fraud + election_margin + gopleg", family = binomial(link = "logit"),
##     data = data)
##
## Coefficients:
##     (Intercept)            fraud  election_margin           gopleg
##        -4.07692          0.23870         -0.01288          0.06828
##
## Degrees of Freedom: 47 Total (i.e. Null);  44 Residual
## Null Deviance:       63.51
## Residual Deviance: 54.34     AIC: 62.34
```

A table of the logit model.

```
#summary(model_1l)
stargazer(model_1l,header=F) # if you want to see a print out in your console,
```

Table 1:

	Dependent variable:
	NA
fraud	0.239
	(0.190)
election_margin	−0.013
	(0.041)
gopleg	0.068**
	(0.032)
Constant	−4.077***
	(1.572)
Observations	48
Log Likelihood	−27.169
Akaike Inf. Crit.	62.339
Note:	*p<0.1; **p<0.05; ***p<0.01

```
                        # then after model_1l typt this ,type='text',
```

The logit predicted probabilities.

```
# generate predicted probabilities automatically
data$pprob_l <- predict(model_1l,type="response")

# generate predicted probabilities manually
data$pprob_manual_l <- pnorm(model_1l$coef['(Intercept)'] +
                                 model_1l$coef['fraud']*data$fraud +
                                 model_1l$coef['election_margin']*data$election_margin +
                                 model_1l$coef['gopleg']*data$gopleg)
```

```r
# test that we did it right
data$pprob_test_l <- data$pprob_l - data$pprob_manual_l
```

```r
summary(data$pprob_test_l) # should be zeros
```

```
Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
```

-0.11210 -0.02176 0.05983 0.03876 0.09996 0.11730

Calculate the average effects with a logit model.

```r
#######1. calculate average effect of photo using observed values----
data$pprob_gopleg_upsd_l <- pnorm(model_1l$coef['(Intercept)'] +
                                 model_1l$coef['fraud']*data$fraud +
                                 model_1l$coef['election_margin']*data$election_margin +
                                 model_1l$coef['gopleg']*gopleg_obs_high)
summary(data$pprob_gopleg_upsd_l)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1947  0.3432  0.3969  0.4302  0.4655  0.9078
```

```r
data$pprob_gopleg_downsd_l <- pnorm(model_1l$coef['(Intercept)'] +
                                 model_1l$coef['fraud']*data$fraud +
                                 model_1l$coef['election_margin']*data$election_margin +
                                 model_1l$coef['gopleg']*gopleg_obs_low)
summary(data$pprob_gopleg_downsd_l)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03118 0.07975 0.10300 0.14200 0.13790 0.62710
```

```r
data$pprob_effect_l <- data$pprob_gopleg_upsd_l - data$pprob_gopleg_downsd_l
summary(data$pprob_effect_l)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1636  0.2635  0.2875  0.2882  0.3148  0.3837
```

A way to do this with probit.

```r
# run probit, show results
(model_1p <- glm('photo ~ fraud + election_margin + gopleg',
                family = binomial(link = "probit"),
                data = data))
```

```
##
## Call:  glm(formula = "photo ~ fraud + election_margin + gopleg", family = binomial(link = "probit"),
##     data = data)
##
## Coefficients:
##     (Intercept)            fraud  election_margin           gopleg
##       -2.252268         0.143559        -0.001425         0.035747
```

```
##
## Degrees of Freedom: 47 Total (i.e. Null);  44 Residual
## Null Deviance:       63.51
## Residual Deviance: 54.75     AIC: 62.75
```

A table printout of the probit model.

```
#summary(model_1p)
stargazer(model_1p,type = 'text',header=F)
```

=== Dependent variable:
——————————— NA
——————————————— fraud 0.144
(0.112)

election_margin -0.001
(0.022)

gopleg 0.036**
(0.016)

Constant -2.252***
(0.828)

Observations 48
Log Likelihood -27.374
Akaike Inf. Crit. 62.748
=== Note: *p<0.1;* ***p<0.05;***
p<0.01

Calculate the predicted probabilities in the probit model.

```
# generate predicted probabilities automatically
data$pprob_p <- predict(model_1p,type="response")

# generate predicted probabilities manually
data$pprob_manual_p <- pnorm(model_1p$coef['(Intercept)'] +
                                 model_1p$coef['fraud']*data$fraud +
                                 model_1p$coef['election_margin']*data$election_margin +
                                 model_1p$coef['gopleg']*data$gopleg)

# test that we did it right
data$pprob_test_p <- data$pprob_p - data$pprob_manual_p
```

A summary of the probit test.

```
summary(data$pprob_test_p) # should be zeros
```

Min. 1st Qu. Median Mean 3rd Qu. Max. 0 0 0 0 0 0

Calculate the average effects and difference for a probit model.

```
#######1. calculate average effect of photo using observed values----
data$pprob_gopleg_upsd_p <- pnorm(model_1p$coef['(Intercept)'] +
                                  model_1p$coef['fraud']*data$fraud +
                                  model_1p$coef['election_margin']*data$election_margin +
                                  model_1p$coef['gopleg']*gopleg_obs_high)
summary(data$pprob_gopleg_upsd_p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3801  0.4035  0.4194  0.4574  0.4653  0.8036
```

```
data$pprob_gopleg_downsd_p <- pnorm(model_1p$coef['(Intercept)'] +
                                  model_1p$coef['fraud']*data$fraud +
                                  model_1p$coef['election_margin']*data$election_margin +
                                  model_1p$coef['gopleg']*gopleg_obs_low)
summary(data$pprob_gopleg_downsd_p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2031  0.2208  0.2331  0.2695  0.2702  0.6290
```

```
data$pprob_effect_p <- data$pprob_gopleg_upsd_p - data$pprob_gopleg_downsd_p
```

```
summary(data$pprob_effect_p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1746  0.1825  0.1855  0.1879  0.1921  0.2071
```

Summary: Interpreting the Coefficients, include the AIC

```
stargazer(model_1p, header = F)
```

Log Likelihood

```
# estimate simple logit model
m <- glm(photo ~ fraud + election_margin + gopleg,
               family = binomial,
               data = data)
m0<-glm(photo~1, family=binomial, data = data)
#Likelihood Ratio Tests
logLik(m0)
```

```
## 'log Lik.' -31.75504 (df=1)
```

```
logLik(m)
```

```
## 'log Lik.' -27.16938 (df=4)
```

```
lr.test = -2*(logLik(m0) - logLik(m))
lr.test
```

```
## 'log Lik.' 9.171311 (df=1)
```

Table 2:

	Dependent variable:
	NA
fraud	0.144
	(0.112)
election_margin	−0.001
	(0.022)
gopleg	0.036**
	(0.016)
Constant	−2.252***
	(0.828)
Observations	48
Log Likelihood	−27.374
Akaike Inf. Crit.	62.748
Note:	*p<0.1; **p<0.05; ***p<0.01

```r
lrtest(m0,m)
```

```
## Likelihood ratio test
##
## Model 1: photo ~ 1
## Model 2: photo ~ fraud + election_margin + gopleg
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   1 -31.755
## 2   4 -27.169  3 9.1713     0.0271 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Deviances

```r
nd<-2*(0 - logLik(m0))
nd
```

```
## 'log Lik.' 63.51007 (df=1)
```

```r
rd<-2*(0-logLik(m))
rd
```

```
## 'log Lik.' 54.33876 (df=4)
```

```r
nd-rd
```

```
## 'log Lik.' 9.171311 (df=1)
```

Bayes

```
#Bayesian Information Criterion
BIC<- -2*logLik(m)+log(data$photo)*(length(coef(m))+1)
BIC
```

```
##  [1] 54.33876     -Inf     -Inf     -Inf     -Inf 54.33876     -Inf
##  [8]     -Inf 54.33876 54.33876 54.33876 54.33876     -Inf 54.33876
## [15]     -Inf 54.33876     -Inf     -Inf     -Inf     -Inf 54.33876
## [22]     -Inf     -Inf     -Inf     -Inf     -Inf 54.33876     -Inf
## [29]     -Inf     -Inf     -Inf 54.33876     -Inf 54.33876     -Inf
## [36] 54.33876     -Inf 54.33876 54.33876 54.33876 54.33876     -Inf
## [43]     -Inf     -Inf     -Inf     -Inf 54.33876     -Inf
```

AIC

```
#Akaike's Information Criterion
length(coef(m))
```

```
## [1] 4
```

```
AIC<- -2*(logLik(m))+2*(length(coef(m)))
AIC
```

```
## 'log Lik.' 62.33876 (df=4)
```

Simulations:

```
# set the medians
x.frau <- median(data$fraud)  # in years
x.ele <- median(data$election_margin)  # in years
x.gopleg <- median(data$gopleg)  # in 10,000s of dollars
```

```
# set beta.hat, Sigma.hat, and X.c
beta.hat <- coef(m)
Sigma.hat <- vcov(m)
X.c <- c(1, x.frau, x.ele, x.gopleg)
```

```
# simulate p.tilde
n.sims <- 1000  # set number of simulations
p.tilde <- numeric(n.sims)  # create holder for simulations
for (i in 1:n.sims) {
  beta.tilde <- mvrnorm(1, beta.hat, Sigma.hat)
  p.tilde[i] <- plogis(X.c%*%beta.tilde)
}
```

```
# characterize the simulations
mean(p.tilde)
```

```
## [1] 0.3195129
```

```r
sd(p.tilde)
```

```
## [1] 0.07880783
```

```r
quantile(p.tilde, c(0.05, 0.95))
```

```
##        5%       95%
## 0.1982026 0.4566960
```

```r
mod <- obsval('photo~fraud+election_margin+gopleg',data=data,
              reg.model = "logit",
              n.draws = 1000,
              effect.var = "gopleg",
              effect.vals = c(0,2), # lowest to mid
              verbose = TRUE)
```

```
## Dependent variable is numeric and binary.
## Estimating model...
## Done estimating model.
## Drawing simulated coefficients from posterior distribution...
## Finished drawing simulated coefficients from posterior distribution...
## Now in obsvalPredict() ...
## Constructing X.matrix ... Generating control predictions ...
## Entered computePreds()...
## Generating predictions for each set of simulated coefficients ...
## Calculated predictions for each set of simulated coefficients.
```

```r
# display model results
summary(mod$model)
```

```
##
## Call:
## glm(formula = fmla, family = binomial(link = "logit"), data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4100  -0.9129  -0.5987   1.0046   2.5114
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -4.07692    1.57239  -2.593  0.00952 **
## fraud            0.23870    0.19032   1.254  0.20978
## election_margin -0.01288    0.04087  -0.315  0.75259
## gopleg           0.06828    0.03224   2.118  0.03421 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 63.510  on 47  degrees of freedom
## Residual deviance: 54.339  on 44  degrees of freedom
```

```
## AIC: 62.339
##
## Number of Fisher Scoring iterations: 5
```

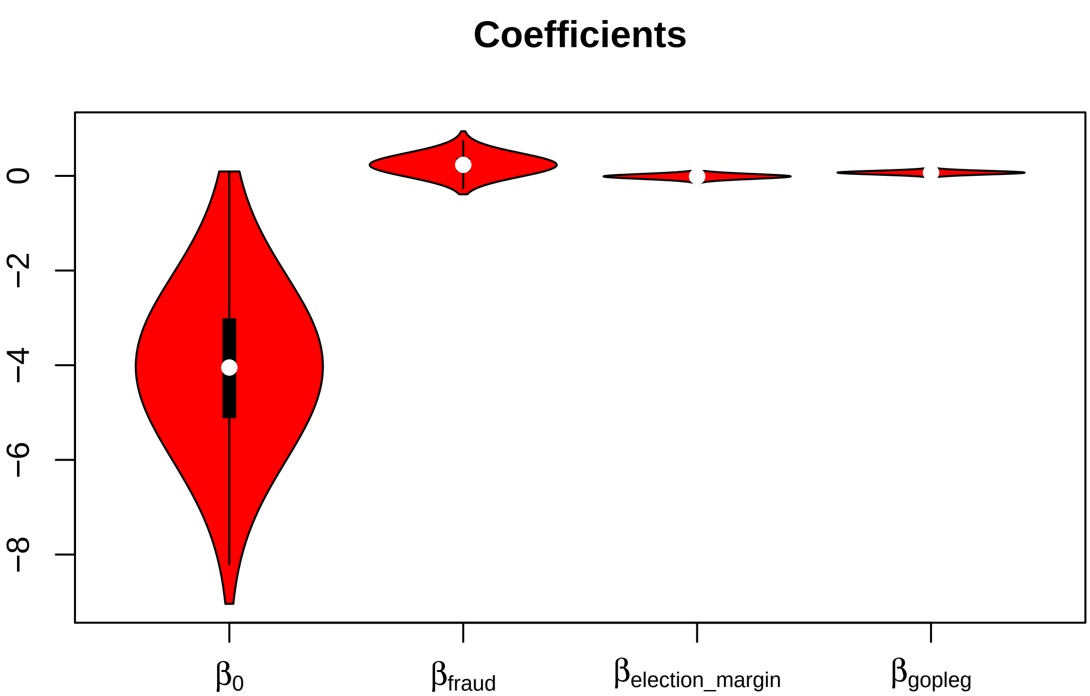
```r
summary(mod$preds)
```

```
##      var_0               var_2
##  Min.   :0.0002953   Min.   :0.0003901
##  1st Qu.:0.0078917   1st Qu.:0.0093692
##  Median :0.0246210   Median :0.0279320
##  Mean   :0.0592378   Mean   :0.0624640
##  3rd Qu.:0.0598677   3rd Qu.:0.0654533
##  Max.   :0.6822379   Max.   :0.6750480
```

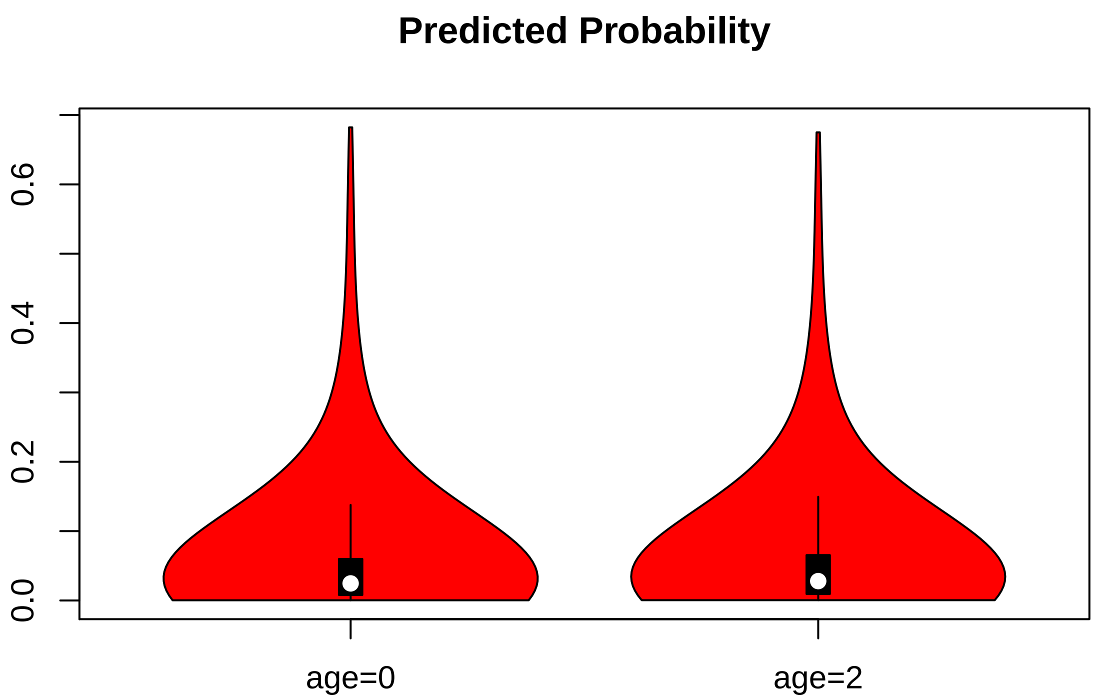
```r
# see the names of everything obsval returns
# names(mod)
```

Violin Plots of the Simulation

```r
vioplot(mod$sim.coef[,1], mod$sim.coef[,2], mod$sim.coef[,3],
        mod$sim.coefs[,4], names=c(expression(beta[0]),expression(beta[fraud]),
                        expression(beta[election_margin]),expression(beta[gopleg])),
        col="red")
title("Coefficients")
```




```r
vioplot(mod$preds[,1], mod$preds[,2], names=c("age=0","age=2"),
        col="red")
title("Predicted Probability")
```




```r
mean(mod$preds[, 1] - mod$preds[, 2])
```

```
## [1] -0.003226183
```

```r
# calculate difference between quantiles
quantile(mod$preds[, 1] - mod$preds[, 2], c(0.025, 0.975))
```

```
##          2.5%         97.5%
## -0.0080405888 -0.0001426362
```