Homework 3 - GVPT 729A

Nick Thompson

9/29/2016

Answer the following questions. Include your code, and report all the results you used to answer the questions. https://raw.githubusercontent.com/Neilblund/729A/master/data/voterid.csv

```
setwd("~/Documents/GitHubRepo/729_Reed_MLE_git/Assignments")
#voterid <- read.csv(file = "voterid.csv",header = TRUE, sep = ",")
#save(voterid, file = "voterid.RData")
load("voterid.RData")
#View(voterid)
```

The link above contains data from "Hicks et al. 2015: A Principle or a Strategy? Voter Identificaqtion Laws and Partisan Competition in the American States"

- `photo` is equatl to 1 if a state has legislation that requires voters to show photo ID at the polling booth, and 0 if they do not have to have this requirement.
- `fraud` is the average number of voter fraud cases prosecuted in a given state since 2001.
- `election_margin` is the average partisan vote margin (% Republican - % Democratic) in a state since 2001.
- `gopleg` is the average % of a state's legislature that is Republican.

Questions

---

1. Use OLS to estimate a linear probability model using `photo` as the dependent variable, and `fraud`, `election_margin`, and `gopleg` as independent variables. Obtain predicted probabilities that `photo` = 1 under two or more hypothetical scenarios. Discuss your results.

```
#sink("log.txt", append = TRUE, type = 'output', split=TRUE)
stargazer(voterid, type = 'text', title = "Descriptive Statistics",
          digits = 5, min.max = F, na.rm = T)
```

```
##
## Descriptive Statistics
## ====================================
## Statistic       N    Mean   St. Dev.
## ------------------------------------
## photo           50 0.38000  0.49031
## fraud           50 1.02500  1.79065
## election_margin 48 15.40035 10.82744
## gopleg          49 48.93150 14.58026
## ------------------------------------
##
```

```
## Descriptive Statistics
## ====
## TRUE
## ----
```

```r
attach(voterid)
mean_p <- mean(photo, na.rm = T)
mean_f <- mean(fraud, na.rm = T)
mean_e <- mean(election_margin, na.rm = T)
mean_g <- mean(gopleg, na.rm = T)
sd_f <- sd(fraud, na.rm = T)
sd_g <- sd(gopleg, na.rm = T)
model.1 <- lm(photo~fraud+election_margin+gopleg, data = voterid); summary(model.1)
```

```
##
## Call:
## lm(formula = photo ~ fraud + election_margin + gopleg, data = voterid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6858 -0.3431 -0.2089  0.4309  1.0120
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.269638   0.254084  -1.061   0.2944
## fraud            0.047491   0.037379   1.271   0.2106
## election_margin  0.001551   0.006216   0.250   0.8041
## gopleg           0.011614   0.004616   2.516   0.0156 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4608 on 44 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.1695, Adjusted R-squared:  0.1128
## F-statistic: 2.993 on 3 and 44 DF,  p-value: 0.04091
```

```r
predict_ols <- function(x1, x2, x3){
  y <- (model.1$coefficients[1] + x1 * model.1$coefficients[2] +
    x2 * model.1$coefficients[3] + x3 * model.1$coefficients[4])
return(as.numeric(y))
}
# Point Predictions for 'fraud' and `gopleg` using the `lm()` function
predict_fraud1 <- predict_ols((mean_f + (sd_f)), mean_e, mean_g); predict_fraud1
```

```
## [1] 0.4562795
```

```r
predict_gopleg1 <- predict_ols(mean_f, mean_e, (mean_g + (sd_g))); predict_gopleg1
```

```
## [1] 0.54058
```

```r
#sink()
```

I calculated the point predictions for `fraud` one standard deviation above its mean. Holding the other variables constant at their means, a one standard deviation increase in a state having a voter fraud case brought before the court inidicates 45.63% point increase in the likelihood of of the state having a photo ID law. This indicates that states with photo ID requirements are 45.63% more likely to have fraud cases identified and tried than states without photo ID laws.

For my second point prediction I calculated one standard deviation above the mean of `gopleg`. Keeping the other variables at their means, on average a one standard deviation increase in the likelihood of a state having a GOP legislature indicates a 54.06% increase in the likelihood of a state having a photo voter ID law.

---

2. Run a logistic regression model using `photo` as the dependent variable, and `fraud`, `election_margin`, and `gopleg` as independent variables. Obtain predicted probabilities that `photo` = 1 under two or more hypothetical scenarios. Discuss your results, and compare with your results from model 1.

```r
model.2 <- glm(photo~fraud+election_margin+gopleg, data = voterid); summary(model.2)
```

```
##
## Call:
## glm(formula = photo ~ fraud + election_margin + gopleg, data = voterid)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6858  -0.3431  -0.2089   0.4309   1.0120
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.269638   0.254084  -1.061   0.2944
## fraud            0.047491   0.037379   1.271   0.2106
## election_margin  0.001551   0.006216   0.250   0.8041
## gopleg           0.011614   0.004616   2.516   0.0156 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2123505)
##
##     Null deviance: 11.2500  on 47  degrees of freedom
## Residual deviance:  9.3434  on 44  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 67.665
##
## Number of Fisher Scoring iterations: 2
```

```r
predict_glm <- function(x1, x2, x3){
  inverse_link <- -1*(model.2$coefficients[1] + x1 * model.2$coefficients[2] +
    x2 * model.2$coefficients[3] + x3 * model.2$coefficients[4])
  y <- 1/(1+exp(inverse_link))
```

```r
return(as.numeric(y))
}

# Point predictions for `fraud` and `gopleg` using the `glm()` function.
predict_fraud3 <- predict_glm((mean_f + sd_f), mean_e, mean_g); predict_fraud3
```

```
## [1] 0.6121312
```

```r
predict_gopleg3 <- predict_glm(mean_f, mean_e, (mean_g + sd_g)); predict_gopleg3
```

```
## [1] 0.6319473
```

```r
table_1 <- data.frame(predict_fraud1,predict_fraud3,predict_gopleg1,predict_gopleg3)
stargazer(table_1, type = "text", nobs = FALSE, min.max = FALSE,
          title = "Table 1 - Comparative Statics")
```

```
##
## Table 1 - Comparative Statics
## ==============================
## Statistic       Mean  St. Dev.
## ------------------------------
## predict_fraud1  0.456
## predict_fraud3  0.612
## predict_gopleg1 0.541
## predict_gopleg3 0.632
## ------------------------------
```

```r
difference_fraud <- (predict_fraud3-predict_fraud1); difference_fraud
```

```
## [1] 0.1558517
```

```r
difference_gopleg <- (predict_gopleg3-predict_gopleg1); difference_gopleg
```

```
## [1] 0.09136737
```

I calculated the same hypothetical situations for the logistic model that I did for the linear model so that comparison was more simple. Table 1 shows the four predictions side by side. The predictions for the `fraud` variables and the `gopleg` variables show stark differences. Holding the other vairables at their means, the average logistic model for `fraud` was 15.58% points higher than the linear model. Likewise, when holding the other variables constant at their means, the average value for `gopleg` under the logistic model was 9.14% percentage points higher than the linear model.

notes

1. There's no "right" or "wrong" scenario here, but you should think about comparing scenarios that are plausible reflections of the real world.

2. We used the code below to get predicted probabilities from the probit model. You will need to modify this slightly to get predictions form a logit, just remember that: $Pr(Y = 1|X1, X2) = \frac{1}{1+e^{-(B_0+B_1X1+B_2X2)}}$

```r
set.seed(42)
n <- 1000
x1 <- rnorm(n)
x2 <- rnorm(n)
X <- cbind(1, x1, x2)
b <- c(1, -0.5, 0.5)
p <- pnorm(X%*%b)
y <- rbinom(n, 1, p)
model <- glm(y ~ x1 + x2, family = "binomial"(link = "probit"))

# with x1 and x2 set at their mean
meanx1 <- mean(x1)
meanx2 <- mean(x2)
pnorm(model$coefficients[1] +
        meanx1*model$coefficients[2] +
        meanx2*model$coefficients[3])
```

```
## (Intercept)
##   0.8344602
```

```r
q.x1 <- quantile(x1, 0.25)

pnorm(model$coefficients[1] +
        q.x1*model$coefficients[2] +
        meanx2*model$coefficients[3])
```

```
## (Intercept)
##   0.9091454
```