

POL SCI 231b (Spring 2017):

Problem Set 7

Spring 2017

University of California, Berkeley

Prof. Thad Dunning/GSI Natalia Garbiras-Díaz

Due Monday, April 3rd, at 9:00 AM (before lecture)

Each group should turn in its problem set solution by email to Natalia Garbiras-Díaz (nataliagarbirasdiaz+231b@berkeley.edu) by Monday at 9 AM. Please work out the problems on your own, before you meet with your group to agree on solutions.

1. A researcher runs a clustered randomized experiment with 297 units divided into 15 clusters of unequal size. The code below generates potential outcomes that are associated with cluster membership (note that average potential outcomes differ for each cluster). It then simulates one realization of the experiment. The data created by this code are in `clustered.experiment.Rda` on bcourses.

```
set.seed(94705)
cluster <- c(rep(1:4, each = 5), rep(5:8,
  each = 18), rep(9:12, each = 25),
  rep(13:15, each = 35))
y0 <- rnorm(length(cluster), cluster,
  1)
y1 <- y0 + rnorm(length(cluster), sqrt(cluster),
  0.1)
treat_cluster <- sample(1:15, 8)
treat <- as.numeric(cluster %in% treat_cluster)
```

```
yobs <- ifelse(treat == 1, y1, y0)
data <- as.data.frame(cbind(y1, y0,
  cluster, treat, yobs))
```

- (a) What is the average causal effect $\frac{1}{297}[\sum_{i=1}^{297} Y_i(1) - \sum_{i=1}^{297} Y_i(0)]$?

```
ACE <- mean(data$y1) - mean(data$y0)
ACE
## [1] 3.134169
```

- (b) Use data from the realization of the experiment to estimate the average causal effect, using the difference of cluster averages.

Thus, following Lecture 7, let the average outcome in the $A = 8$ clusters assigned to treatment be $Y^T = \frac{1}{A} \sum_{k \in A} \bar{Y}_k(1)$, where $\bar{Y}_k(1)$ is the average in the k th cluster, and let the average outcome in the $B = 7$ clusters assigned to control be $Y^C = \frac{1}{B} \sum_{k \in B} \bar{Y}_k(0)$. The difference of the cluster averages is $\widehat{ATE}_{cluster} = Y^T - Y^C$.

It may be helpful to use the `ddply` command in the `plyr` package to calculate cluster means.

```
library(plyr)
cl <- ddply(data, ~cluster, summarise,
  mean_yobs = mean(yobs), treat = mean(treat))
cl_hatATE <- mean(cl$mean_yobs[cl$treat ==
  1]) - mean(cl$mean_yobs[cl$treat ==
  0])
cl_hatATE
## [1] 3.844448
```

- (c) Calculate the standard error of $\widehat{ATE}_{cluster}$, using the approach in Lecture 7. That is, use the conservative Neyman estimator of the variance, where the elements of the formula are the cluster averages defined in part (b): $\text{Var}(Y^T - Y^C) = \text{Var}(Y^T) + \text{Var}(Y^C)$, where Var refers to the variance of the sampling distribution of the estimators.

Now, use the ratio of $\widehat{ATE}_{cluster}$ and the estimated standard error to conduct a t -test and calculate a p -value, that is, the probability that you would observe the data from the realized experiment under the null hypothesis that $ATE = 0$.

```
var1 <- var(cl$mean_yobs[cl$treat ==
  1])/8
var0 <- var(cl$mean_yobs[cl$treat ==
```

```

    0))/7
se_cl_hatATE <- sqrt(var1 + var0)
se_cl_hatATE

## [1] 2.595216

t_ratio <- abs(cl_hatATE/se_cl_hatATE)
t_ratio

## [1] 1.481359

# Df
df.num <- ((var1 + var0)^2)
df.den <- (var1^2)/(8 - 1) + (var0^2)/(7 -
    1)
df <- df.num/df.den

# P-value
p <- 2 * pt(t_ratio, df, lower.tail = F)
p

## [1] 0.1633637

# or else just
with(cl, t.test(mean_yobs ~ treat))

##
## Welch Two Sample t-test
##
## data: mean_yobs by treat
## t = -1.4814, df = 12.454,
## p-value = 0.1634
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -9.476159 1.787263
## sample estimates:
## mean in group 0 mean in group 1
## 7.440899 11.285347

```

- (d) Suppose you erroneously assumed that the data were produced by individual rather than cluster randomization. What standard error, t ratio, and p value would you calculate?

```

var1 <- var(data$yobs[data$treat ==
    1])/sum(data$treat == 1)
var0 <- var(data$yobs[data$treat ==
    0])/sum(data$treat == 0)

```

```

se_ind_hatATE <- sqrt(var1 + var0)
se_ind_hatATE

## [1] 0.4751558

hatATE_ind <- mean(data$yobs[data$treat ==
  1]) - mean(data$yobs[data$treat ==
  0])
t_ratio_ind <- abs(hatATE_ind/se_ind_hatATE)
t_ratio_ind

## [1] 7.004645

# Df
df.num <- ((var1 + var0)^2)
df.den <- (var1^2)/(sum(data$treat ==
  1) - 1) + (var0^2)/(sum(data$treat ==
  0) - 1)
df <- df.num/df.den
df

## [1] 292.913

# P-value
p <- 2 * pt(t_ratio_ind, df, lower.tail = F)
p

## [1] 1.701454e-11

# or else just
with(data, t.test(yobs ~ treat))

##
## Welch Two Sample t-test
##
## data: yobs by treat
## t = -7.0046, df = 292.91,
## p-value = 1.701e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -4.263450 -2.393146
## sample estimates:
## mean in group 0 mean in group 1
## 10.12706 13.45536

```

- (e) Now, use a simulation to create 10,000 realizations of the experiment and construct a plot of the sampling distribution of $\widehat{ATE}_{cluster}$. In addi-

tion,

- For each of the replicates, calculate the nominal standard error, t -ratio, and p -value as in part (c). Also calculate the nominal standard error, t -ratio, and p -value that you would calculate if you erroneously assumed that the data were produced by individual rather than cluster randomization, as in part (d). Compare the standard deviation of the 10,000 $\widehat{ATE}_{cluster}$ to the averages of the nominal standard errors constructed as per parts (c) and (d). What do you conclude?
 - Compare the mean of the bootstrap $\widehat{ATE}_{cluster}$ to the ATE from (a). What do you conclude?
- (f) Double the size of the experiment by creating a new dataset that just combines two copies of the original data (i.e., `data2 <- rbind(data, data)`). Repeat the exercise in (c) with this new dataset.
- (g) Repeat (d), now using a data set duplicating the dataset with 30 clusters.

```
options(scipen = 999)

### t-test function
ttest <- function(y, x, two.tailed = TRUE) {

  # Calculating difference in means
  mean1 <- mean(y[x == 1], na.rm = T)
  mean0 <- mean(y[x == 0], na.rm = T)
  diff <- mean1 - mean0

  # Calculating SE of the difference
  N1 <- length(na.omit(y[x == 1]))
  N0 <- length(na.omit(y[x == 0]))
  var1 <- var(y[x == 1], na.rm = T)
  var0 <- var(y[x == 0], na.rm = T)
  varN1 <- var1/N1
  varN0 <- var0/N0
  se.diff <- sqrt(varN1 + varN0)

  # T-statistic
  t <- diff/se.diff

  # Degrees of freedom
  df.num <- ((varN1 + varN0)^2)
  df.den <- (varN1^2)/(N1 - 1) + (varN0^2)/(N0 -
    1)
  df <- df.num/df.den
```

```

# P-value
if (two.tailed == TRUE) {
  p <- 2 * pt(abs(t), df, lower.tail = F)
}

if (two.tailed == FALSE) {
  p <- pt(t, df, lower.tail = F)
}

# Preparing output
res <- c(diff, se.diff, t, p)
names(res) <- c("Difference", "SE_Diff",
               "t-stat", "p-value")

return(c(res))
}

boot_dist <- function(sdata = data) {

  ### 1. Individual randomization sim
  ### treatment and observed y
  sdata$indtreat <- sample(sdata$treat,
                          length(sdata$treat), replace = F)
  sdata$yobs_ind <- ifelse(sdata$indtreat ==
                          1, y1, y0)
  # int t-test
  ind <- ttest(sdata$yobs_ind, sdata$indtreat)

  # 2. Cluster randomization setup
  nclusters <- length(unique(sdata$cluster))
  nclusterstreat <- length(unique(sdata$cluster[sdata$treat ==
  1]))
  nclusterscontrol <- nclusters -
    nclusterstreat
  # sim treat and observed y
  sdata$cltreat <- as.numeric(sdata$cluster %in%
    sample(1:nclusters, nclusterstreat))
  sdata$yobs_cl <- ifelse(sdata$cltreat ==
    1, sdata$y1, sdata$y0)
  cl <- ddply(sdata, ~cluster, summarise,
    mean_yobscl = mean(yobs_cl),
    cltreat = mean(cltreat))
  # cluster means t-test
  clus <- ttest(cl$mean_yobscl, cl$cltreat)

```

```

# RETURN
results <- c(ind, clus)
names(results) <- c("IND_hatATE",
  "IND_SE_hatATE", "IND_t", "IND_p",
  "CL_hatATE", "CL_SE_hatATE",
  "CL_t", "CL_p")
return(results)
}

# SIMULATION 1
sim1 <- replicate(10000, boot_dist(sdata = data))
# SIMULATION 2
data2 <- data
treat_cluster <- sample(1:15, 8)
data2$treat <- as.numeric(data2$cluster %in%
  treat_cluster)
data2$yobs <- ifelse(treat == 1, data2$y1,
  data2$y0)
data2$cluster <- data2$cluster + 15
data2 <- rbind(data, data2)
sim2 <- replicate(10000, boot_dist(sdata = data2))
# SIMULATION 3
data3 <- data2
treat_cluster <- sample(1:30, 16)
data3$treat <- as.numeric(data3$cluster %in%
  treat_cluster)
data3$yobs <- ifelse(treat == 1, data3$y1,
  data3$y0)
data3$cluster <- data3$cluster + 30
data3 <- rbind(data2, data3)
sim3 <- replicate(10000, boot_dist(sdata = data3))

round(cbind(apply(sim1, 1, mean), apply(sim2,
  1, mean), apply(sim3, 1, mean)),
  3)

##           [,1]  [,2]  [,3]
## IND_hatATE   3.139 3.142  3.132
## IND_SE_hatATE 0.478 0.338  0.239
## IND_t        6.570 9.295 13.100
## IND_p        0.000 0.000  0.000
## CL_hatATE    2.710 2.705  2.674
## CL_SE_hatATE 2.535 1.762  1.236
## CL_t         1.081 1.539  2.164

```

```
## CL_p          0.374 0.253  0.120
```

First, we compare the standard deviation of the 10,000 ATE to the averages of the nominal cluster standard errors constructed as per parts (c) and (d).

```
# For each simulation, calculate the
# standard deviation of the 10,000
# ATE
SE_sim <- c(sd(sim1[5, ]), sd(sim2[5,
  ]), sd(sim3[5, ]))

# Now get the averages of the
# nominal cluster standard errors
nominal <- cbind(apply(sim1, 1, mean),
  apply(sim2, 1, mean), apply(sim3,
    1, mean))[c(2, 6), ]

# Put everything together
round(rbind(SE_sim, nominal), 3)

##           [,1]  [,2]  [,3]
## SE_sim      2.511 1.773 1.212
## IND_SE_hatATE 0.478 0.338 0.239
## CL_SE_hatATE  2.535 1.762 1.236
```

The comparison shows the bias in the nominal standard errors assuming individual level randomization. It also suggests that the nominal standard errors assuming cluster level randomization are an unbiased estimator of the true standard error of the sampling distribution of $\widehat{ATE}_{cluster}$.

Now we can compare the mean of the bootstrap ATE to the ATE from (a).

```
# ATE from (a)
ACE

## [1] 3.134169

# mean of the bootstrap ATE
cbind(apply(sim1, 1, mean), apply(sim2,
  1, mean), apply(sim3, 1, mean))[5,
  ]

## [1] 2.709846 2.704778 2.673502

# bias
ACE - cbind(apply(sim1, 1, mean), apply(sim2,
  1, mean), apply(sim3, 1, mean))[5,
  ]
```



```
## [1] 0.4243236 0.4293913 0.4606670

# Mean square error
c(mean((sim1[5, ] - ACE)^2), mean((sim2[5,
  ] - ACE)^2), mean((sim3[5, ] - ACE)^2))

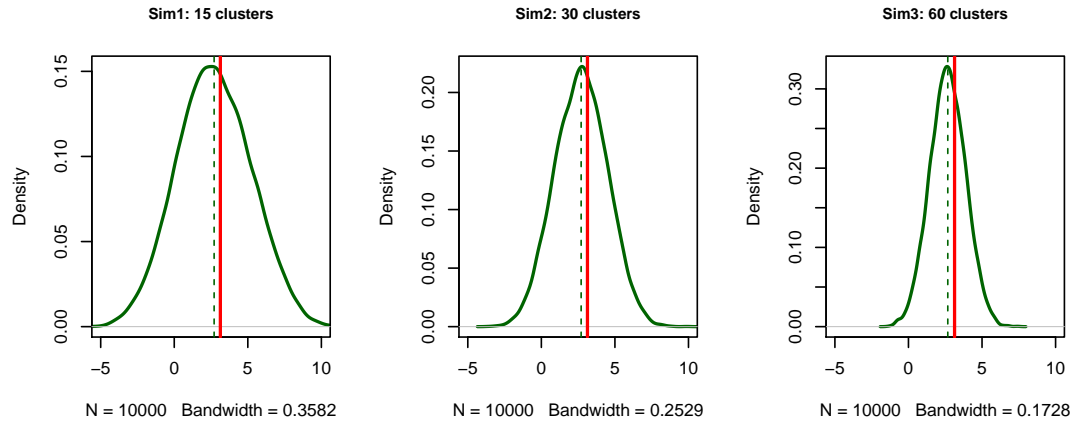
## [1] 6.486587 3.326623 1.680209
```

And we can plot the distributions:

```
par(mfrow = c(1, 3))
plot(density(sim1[5, ]), cex.main = 0.9,
     main = "Sim1: 15 clusters", xlim = c(-5,
       10), lwd = 2, col = "darkgreen")
abline(v = mean(sim1[5, ]), col = "darkgreen",
       lty = 2)
abline(v = ACE, col = "red", lwd = 2)

plot(density(sim2[5, ]), cex.main = 0.9,
     main = "Sim2: 30 clusters", xlim = c(-5,
       10), lwd = 2, col = "darkgreen")
abline(v = mean(sim2[5, ]), col = "darkgreen",
       lty = 2)
abline(v = ACE, col = "red", lwd = 2)

plot(density(sim3[5, ]), cex.main = 0.9,
     main = "Sim3: 60 clusters", xlim = c(-5,
       10), lwd = 2, col = "darkgreen")
abline(v = mean(sim3[5, ]), col = "darkgreen",
       lty = 2)
abline(v = ACE, col = "red", lwd = 2)
```



This comparison illustrates the bias in $\widehat{ATE}_{cluster}$ when the number of observations

varies by cluster. However, the bias and the mean square error decrease as the number of clusters increases.

(h) **What do you conclude from your results in this exercise?**

The exercise confirms the point made in lecture that $\widehat{ATE}_{cluster}$ is not unbiased when the number of units is not the same for each cluster.

2. Maximum Likelihood Estimation

(a) **Likelihood Function.** Suppose that we observe the following independent data:

```
y <- c(10, 4, 5, 3, 9, 2, 7, 3, 6, 4)
```

Plot the log-likelihood function using the Poisson distribution (hint: how do we get the likelihood function of independent data?). Recall that a Poisson distribution describes a discrete variable X using a parameter λ such that for $k = 0, 1, 2, \dots$

$$Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

The positive number λ is equal to the expected value of X and also its variance.

$$\lambda = E(X) = \text{var}(X)$$

Looking at the plot, what does the maximum appear to be?

We have to re-write this probability as a likelihood. First, we derive the full probability distribution:

$$\begin{aligned} \Pr(y_i = y | \lambda) &= \frac{\lambda^y e^{-\lambda}}{y!} \\ \Pr(y | \lambda) &= \prod_{i=1}^n \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} \end{aligned}$$

We can express this as a likelihood:

Next, we take logs to get the log-likelihood function

$$\log(L(\lambda|y)) = \log\left(\prod_{i=1}^n \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}\right)$$

$$\log(L(\lambda|y)) = \sum_{i=1}^n [\log(\lambda^{y_i}) + \log(e^{-\lambda}) - \log(y_i!)]$$

$$\log(L(\lambda|y)) = \sum_{i=1}^n [y_i \log(\lambda) - \lambda - \log(y_i!)]$$

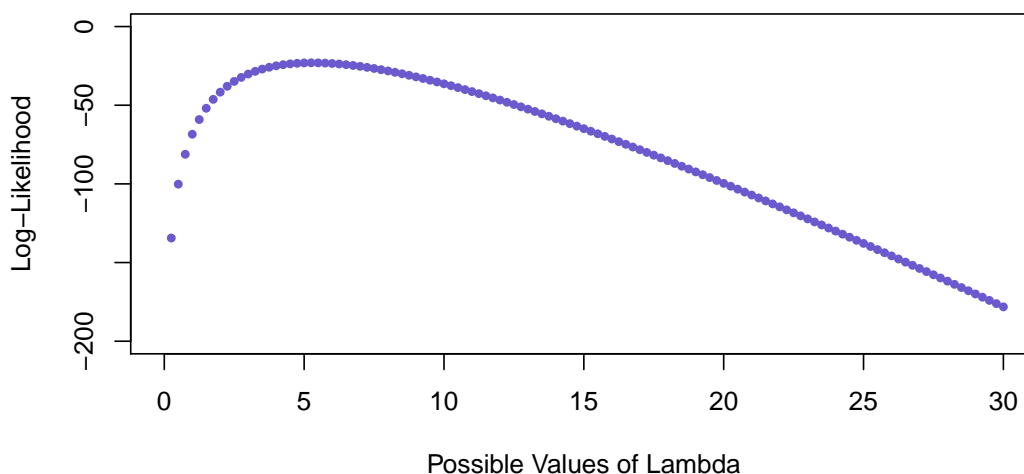
```
##### Plotting the log likelihood function using the Poisson
# distribution
```

```
log.lik <- function(l){ #Let l be lambda
  sum(y * log(l) - l - log(factorial(y)))
}
```

```
p.1<- seq(0,30,0.25) #p.1 are possible values of l
```

```
#plot the log likelihood function
plot(p.1, lapply(p.1,log.lik), pch=16, ylim=c(-200,0),
     #add titles
     xlab="Possible Values of Lambda", ylab="Log-Likelihood",
     col="slateblue", cex=.75,
     main="Graphing the Log-Likelihood Function Using the
     Poisson Distribution")
```

Graphing the Log-Likelihood Function Using the Poisson Distribution



Now use R to solve for the MLE. There are a variety of commands you can use (you can check `optim` or `optimize`, for example). Below is some example code using `optim`.

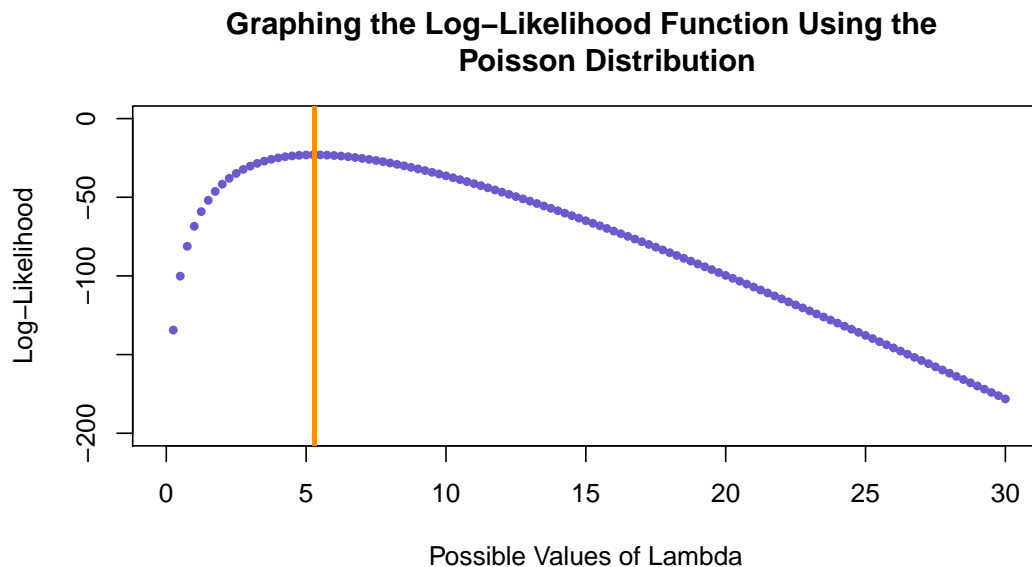
```
##### Using R to solve for the MLE

mle<-optim(
  c(1), #starting values
  log.lik, #the function we are optimizing
  control=list(fnscale=-1), #this tells optim to maximize
  lower=0, #lower bound for optimization
  upper=100, #upper bound
  method="Brent" #one dimensional optimization method
)
mle

## $par
## [1] 5.3
##
## $value
## [1] -23.04246
##
## $counts
## function gradient
##      NA      NA
##
## $convergence
## [1] 0
##
## $message
## NULL

plot(p.1, lapply(p.1,log.lik), pch=16, ylim=c(-200,0),
     xlab="Possible Values of Lambda", ylab="Log-Likelihood",
     col="slateblue", cex=.75,
     main="Graphing the Log-Likelihood Function Using the
     Poisson Distribution")

abline(v=mle$par, col="darkorange", lwd=3)
```



Now compare the MLE estimate of λ to the mean of y . Interpret your results.

```
round(mean(y)) == round(mle$par)

## [1] TRUE
```

Visually, the maximum likelihood appears to be around 5. Using R, we calculate the precise MLE of 5.3. The MLE is the same as the mean of the data. This makes sense since $\lambda = E(y)$. Maximizing the log-likelihood function using the Poisson distribution gives us the most likely λ given the data—which is the mean of the data.

3. **Logistic regression.** A scholar is interested in evaluating the effect of $X \in \mathbb{R}$ (X is continuous from $-\infty$ to ∞) on the probability that $Y = 1$. (Y is dichotomous). The scholar estimates two different logistic regression models M_1 and M_2 .

M_1 :

$$\text{Prob}(Y_i = 1) = \Lambda(\alpha_1 + \beta_{X1}X_i + \gamma_1 Z_{i,M_1}) \quad (1)$$

and

M_2 :

$$\text{Prob}(Y_i = 1) = \Lambda(\alpha_2 + \beta_{X2}X_i + \gamma_2 Z_{i,M_2}). \quad (2)$$

Here, Λ is the logistic distribution function. There are possibly different covariates in each model: Z_{i,M_1} in the first model and Z_{i,M_2} in the second model. Also, γ_j and Z_{i,M_j} may be vectors. X is the same variable in models M_1 and M_2 .

Now, suppose the scholar first estimates logistic regression model M_1 ; this yields an estimated coefficient for X of $\hat{\beta}_{X1} = 0.3$. The scholar then estimates the second model M_2 ; this yields an estimated coefficient for X of $\hat{\beta}_{X2} = 0.8$.

For the following questions, we are setting aside the more vexing problems for causal inference—like, both models can't be simultaneously true. Just assume the models in each case and answer the questions, taking each model as given:

- (a) For each model $M1$ and $M2$, express the marginal effect of a one-unit increase in X on the probability that $Y_i = 1$ in terms of the models in equations (1) and (2), respectively.

The marginal effect under the M_1 model is

$$\lambda(\alpha_1 + \beta_{X1}X_i + \gamma_1 Z_{i,M_1}) * \beta_{X1}, \quad (3)$$

where $\lambda(\alpha_1 + \gamma_1 Z_{i,M_1})$ is the density of logistic distribution evaluated at $\alpha_1 + \gamma_1 Z_{i,M_1}$, and the marginal effect under model M_2 is

$$\lambda(\alpha_2 + \beta_{X2}X_i + \gamma_2 Z_{i,M_2}) * \beta_{X2}, \quad (4)$$

where $\lambda(\alpha_2 + \gamma_2 Z_{i,M_2})$ is the density of the logistic distribution evaluated at $\alpha_2 + \gamma_2 Z_{i,M_2}$.

- (b) The M2 estimate implies that the estimated effect on $Pr(Y = 1)$ of increasing X by one unit is greater than is implied by the M1 estimate. True or false? Explain your answer.

This is false. The change in probability related to a parameter β (and the change in the estimated probability related to $\hat{\beta}$) depends on the baseline probability, which was not specified in the above statement. If we compare the estimated effect for observations with very different baseline probabilities, the estimated effect might be smaller even when the $\hat{\beta}$ is larger. Look at the marginal effects in (i): under model M1, the change in $Pr(Y = 1)$ due to increasing X by one unit depends on the logistic density evaluated at $\alpha_1 + \gamma_1 Z_{i,M_1}$, while under model M1 it depends on that density evaluated at $\alpha_2 + \gamma_2 Z_{i,M_2}$.

Even if we are thinking about the "same" unit, as we do for India-Pakistan-1986, the baseline probability might be very different for models M1 and M2, such that the estimated effect in probability will be smaller under M2 than under M1 if the baseline probability is sufficiently more extreme (close to 0 or 1) under M2 than it was under M1.

- (c) Holding the baseline probability of Y constant, the M2 estimate implies that the estimated effect on $Pr(Y = 1)$ of increasing X by one unit is greater than is implied by the M1 estimate. True or false? Explain your answer.

True. Once you hold the baseline probability constant, a larger coefficient will correspond to a larger marginal effect. Again, look at (i).

- (d) For any particular unit (say India-Pakistan-1986), the M2 estimate implies that the estimated effect on $Pr(Y = 1)$ of increasing X by one unit is greater than is implied by the M1 estimate. True or false? Explain your answer.

This is false. As mentioned for the first question, this will depend on the baseline probability of the unit, and the baseline probability for the same unit will generally be different under two different statistical models.

4. **Probit Model and Interpretation.** For this exercise you will need the `camp1.rda` dataset available on bcourses. Consider the probit model with democratic win (DWIN) as the dependent variable, and the following independent variables:

- JULYECQ2: 2nd Quarter GNP Growth
- PRESINC: Elected Incumbent Seeking Reelection (1=Democrat, 0=no incumbent, 1=Republican)
- ADAACA: State Liberalism Index (ADA & ACA)
- An interaction term between PRESINC and JULYECQ2.

We have the following hypotheses:

H1: *Economic growth in the months prior to the election increases the chances that the Democrat will win.*

H2: *The Democrat has a better chance of winning if he/she is the incumbent President.*

H3: *The more liberal the state, the more likely the Democrat will win.*

H4: *Growth prior to the election only helps the Democrat if he or she is the incumbent.*

Run the analysis in R including an intercept in the model. You can write your own function or use `glm` in R.

```
glm(y ~ x, family = binomial(link = "probit"))
```

```
data$presinc_julyecq2 <- data$presinc *
  data$julyecq2
model <- with(data, glm(dwin ~ julyecq2 +
  presinc + adaaca + presinc_julyecq2,
  family = binomial(link = "probit")))
model

##
## Call:  glm(formula = dwin ~ julyecq2 + presinc + adaaca + presinc_julyecq2,
##        family = binomial(link = "probit"))
##
## Coefficients:
##      (Intercept)
##      -0.461229
##      julyecq2
##      0.020190
##      presinc
##      0.489485
```

```
##          adaaca
##          0.003678
## presinc_julyecq2
##          0.463499
##
## Degrees of Freedom: 543 Total (i.e. Null); 539 Residual
## Null Deviance:      675.7
## Residual Deviance: 518.5  AIC: 528.5
```

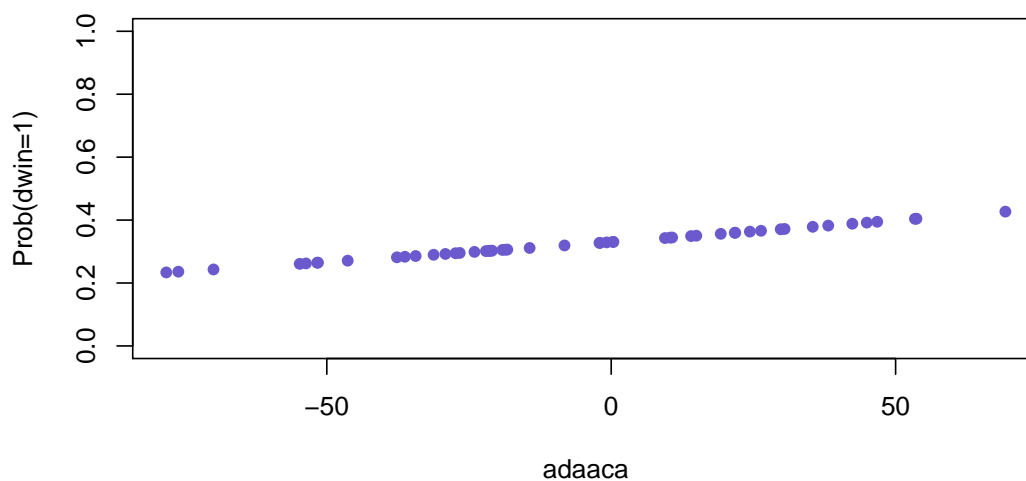
- (a) Suppose you could manipulate *ADAACA*, the state liberalism index. (Just suppose). What is the estimated effect of that variable on a Democratic win, holding all other variables at their median values in the dataset?

Here we will find the predicted

```
unique_adaaca <- unique(data$adaaca) # unique values for ADAACA

l <- as.data.frame(cbind(median(data$julyecq2),
  median(data$presinc)))
fdata <- cbind(l, unique_adaaca, l$V1 *
  l$V2)
names(fdata) <- c("julyecq2", "presinc",
  "adaaca", "presinc_julyecq2")

plot(unique_adaaca, predict(model, newdata = fdata,
  type = "response"), pch = 16, col = "slateblue",
  ylim = c(0, 1), xlab = "adaaca",
  ylab = "Prob(dwin=1)")
```



- (b) Arbitrarily define two levels (low and high) for the variable measuring growth, JULYECQ2, so that 1.5 is “high” growth and -1.5 is “low” growth.

Consider the following four scenarios:

- 1) High growth, incumbent Democratic candidate running.
- 2) High growth, incumbent candidate not running.
- 3) Low growth, incumbent Democratic candidate running.
- 4) Low growth, incumbent candidate not running.

Across the range of values of ADAACA in the dataset, plot the predicted value of $\text{Prob}(\text{DWIN} = 1)$ for each of the four scenarios. (You can use the `predict()` function in R to generate their respective predicted values). Generate two figures: one with the plots for scenarios 1 and 3, and the other with those for scenarios 2 and 4. Please provide a substantive discussion of your results in light of the above hypotheses.

First, we will create four fake datasets, one for each scenario:

```
high_growth_dem_running <- as.data.frame(cbind(1.5,
  1, unique_adaaca))
low_growth_dem_running <- as.data.frame(cbind(-1.5,
  1, unique_adaaca))
high_growth_incnot_running <- as.data.frame(cbind(1.5,
  0, unique_adaaca))
low_growth_incnot_running <- as.data.frame(cbind(-1.5,
  0, unique_adaaca))

names(high_growth_dem_running) <- names(low_growth_dem_running) <- names(high_g
  "presinc", "adaaca")

high_growth_dem_running$presinc_julyecq2 <- high_growth_dem_running$julyecq2 *
  high_growth_dem_running$presinc
low_growth_dem_running$presinc_julyecq2 <- low_growth_dem_running$julyecq2 *
  low_growth_dem_running$presinc
high_growth_incnot_running$presinc_julyecq2 <- high_growth_incnot_running$julye
  high_growth_incnot_running$presinc
low_growth_incnot_running$presinc_julyecq2 <- low_growth_incnot_running$julyecq
  low_growth_incnot_running$presinc
```

Now we want to get the predicted value of $\text{Prob}(\text{dwin} = 1)$ for these scenarios.

```
par(mfrow = c(1, 2))

plot(unique_adaaca, predict(model, newdata = high_growth_dem_running,
  type = "response"), pch = 16, col = "slateblue",
  ylim = c(0, 1), xlab = "adaaca",
  ylab = "Prob(dwin=1)", main = "Democrat Running")
```

```

points(unique_adaaca, predict(model,
  newdata = low_growth_dem_running,
  type = "response"), pch = 16, col = "darkorange")

plot(unique_adaaca, predict(model, newdata = high_growth_incnot_running,
  type = "response"), pch = 16, col = "slateblue",
  ylim = c(0, 1), xlab = "adaaca",
  ylab = "Prob(dwin=1)", main = "Incumbent Not Running")
points(unique_adaaca, predict(model,
  newdata = low_growth_incnot_running,
  type = "response"), pch = 16, col = "darkorange")

```

