# Homework 3 Solutions

## 7.14

### (a)

$$\hat{\theta} = \hat{\beta}_1 \hat{\beta}_2$$

where $\hat{\beta}_1$ and $\hat{\beta}_2$ come from the regression of $Y$ on $X_1$ and $X_2$.

### (b)

First, notice that our usual arguments imply that

$$\sqrt{n}\begin{pmatrix} \hat{\beta}_1 - \beta_1 \\ \hat{\beta}_2 - \beta_2 \end{pmatrix} \xrightarrow{d} \mathcal{N}(0, \mathbf{V}_\beta)$$

where $\mathbf{V}_\beta = \mathbb{E}[XX']^{-1}\mathbf{\Omega}\mathbb{E}[XX']^{-1}$, where we take $X = (X_1, X_2)'$ and where $\mathbf{\Omega} = \mathbb{E}[XX'e^2]$. Note that $\mathbf{V}_\beta$ is a $2 \times 2$ variance matrix.

Next, notice that we can write $\theta = r(\beta_1, \beta_2)$ and $\hat{\theta} = r(\hat{\beta}_1, \hat{\beta}_2)$ where $r(b_1, b_2) = b_1 b_2$. Moreover, using a mean value theorem argument, we have that

$$r(\hat{\beta}_1, \hat{\beta}_2) = r(\beta_1, \beta_2) + \nabla r(\bar{\beta}_1, \bar{\beta}_2)' \begin{pmatrix} \hat{\beta}_1 - \beta_1 \\ \hat{\beta}_2 - \beta_2 \end{pmatrix}$$

where

$$\nabla r(\bar{\beta}_1, \bar{\beta}_2) := \left.\begin{bmatrix} \frac{\partial r(b_1, b_2)}{\partial b_1} \\ \frac{\partial r(b_1, b_2)}{\partial b_2} \end{bmatrix}\right|_{b_1 = \bar{\beta}_1, b_2 = \bar{\beta}_2} = \left.\begin{bmatrix} b_2 \\ b_1 \end{bmatrix}\right|_{b_1 = \bar{\beta}_1, b_2 = \bar{\beta}_2} = \begin{bmatrix} \bar{\beta}_2 \\ \bar{\beta}_1 \end{bmatrix}$$

This implies that

$$\begin{aligned}
\sqrt{n}(\hat{\theta} - \theta) &= \begin{bmatrix} \bar{\beta}_2 \\ \bar{\beta}_1 \end{bmatrix}' \sqrt{n}\begin{pmatrix} \hat{\beta}_1 - \beta_1 \\ \hat{\beta}_2 - \beta_2 \end{pmatrix} \\
&= \begin{bmatrix} \beta_2 \\ \beta_1 \end{bmatrix}' \sqrt{n}\begin{pmatrix} \hat{\beta}_1 - \beta_1 \\ \hat{\beta}_2 - \beta_2 \end{pmatrix} + \underbrace{\left(\begin{bmatrix} \bar{\beta}_2 \\ \bar{\beta}_1 \end{bmatrix} - \begin{bmatrix} \beta_2 \\ \beta_1 \end{bmatrix}\right)'}_{=o_p(1)} \underbrace{\sqrt{n}\begin{pmatrix} \hat{\beta}_1 - \beta_1 \\ \hat{\beta}_2 - \beta_2 \end{pmatrix}}_{=O_p(1)} \\
&= \begin{bmatrix} \beta_2 \\ \beta_1 \end{bmatrix}' \sqrt{n}\begin{pmatrix} \hat{\beta}_1 - \beta_1 \\ \hat{\beta}_2 - \beta_2 \end{pmatrix} + o_p(1) \\
&\xrightarrow{d} \mathcal{N}(0, V)
\end{aligned}$$

where the second equality holds by adding and subtracting, the third equality holds because $\begin{pmatrix} \bar{\beta}_1 \\ \bar{\beta}_2 \end{pmatrix}$

is between $\hat{\beta}$ and $\beta$ (and because $\hat{\beta}$ is consistent for $\beta$), and where

$$V = \begin{bmatrix} \beta_2 \\ \beta_1 \end{bmatrix}' \mathbf{V}_\beta \begin{bmatrix} \beta_2 \\ \beta_1 \end{bmatrix}$$

## (c)

To calculate a 95% confidence interval, the main step is to estimate $V$. The natural estimate is given by

$$\begin{bmatrix} \hat{\beta}_2 \\ \hat{\beta}_1 \end{bmatrix}' \hat{\mathbf{V}}_\beta \begin{bmatrix} \hat{\beta}_2 \\ \hat{\beta}_1 \end{bmatrix}$$

and where we use the usual estimate of $\mathbf{V}_\beta$ that is given by

$$\hat{\mathbf{V}}_\beta = \left( \frac{1}{n} \sum_{i=1}^{n} X_i X_i' \right)^{-1} \frac{1}{n} \sum_{i=1}^{n} X_i X_i' \hat{e}_i^2 \left( \frac{1}{n} \sum_{i=1}^{n} X_i X_i' \right)^{-1}$$

and then we can construct a 95% confidence interval by

$$\hat{C} = \left[ \hat{\theta} \pm 1.96 \frac{\sqrt{\hat{V}}}{\sqrt{n}} \right]$$

## 7.17

## (a)

To start with, let's write $\theta = r(\beta_1, \beta_2) = \beta_1 - \beta_2$. The key step is to derive an expression for $\sqrt{n}(\hat{\theta} - \theta)$. This is a linear function of the parameters, i.e., we can write $\theta = \mathbf{R}'\beta$ where $\mathbf{R} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Therefore, we have that

$$\begin{aligned} \sqrt{n}(\hat{\theta} - \theta) &= \mathbf{R}' \sqrt{n}(\hat{\beta} - \beta) \\ &\xrightarrow{d} \mathcal{N}(0, V) \end{aligned}$$

where

$$\begin{aligned} V &= \mathbf{R}' \mathbf{V}_\beta \mathbf{R} \\ &= \begin{bmatrix} 1 \\ -1 \end{bmatrix}' \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} (V_{11} - V_{21}) & (V_{12} - V_{22}) \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= V_{11} - V_{21} - V_{12} + V_{22} \end{aligned}$$

where $V_{ij}$ denotes the element in the ith row and jth column in $\mathbf{V}_\beta$. This is the main theoretical result that we needed to show, but we would still need to estimate $V$ in order to come up with a confidence interval. Before doing that, it is useful to note that we can write a $2 \times 2$ variance matrix, like $\mathbf{V}_\beta$ as

$$\mathbf{V}_\beta = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} = \begin{bmatrix} V_{11} & \rho\sqrt{V_{11}}\sqrt{V_{22}} \\ \rho\sqrt{V_{11}}\sqrt{V_{22}} & V_{22} \end{bmatrix}$$

which holds because the diagonal elements of this matrix are variance, and the off diagonals are covariances (and recalling that $\text{cov}(X,Y) = \text{corr}(X,Y)\sqrt{\text{var}(X)}\sqrt{\text{var}(Y)}$ — which just holds from the definition of correlation). This suggests that,

$$\hat{V} = \hat{V}_{11} - 2\hat{\rho}\sqrt{\hat{V}_{11}}\sqrt{\hat{V}_{22}} + \hat{V}_{22}$$

which further implies that

$$\begin{aligned} \frac{\hat{V}}{n} &= \frac{\hat{V}_{11}}{n} - 2\hat{\rho}\frac{\sqrt{\hat{V}_{11}}}{\sqrt{n}}\frac{\sqrt{\hat{V}_{22}}}{\sqrt{n}} + \frac{\hat{V}_{22}}{n} \\ &= \text{se}(\hat{\beta}_1)^2 - 2\hat{\rho}\,\text{se}(\hat{\beta}_1)\,\text{se}(\hat{\beta}_2) + \text{se}(\hat{\beta}_2)^2 \end{aligned}$$

Finally, we can write down a 95% confidence interval as

$$\begin{aligned} \hat{C} &= \left[\hat{\theta} \pm 1.96\sqrt{\frac{\hat{V}}{n}}\right] \\ &= \left[\hat{\theta} \pm 1.96\sqrt{\text{se}(\hat{\beta}_1)^2 - 2\hat{\rho}\,\text{se}(\hat{\beta}_1)\,\text{se}(\hat{\beta}_2) + \text{se}(\hat{\beta}_2)^2}\right] \end{aligned}$$

where the first line is just the usual confidence interval (i.e., estimate plus or minus critical value times standard error), and the second equality plugs in the expression for $\hat{V}/n$ derived above.

## (b)

No, it is not possible to calculate $\hat{\rho}$ from the information given in the problem. Besides the estimates of $\hat{\beta}_1$ and $\hat{\beta}_2$, the only other information that we have is about $\text{se}(\hat{\beta}_1)$ and $\text{se}(\hat{\beta}_2)$ — which does not tell us about their correlation.

## (c)

I think the way to think about this problem is to think about the largest possible confidence interval given the information that we have. If this confidence interval does not include 0, then it would support the author's claim. As a side-comment, this is actually a really interesting question (at least in my view) because: on the one hand, you can immediately see that the 95% confidence interval for $\beta_1$ would not include the estimated value of $\beta_2$ (which is probably what the author is thinking), on the other hand, if you compute both confidence intervals for $\beta_1$ and $\beta_2$, they overlap (which would suggest that they are not different from each other). These are just heuristic arguments though, and our calculations above indicate that it is actually more complicated than either of these scenarios. Anyway…the widest possible confidence interval here will occur when $\hat{\rho} = -1$ (you can see this because it shows up in the negative term in the square root). Therefore, the widest

possible confidence interval is given by

$$\begin{aligned}\hat{C}^{wide} &= \left[\hat{\theta} \pm 1.96\sqrt{\text{se}(\hat{\beta}_1)^2 + 2\text{se}(\hat{\beta}_1)\text{se}(\hat{\beta}_2) + \text{se}(\hat{\beta}_2)^2}\right] \\ &= \left[0.2 \pm 1.96\sqrt{4 \times 0.07^2}\right] \\ &= [-0.07, 0.47]\end{aligned}$$

This includes 0, which suggests that the author's claim is not correct. The information that we have from the problem does not necessarily imply that $\hat{\beta}_1$ and $\hat{\beta}_2$ are statistically different from each other.

## 7.28

### (a)

**We did part (a) on the previous homework, I am showing those results here so we can compare to them later**

```
# read data
library(haven)
cps <- read_dta("cps09mar.dta")

# construct subset of white, male, Hispanic
data <- subset(cps, race == 1 & female == 0 & hisp == 1)

# construct experience and wage
data$exp <- data$age - data$education - 6
data$wage <- data$earnings / (data$hours * data$week)

# run regression
Y <- log(data$wage)
X <- cbind(data$education, data$exp, data$exp^2 / 100, 1)
bet <- solve(t(X) %*% X) %*% t(X) %*% Y
round(bet, 5)
```

```
         [,1]
[1,]  0.09045
[2,]  0.03538
[3,] -0.04651
[4,]  1.18521
```

```
# construct standard errors
ehat <- as.numeric(Y - X %*% bet)
Xe <- X * ehat
n <- nrow(data)
Omeg <- t(Xe) %*% Xe / n
XX <- t(X) %*% X / n
V <- solve(XX) %*% Omeg %*% solve(XX)
```

```r
se <- sqrt(diag(V)) / sqrt(n)
round(data.frame(beta = bet, se = se), 5)
```

```
      beta      se
1  0.09045 0.00292
2  0.03538 0.00258
3 -0.04651 0.00530
4  1.18521 0.04608
```

**(b)**

$$\theta = \frac{\beta_1}{\beta_2 + 2\beta_3(10)/100} = \frac{\beta_1}{\beta_2 + \frac{1}{5}\beta_3}$$

```r
thet <- bet[1] / (bet[2] + bet[3] / 5)
thet
```

```
[1] 3.468335
```

**(c)**

We can use a Delta method argument for this. In particular, define

$$r(b) = \frac{b_1}{b_2 + \frac{1}{5}b_3}$$

and, therefore, we have that

$$\sqrt{n}(\hat{\theta} - \theta) = \nabla r(\beta)' \sqrt{n}(\hat{\beta} - \beta) + o_p(1)$$

where

$$\nabla r(\beta) = \left[\begin{array}{c} \frac{\partial r(b)}{\partial b_1} \\ \frac{\partial r(b)}{\partial b_2} \\ \frac{\partial r(b)}{\partial b_3} \\ \frac{\partial r(b)}{\partial b_4} \end{array}\right]\Bigg|_{b=\beta} = \left[\begin{array}{c} \frac{1}{b_2 + \frac{1}{5}b_3} \\ -\frac{b_1}{(b_2 + \frac{1}{5}b_3)^2} \\ -\frac{b_1}{5(b_2 + \frac{1}{5}b_3)^2} \\ 0 \end{array}\right]\Bigg|_{b=\beta}$$

Thus, we have that

$$\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{d} \mathcal{N}(0, \Gamma)$$

where

$$\Gamma = \nabla r(\beta)' \mathbf{V}_\beta \nabla r(\beta)$$

which we can estimate by

$$\hat{\Gamma} = \begin{bmatrix} \frac{1}{\hat{\beta}_2 + \frac{1}{5}\hat{\beta}_3} \\ -\frac{\hat{\beta}_1}{(\hat{\beta}_2 + \frac{1}{5}\hat{\beta}_3)^2} \\ -\frac{\hat{\beta}_1}{5(\hat{\beta}_2 + \frac{1}{5}\hat{\beta}_3)^2} \\ 0 \end{bmatrix}' \hat{\mathbf{V}}_\beta \begin{bmatrix} \frac{1}{\hat{\beta}_2 + \frac{1}{5}\hat{\beta}_3} \\ -\frac{\hat{\beta}_1}{(\hat{\beta}_2 + \frac{1}{5}\hat{\beta}_3)^2} \\ -\frac{\hat{\beta}_1}{5(\hat{\beta}_2 + \frac{1}{5}\hat{\beta}_3)^2} \\ 0 \end{bmatrix}$$

and

$$\text{s.e.}(\hat{\theta}) = \frac{\sqrt{\hat{\Gamma}}}{\sqrt{n}}$$

```
# se(\hat{\theta})
r1 <- 1 / (bet[2] + bet[3] / 5)
r2 <- -bet[1] / (bet[2] + bet[3] / 5)^2
r3 <- -bet[1] / (5 * (bet[2] + bet[3] / 5)^2)
r4 <- 0
r <- as.matrix(c(r1, r2, r3, r4))
Gamma <- t(r) %*% V %*% r
se_theta <- sqrt(Gamma) / sqrt(n)
se_theta
```

```
          [,1]
[1,] 0.2267341
```

**(d)**

```
# 90% confidence interval
ci_thet_L <- thet - 1.645 * se_theta
ci_thet_U <- thet + 1.645 * se_theta
paste0("[", round(ci_thet_L, 3), ", ", round(ci_thet_U, 3), "]")
```

```
[1] "[3.095, 3.841]"
```

**(e)**

```
# compute regression intervals and 95% confidence interval
x <- c(12, 20, 20^2 / 100, 1)
m <- t(x) %*% bet
m
```

```
         [,1]
[1,] 2.792167
```

```r
Vm <- t(x) %*% V %*% x
sem <- sqrt(Vm) / sqrt(n)
L <- m - 1.96 * sem
U <- m + 1.96 * sem
paste0("[", round(L, 3), ", ", round(U, 3), "]")
```

```
[1] "[2.769, 2.815]"
```

### Extra Question 1

```r
set.seed(1234) # set seed for reproducibility

B <- 1000 # 1000 bootstrap iterations

boot_res <- list() # list to hold bootstrap results
for (b in 1:B) {
  # draw a sample
  index <- sample(1:n, replace=TRUE)
  boot_data <- data[index, ]
  Yb <- log(boot_data$wage)
  Xb <- cbind(boot_data$education, boot_data$exp, boot_data$exp^2/100, 1)
  betb <- solve(t(Xb)%*%Xb)%*%t(Xb)%*%Yb
  boot_res[[b]] <- betb
}

# bootstrap list to matrix
boot_res <- do.call(cbind, boot_res)
boot_res <- t(boot_res) # and take transpose for cov call below

# bootstrap estimate of asymptotic variance
Vb <- cov(boot_res)*n

# standard errors
seb <- sqrt(diag(Vb))/sqrt(n)

# compare to analytical standard errors computed above
round(data.frame(analytical=se, bootstrap=seb),5)
```

```
  analytical bootstrap
1    0.00292   0.00290
2    0.00258   0.00257
3    0.00530   0.00525
4    0.04608   0.04568
```

You can see that these are not exactly the same, but they are very close.

## Extra Question 2

### (a)

To start with recall that we are going to estimate the parameters in the probit model by solving the following optimization problem:

$$\hat{\beta} = \underset{b}{\text{argmax}} \frac{1}{n} \sum_{i=1}^{n} Y_i \log(\Phi(X_i'b)) + (1 - Y_i) \log(1 - \Phi(X_i'b))$$

and where it is also helpful to recall that we defined the score as the derivative of this objective function taken with respect to the parameters:

$$S_n(b) = \frac{1}{n} \sum_{i=1}^{n} \frac{(Y_i - \Phi(X_i'b))\phi(X_i'b)}{\Phi(X_i'b)(1 - \Phi(X_i'b))} X_i$$

```
library(haven)
data <- read_dta("cps09mar.dta")
data <- subset(data, female==0)
data$black <- as.integer(data$race == 2)
Y <- data$union
X <- as.matrix(data[,c("age","education","black","hisp")])
X <- cbind(1,X)
n <- nrow(data)

# log-likelihood as function of parameters
ll <- function(b, X, Y) {
  G <- pnorm(X%*%b)
  mean( Y*log(G) + (1-Y)*log(1-G) )
}

# score function
s <- function(b, X, Y) {
  G <- pnorm(X%*%b)
  g <- dnorm(X%*%b)

  # calculates mean across units (returning k-dim vector)
  apply(as.numeric(Y*(g/G))*X - as.numeric((1-Y)*(g/(1-G)))*X, 2, mean)
}

k <- ncol(X)
start_bet <- rep(0,k)
prob_est <- optim(start_bet, ll, gr=s,
                  X=X, Y=Y,
                  method="BFGS",
                  control=list(fnscale=-1))
bet <- prob_est$par
```

In order to calculate standard errors, recall that we showed in class that

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{d} \mathcal{N}(0, \mathbf{\Omega}^{-1})$$

where $\mathbf{\Omega} = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))} XX'\right]$, and that we could estimate $\mathbf{\Omega}$ by

$$\hat{\mathbf{\Omega}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\phi(X_i'\hat{\beta})^2}{\Phi(X_i'\hat{\beta})(1-\Phi(X_i'\hat{\beta}))} X_i X_i'$$

```
idx <- as.numeric(X%*%bet)
O1 <- ( dnorm(idx)^2 / ( pnorm(idx)*(1-pnorm(idx)) ) ) * X
Omeg <- t(O1)%*%X/n
Omeg_inv <- solve(Omeg)
se <- sqrt(diag(Omeg_inv))/sqrt(n)
round(cbind.data.frame(bet=bet, se=se, t=bet/se), 6)
```

```
                bet       se          t
          -1.892941 0.106813 -17.721951
age        0.007384 0.001416   5.215583
education -0.028084 0.006212  -4.521237
black     -0.063187 0.060269  -1.048417
hisp      -0.289155 0.056130  -5.151499
```

Finally, let's compare these results to the ones that we get from R's `glm` command

```
# compare to results from R
R_probit <- glm(union ~ age + education + black + hisp,
                family=binomial(link=probit), data=data)
summary(R_probit)
```

```
Call:
glm(formula = union ~ age + education + black + hisp, family = binomial(link = probit),
    data = data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.953818   0.107176 -18.230  < 2e-16 ***
age          0.007925   0.001419   5.584 2.35e-08 ***
education   -0.025505   0.006221  -4.100 4.14e-05 ***
black       -0.054083   0.060004  -0.901    0.367
hisp        -0.297745   0.056865  -5.236 1.64e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 6282.0  on 29139  degrees of freedom
Residual deviance: 6210.7  on 29135  degrees of freedom
AIC: 6220.7

Number of Fisher Scoring iterations: 6
```

These are slightly different from each other. I checked the documentation for `glm` and it uses an "iteratively reweighted least squares" estimation procedure; this is different from the optimization procedure that I used and explains the difference between the estimates.

### (b)

To start with, let's compute estimates of average marginal contrasts. Given what we have already done, this is fairly easy.

```
# compute average marginal contrasts
mc <- dnorm(X%*%bet) %*% t(bet)
amc <- apply(mc, 2, mean)
names(amc) <- colnames(X)
round(amc, 6)
```

```
                age education     black      hisp
-0.100908  0.000394 -0.001497 -0.003368 -0.015414
```

### (c)

Next, let's derive the limiting distribution of the the average marginal contrasts and use this to compute standard errors.

Starting from the hint in the problem

$$\begin{aligned}
\sqrt{n}(\widehat{AMC} - AMC) &= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(X_i'\hat{\beta})\hat{\beta} - \mathbb{E}[\phi(X'\beta)\beta]\right) \\
&= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(X_i'\hat{\beta})\hat{\beta} - \frac{1}{n}\sum_{i=1}^{n}\phi(X_i'\hat{\beta})\beta\right) \qquad \text{(A)} \\
&+ \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(X_i'\hat{\beta})\beta - \frac{1}{n}\sum_{i=1}^{n}\phi(X_i'\beta)\beta\right) \qquad \text{(B)} \\
&+ \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(X_i'\beta)\beta - \mathbb{E}[\phi(X'\beta)\beta]\right) \qquad \text{(C)}
\end{aligned}$$

It is helpful to recall from class that we defined

$$\begin{aligned}
\mathbf{Q} &= -\mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}XX'\right] \\
\psi(Y,X,b) &= -\frac{(Y-\Phi(X'b))\phi(X'b)}{\Phi(X'b)(1-\Phi(X'b))}X
\end{aligned}$$

and we showed that

$$\sqrt{n}(\hat{\beta}-\beta) = -\mathbf{Q}^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi(Y_i, X_i, \beta) + o_p(1)$$

which was the intermediate step before we applied the CLT.

Now, let's consider each term in turn in the expression for $\sqrt{n}(\widehat{AMC} - AMC)$, starting with the first one.

$$\begin{aligned}
(A) &= \frac{1}{n}\sum_{i=1}^{n}\phi(X_i'\hat{\beta})\sqrt{n}(\hat{\beta}-\beta) \\
&= \mathbb{E}[\phi(X'\beta)]\sqrt{n}(\hat{\beta}-\beta) + o_p(1) \\
&= \mathbb{E}[\phi(X'\beta)]\mathbf{Q}^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi(Y_i, X_i, \beta) + o_p(1) \\
&= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbb{E}[\phi(X'\beta)]\mathbf{Q}^{-1}\psi(Y_i, X_i, \beta) + o_p(1) \\
&:= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}A_i + o_p(1)
\end{aligned}$$

where the second equality holds by the law of large numbers and CMT, the third line holds by what we showed in class for $\sqrt{n}(\hat{\beta}-\beta)$, the fourth line just rearranges in a way that will be convenient below, and the fifth equality just introduces a more concise notation.

Next, consider the second term in the hint (this is hardest term to deal with). Notice that we can write

$$\begin{aligned}
(B) &= \beta\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\phi(X_i'\hat{\beta}) - \phi(X_i'\beta) \\
&= \beta\frac{1}{n}\sum_{i=1}^{n}\phi'(X_i'\beta)X_i'\sqrt{n}(\hat{\beta}-\beta) + o_p(1) \\
&= -\beta\frac{1}{n}\sum_{i=1}^{n}X_i'\beta\phi(X_i'\beta)X_i'\sqrt{n}(\hat{\beta}-\beta) + o_p(1) \\
&= -\beta\mathbb{E}[X'\beta\phi(X'\beta)X']\sqrt{n}(\hat{\beta}-\beta) + o_p(1) \\
&= \beta\mathbb{E}[X'\beta\phi(X'\beta)X']\mathbf{Q}^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi(Y_i, X_i, \beta) + o_p(1) \\
&= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\beta\mathbb{E}[X'\beta\phi(X'\beta)X']\mathbf{Q}^{-1}\psi(Y_i, X_i, \beta) + o_p(1) \\
&:= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}B_i + o_p(1)
\end{aligned}$$

where the first equality holds just by re-arranging, the second equality holds using delta method / mean value theorem type of argument, the third equality holds because $\phi'(z) = -z\phi(z)$ (this is a property of a standard normal distribution), the fourth equality holds by the law of large numbers and CMT, the fifth equality holds by what we showed in class for the asymptotically linear representation of $\sqrt{n}(\hat{\beta}-\beta)$, the sixth equality just re-arranges by pushing inside the sum

the expectation terms, and the last line defines $B_i$.

Finally, the third term is immediately equal to

$$\begin{aligned}(C) &= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\beta(\phi(X_i'\beta) - \mathbb{E}[\phi(X'\beta)]) \\ &:= \frac{1}{\sqrt{n}}\sum_{i=1}^{n} C_i\end{aligned}$$

Thus, we can write

$$\sqrt{n}(\widehat{AMC} - AMC) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}(A_i + B_i + C_i) + o_p(1)$$

If you plug in the expressions for $A_i, B_i, C_i$, this expression will look very complicated, but it is mean 0, and we can apply the CLT to it. In fact, it immediately follows that

$$\sqrt{n}(\widehat{AMC} - AMC) \xrightarrow{d} N(0, \mathbf{V})$$

where

$$\mathbf{V} = \mathbb{E}[(A + B + C)(A + B + C)']$$

Estimating $\mathbf{V}$ just involves plugging in sample quantities for population quantities — again: these expressions will be long, but if you do it carefully, everything should work. This is what we will do next.

### (d)

```
# compute standard errors
idx <- as.numeric(X%*%bet) #nx1
Q <- -Omeg # kxk
G <- pnorm(X%*%bet) # nx1
g <- dnorm(X%*%bet) # nx1
# psi is nxk matrix
psi <- -as.numeric(Y*(g/G))*X + as.numeric((1-Y)*(g/(1-G)))*X

# compute A
a1 <- mean(g)
A <- -a1*psi%*%solve(Q)

# compute B
b1 <- t(apply(as.numeric(idx * g) * X, 2, mean))
B <- t(-as.matrix(bet)%*%b1%*%solve(Q)%*%t(psi))

# compute C
C <- as.numeric(g - mean(g))%*%t(bet)
```

```
# compute variance
inf_func <- A + B + C
# estimate variance
amc_V <- t(inf_func)%*%inf_func/n
amc_se <- sqrt(diag(amc_V))/sqrt(n)
round(cbind.data.frame(amc, amc_se, t=(amc/amc_se)),4)
```

```
              amc amc_se        t
          -0.1009 0.0065 -15.5566
age        0.0004 0.0001   5.4786
education -0.0015 0.0003  -4.9531
black     -0.0034 0.0033  -1.0345
hisp      -0.0154 0.0031  -4.8964
```

Let's compare these to what you get if you use R to compute average marginal contrasts.

```
library(marginaleffects)
avg_slopes(R_probit)
```

```
      Term Contrast  Estimate Std. Error      z Pr(>|z|)    S     2.5 %
 age          dY/dX  0.000421   7.63e-05  5.511   <0.001 24.7  0.000271
 black        1 - 0 -0.002746   2.91e-03 -0.943    0.346  1.5 -0.008454
 education    dY/dX -0.001354   3.33e-04 -4.070   <0.001 14.4 -0.002006
 hisp         1 - 0 -0.012818   1.97e-03 -6.512   <0.001 33.7 -0.016676
    97.5 %
  0.000570
  0.002961
 -0.000702
 -0.008960

Type: response
```

As before, we're getting slightly different results. This is expected though; recall, that our original probit estimates were slightly different from the ones coming from R which would suggest that we'd expect slightly different AMCs. That said, these are reasonable close and suggest that we do not have a coding error or anything like that.

### (e)

This is quite similar to what we have done for the bootstrap before. The code below uses parallel processing to speed up computation using the `pbapply` package. The bootstrap is an example of what's sometimes called an "embarrassingly parallel" problem – this is kind of a strange name, but it just means that it's an obvious place to use parallel processing. The reason is that each bootstrap iteration is fully independent of other bootstrap iterations, so you can run lots of these at the same time and then compute standard errors (or whatever you want) after you have carried out all of the bootstrap iterations.

```r
# finally, compute standard errors using the bootstrap
biters <- 100
library(pbapply) # for computing in parallel
boot_res <- pblapply(1:biters, function(b) {

  # draw new data with replacement
  boot_rows <- sample(1:n, size=n, replace=TRUE)
  boot_data <- data[boot_rows,]
  boot.Y <- boot_data$union
  boot.X <- as.matrix(boot_data[,c("age","education","black","hisp")])
  boot.X <- cbind(1,boot.X)

  # estimate probit using new data
  boot_est <- optim(start_bet, ll, gr=s,
                    X=boot.X,
                    Y=boot.Y,
                    method="BFGS",
                    control=list(fnscale=-1))
  boot_bet <- boot_est$par

  # compute average marginal contrasts
  boot_mc <- dnorm(boot.X%*%boot_bet) %*% t(boot_bet)
  boot_amc <- apply(boot_mc, 2, mean)

  # return results
  boot_amc
}, cl=2)

# run bootstrap
boot_res <- do.call("rbind", boot_res)

# compute bootstrap standard errors
boot_se <- apply(boot_res, 2, sd)

# compare to earlier standard errors
round(cbind.data.frame(amc=amc, amc_se=amc_se, boot_se=boot_se),4)
```

```
               amc amc_se boot_se
          -0.1009 0.0065  0.0059
age        0.0004 0.0001  0.0001
education -0.0015 0.0003  0.0003
black     -0.0034 0.0033  0.0031
hisp      -0.0154 0.0031  0.0035
```

These standard errors are very similar to the ones we calculated using asymptotic theory. Also, notice that, for me, it takes about a minute to compute the bootstrap standard errors, but only a second or two to compute the asymptotic standard errors. However, it only took 5 or 10 minutes for me to write the bootstrap code while it took me close to two hours to figure out the limiting

distribution of the average marginal contrasts and write the code for them (these are also probably more prone to making mistakes here because the arguments are more complicated).