ARE 212 Problem Set 4

Aline Abayo, Eleanor Adachi, Anna Cheyette, Karla Neri, and Stephen Stack

2024-03-03

```
# Comment out after installing
# install.packages("pacman")

options(scipen = 999)

# Load packages
library(pacman)
p_load(dplyr, haven, readr, knitr, readxl, psych, ggplot2, stats4)
#p_load(lmSupport, magrittr, qwraps2, car, lmtest, stargazer,sandwich)
# Remove lmSupport, add tidyverse and dplyr and margins
p_load(magrittr, qwraps2, car, lmtest, stargazer,sandwich, tidyverse, margins)

# get directory of current file
current_directory <-
  dirname(dirname(rstudioapi::getSourceEditorContext()$path))
```

Exercise 1

For this question, please download the dataset pset4_2024.dta from bCourses.

```
# Load data
my_data <- read_dta(file.path(current_directory, "data", "pset4_2024.dta"))
```

We would like to estimate the model:

$logquantity_i = \beta_1 + fuel_i\beta_2 + logprice_i\beta_3 + weight_i\beta_4 + \epsilon_i$ (eq. 1)

1. Estimate the model above in (eq.1) via OLS.

NOTE: Per Sofia's announcement on February 23rd, using weight instead of weight squared.

```
# Create lprice and lqu
my_data$lprice <- log(my_data$price)
my_data$lqu <- log(my_data$qu)

# declare X and y variables
X1 <- cbind(1, my_data$fuel, my_data$lprice, my_data$weight)
y1 <- my_data$lqu

n1 <- length(y1)
# degrees of freedom
df <- nrow(X1) - ncol(X1)
# Find coefficient vector
```

```r
b1 <- solve(t(X1) %*% X1) %*% t(X1) %*% y1
b1
```

```
##              [,1]
## [1,] 13.021293997
## [2,] -0.091952918
## [3,] -0.797538193
## [4,] -0.001074254
```

```r
# projection matrix of reg y on X
P <- X1%*%solve(t(X1)%*%X1)%*%t(X1)
# residual maker of reg y on X: M= I - P
M <- diag(n1)-P
# residuals
e <- M%*%y1
# varcov matrix b
s2 <- as.numeric(t(e)%*%e)/df
vb <- s2*solve(t(X1)%*%X1)
# std error of b
seOLS1 <- sqrt(diag(vb))
```

2. Conduct a Breusch Pagan Test

Conduct a Breusch-Pagan test for heteroskedastic errors using the canned reg12<-lm(lqu~ etc) and bptest(reg13). Do we have a problem?

From Lecture 9: For Breusch-Pagan test, let $\sigma_i^2 = \sigma^2 f(\alpha_0 + \alpha' z_i)$

where z_i is a vector of independent variables. If $\alpha = 0$, then the model is homoskedastic.

Null hypothesis: H_0: $\alpha = 0$ (homoskedastic)

Alternative hypothesis: H_a: $\alpha \neq 0$ (heteroskedastic)

```r
reg2 <- lm(lqu~fuel+lprice+weight, my_data)
summary(reg2)
```

```
##
## Call:
## lm(formula = lqu ~ fuel + lprice + weight, data = my_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3033 -0.9477  0.1108  1.0524  3.1891
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept) 13.021294   0.223353  58.299 < 0.0000000000000002 ***
## fuel        -0.091953   0.025616  -3.590              0.00034 ***
## lprice      -0.797538   0.148890  -5.357         0.0000000953 ***
## weight      -0.001074   0.000240  -4.475         0.0000080823 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.404 on 1874 degrees of freedom
## Multiple R-squared:  0.1787, Adjusted R-squared:  0.1774
## F-statistic: 135.9 on 3 and 1874 DF,  p-value: < 0.00000000000000022
```

```
bptest(reg2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  reg2
## BP = 56.873, df = 3, p-value = 0.000000000002735
```

The test statistic is 56.873 and the corresponding p-value is 0.000000000002735. Since the p-value is less than 0.05, we reject the null hypothesis and conclude that heteroscedasticity is present in the regression model.

Yes, we have a problem because our current standard error calculation assumes homoskedasticity.

3. Calculate the White robust standard errors.

Calculate the White robust standard errors.

```
# element-by-element multiplication of epsilons and X matrix
Xe <- cbind(e, my_data$fuel*e, my_data$lprice*e, my_data$weight*e)

# calculate White robust variance-covariance matrix, assume large samples
Vb_whiteRobust <- solve(t(X1)%*%X1) %*% t(Xe)%*%Xe %*% solve(t(X1)%*%X1)

# calculate White robust standard errors
seWhite1 <- sqrt(diag(Vb_whiteRobust))
seWhite1
```

```
## [1] 0.2238751058 0.0232433645 0.1485702322 0.0002495783
```

Comment on how they compare to the traditional OLS standard errors. Is this the right way to go about dealing with potential heterogeneity problems?

```
seOLS1
```

```
## [1] 0.2233526284 0.0256164221 0.1488903142 0.0002400311
```

```
seWhite1
```

```
## [1] 0.2238751058 0.0232433645 0.1485702322 0.0002495783
```

This is probably the best way for us to deal with heteroskedasticity assuming that we don't know the specific form of heteroskedasticity (i.e., we don't know Ω). We observe that some of White robust standard errors are smaller than their OLS counterparts and some are larger.

The sample size **1878** is large compared to the number of explanatory variables so the small sample correction is not necessary.

4. Delta method

Suppose that there is a model where a structural parameter of interest γ is defined as

$\gamma = log(\beta_2 + 2)(\beta_3 + 3\beta_4)$

Using the OLS estimation results from eq. 1, calculate $\hat{\gamma}$ and its white standard error (hint: think Delta Method).

Null hypothesis: $g(\beta) = q$

Alternative hypothesis: $g(\beta) \neq q$

From Lecture 9: Let G be the matrix of first derivatives of β, $G(\beta) = \frac{\partial g(\beta)}{\partial \beta'}$

Delta method: $\sqrt{N}(g(b_N) - g(\beta)) \xrightarrow{\text{d}} N(0, G(\beta)\sigma^2 Q^{-1} G(\beta)')$

```
b1_2 <- b1[[2]]
b1_3 <- b1[[3]]
b1_4 <- b1[[4]]

# gamma hat
gammahat <- log(b1_2 + 2)*(b1_3 + 3*b1_4)
gammahat
```

```
## [1] -0.5173558
```

```
# Delta method white robust
# gradient relative to beta0 is 0
G <- cbind(0, (b1_3+3*b1_4)/(b1_2+2), log(b1_2+2), 3*log(b1_2+2) )
Vgwhite <- G %*% Vb_whiteRobust %*% t(G)

# Delta method White robust standard errors
segwhite <- sqrt(diag(Vgwhite))
segwhite
```

```
## [1] 0.1008321
```

Exercise 2

Let the equation (eq.2) be the linear model of log quantity lqu, where lprice=log(price), given by

$lqu_i = \beta_0 + fuel_i\beta_1 + lprice_i\beta_2 + year_i\beta_3 + weight_i\beta_4 + luxury_i\beta_5 + \epsilon_i$ (eq. 2)

Given the OLS estimates,

1. Please interpret your results in terms of the log price variable OLS coefficient.

```
# declare X and y variables
X2 <- cbind(1, my_data$fuel, my_data$lprice, my_data$year, my_data$weight, my_data$luxury)
y2 <- my_data$lqu

n2 <- length(y2)
# degrees of freedom
df <- nrow(X2) - ncol(X2)
# Find coefficient vector
b2 <- solve(t(X2) %*% X2) %*% t(X2) %*% y2
b2
```

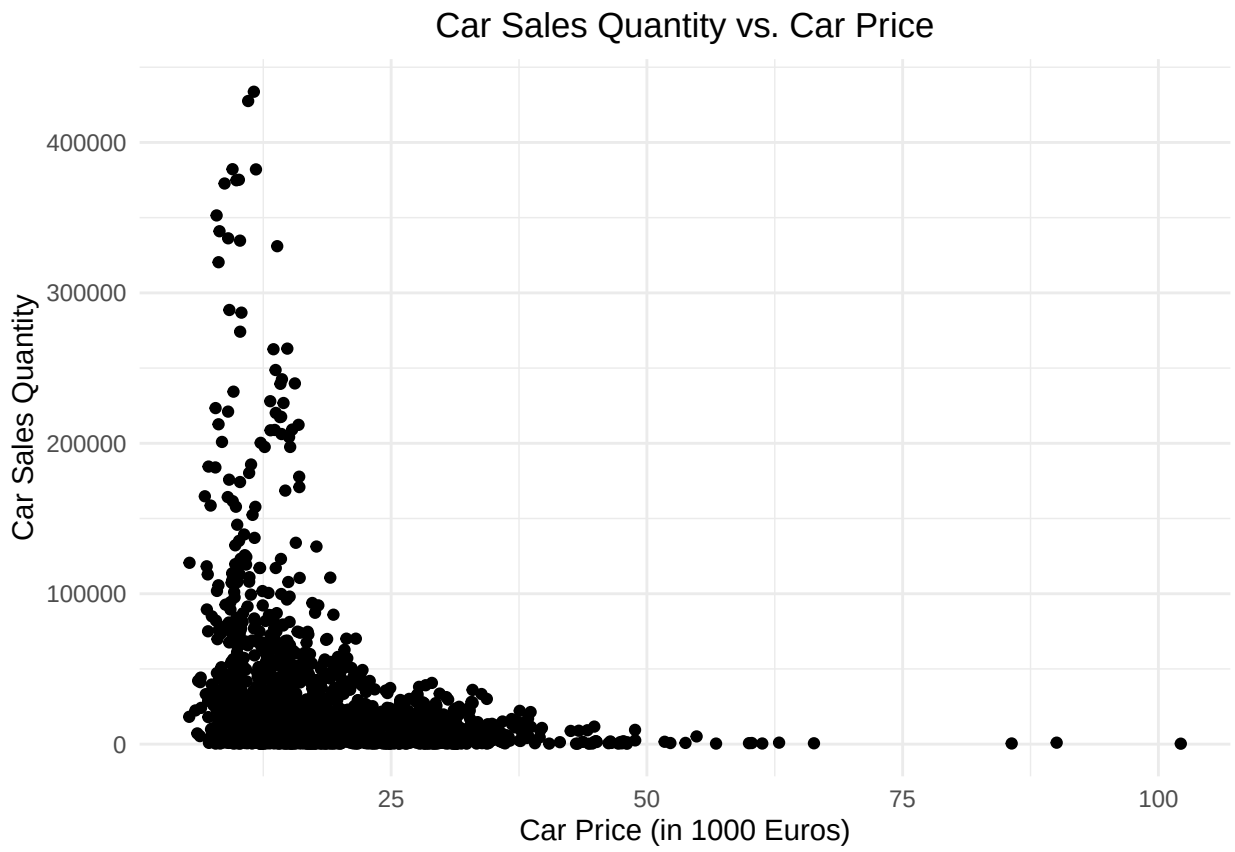
```
##                [,1]
## [1,] 68.7789505827
## [2,] -0.1524851356
## [3,] -1.4418206274
## [4,] -0.0274178787
## [5,] -0.0002186054
## [6,]  0.9853449176
```

```
# projection matrix of reg y on X
P <- X2%*%solve(t(X2)%*%X2)%*%t(X2)
# residual maker of reg y on X: M= I - P
M <- diag(n2)-P
```

```r
# residuals
e <- M%*%y2
# varcov matrix b
s2 <- as.numeric(t(e)%*%e)/df
vb <- s2*solve(t(X2)%*%X2)
# std error of b
se2 <- sqrt(diag(vb))
```

The `lprice` coefficient from OLS is **-1.4418206**. This means that for every 1% increase in `price`, `qu` will decrease by about 1.44%. This is the price elasticity of demand.

```r
# create price-qu scatterplot
scatter <- ggplot() +
  geom_point(aes(x = my_data$price, y = my_data$qu)) +
  labs(x = "Car Price (in 1000 Euros)",
       y = "Car Sales Quantity",
       title = str_wrap(
         "Car Sales Quantity vs. Car Price",
         40)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
scatter
```



2. Omitted variable bias

Some factors that make you buy a car can also be correlated with higher prices (higher log prices). Please list a couple of such factors.

Other factors could be correlated with higher prices include safety features, quality/durability, and comfort.

Explain briefly why these omitted variables would cause the OLS estimate of equation (eq.1) to be biased for the true effect of an increase in log price (using the omitted variable bias approach) and why we cannot say that we are changing log price holding everything else constant in the OLS approach.

These omitted variables could cause us to be overestimating (in terms of magnitude) the true effect of `lprice` on `lqu`. In other words, the true β_3 is likely smaller in magnitude (less negative) since some of the downward pressure on `lqu` is being attributed to `lprice` instead of the omitted variables.

We cannot say that we are holding everything else constant because there are omitted variables that we are not controlling for.

3. Omit fuel

By the way, if we omit fuel from equation (eq.2) how does your OLS estimate of the log price change? What does this imply about the covariance between log price and fuel?

```
# declare variables
X2c <- cbind(1, my_data$lprice, my_data$year, my_data$weight, my_data$luxury)

# Find coefficient vector
b2c <- solve(t(X2c) %*% X2c) %*% t(X2c) %*% y2
b2c
```

```
##               [,1]
## [1,] 23.1508249820
## [2,] -1.3808902626
## [3,] -0.0046579168
## [4,] -0.0009874881
## [5,]  0.9964318660
```

The `lprice` coefficient is now smaller in magnitude, **-1.3808903**. This implies that there is a positive correlation between `lprice` and `fuel`.

We would expect higher sales of more fuel efficient cars and lower sales of more expensive cars. However, more fuel efficient cars also tend to be more expensive. Omitting fuel efficiency makes decrease in sales for more expensive cars look smaller.

Exercise 3

Specify an (eq.3) that is the linear model of the effect of cost factors common to all European countries (measured by the average log prices of a certain car type in other countries in Europe, not including the country in the dataset) denoted average_o1, with the same other regressors than equation (eq.2) as follows

$lprice_i = \alpha_0 + fuel_i\alpha_1 + year_i\alpha_2 + weight_i\alpha_3 + luxury_i\alpha_4 + laverage\_o1_i\alpha_5 + v_i$ (eq. 3)

Please estimate this model by OLS and interpret your results in terms of the coefficient of average_o1.

```
# Take log of average_o1
my_data$laverage_o1 <- log(my_data$average_o1)

# declare X and y variables
X3 <- cbind(1, my_data$fuel, my_data$year, my_data$weight, my_data$luxury, my_data$laverage_o1)
y3 <- my_data$lprice

# Find coefficient vector
b3 <- solve(t(X3) %*% X3) %*% t(X3) %*% y3
b3
```

```
##               [,1]
## [1,] 25.5281059242
## [2,] -0.0026079937
## [3,] -0.0126962776
## [4,]  0.0004580575
## [5,]  0.0053356366
## [6,]  0.7720019382
```

The coefficient of `laverage_o1` is **0.7720019**. This means that controlling for the other specified variables, `lprice` in this country will increase by about 0.77% when the average of prices in other countries increase by 1%.

Exercise 4

Specify an (eq.4) that is the linear model of the effect of the average of other countries' logprices on the log of quantity, with the same other regressors than equation (eq.2) as follows

$lqu_i = \delta_0 + fuel_i\delta_1 + year_i\delta_2 + weight_i\delta_3 + luxury_i\delta_4 + laverage\_o1_i\delta_5 + u_i$ (eq. 4)

This is called the reduced form model. Please estimate this model by OLS and interpret the results in terms of the coefficient of this reduced form regression of average_o1.

```
# declare X and y variables
X4 <- cbind(1, my_data$fuel, my_data$year, my_data$weight, my_data$luxury, my_data$laverage_o1)
y4 <- my_data$lqu

# Find coefficient vector
b4 <- solve(t(X4) %*% X4) %*% t(X4) %*% y4
b4
```

```
##               [,1]
## [1,] 26.176493454
## [2,] -0.146165397
## [3,] -0.006309850
## [4,] -0.001169946
## [5,]  0.936857361
## [6,] -0.927842632
```

The coefficient of `laverage_o1` in the reduced form model of `lqu` is **-0.9278426**. This is smaller in magnitude than the coefficient of `lprice` in Equation 2. This implies that the relationship between `lprice` and `lqu` in the same country is stronger than the relationship between the log of average prices in other countries, but that the direction of the relationship is the same and `average_o1` could serve as a proxy for `price` if we were unable to measure `price`.

Exercise 5

So far we have estimated the car price elasticity using ordinary least squares (OLS). Using price variation in other European countries, however, provides an opportunity to measure the price elasticity in our country of interest using instrumental variables (IV, 2SLS, two-stage least squares). Even though equation (eq.1) has a lot of regressors controlling for factors that could affect the log quantity of cars, we are worried that logprice in the country could be correlated with factors affecting log(quantity) that are not controlled for in the linear model in equation (eq.2), namely with ϵ_i.

1. Instrumental variables

Estimate equation (eq. 1 or 2???) by instrumental Variables using the variable LogAverageOther1 as an instrument for logprice. Please interpret the IV estimate of the logprice coefficient.

NOTE: Problem set instructions said use eq.1 but this appeared to be a typo. We have used eq.2 for all parts of Exercise 5.

```
# define variables X and y and instrument Z
X5 <- cbind(
  1, my_data$fuel, my_data$lprice, my_data$year, my_data$weight, my_data$luxury)
y5 <- my_data$lqu
Z5 <- cbind(
  1, my_data$fuel, my_data$laverage_o1, my_data$year, my_data$weight,
  my_data$luxury)

# IV coefficient estimates
b5_iv <- solve(t(Z5) %*% X5) %*% t(Z5) %*% y5
b5_iv
```

```
##               [,1]
## [1,] 56.8578477573
## [2,] -0.1492998550
## [3,] -1.2018656767
## [4,] -0.0215690704
## [5,] -0.0006194225
## [6,]  0.9432700790
```

Using `laverage_o1` as an instrumental variable, the coefficient of `lprice` is estimated to be **-1.2018657**. This means that after correcting for omitted variables, `lqu` decreases by about 1.2% when `lprice` increases by 1%. This is smaller in magnitude than the OLS estimate because the IV applies a correction to address omitted variable bias.

2. 2SLS

Estimate the first-stage regression and in the second stage substitute for logprice the predicted values of the first-stage regression. Please interpret the 2SLS estimate of the logprice coefficient.

First-stage: Regress x_k on Z

$lprice_i = \alpha_0 + fuel_i\alpha_1 + year_i\alpha_2 + weight_i\alpha_3 + luxury_i\alpha_4 + laverage\_o1_i\alpha_5 + v_i$

```
#first stage
Z5_fs <- cbind(
  1, my_data$fuel, my_data$laverage_o1, my_data$year, my_data$weight,
  my_data$luxury)
a_fs <- solve(t(Z5_fs) %*% Z5_fs) %*% t(Z5_fs) %*% my_data$lprice
lpricehat <- Z5_fs %*% a_fs
```

Second-stage: Regress `lqu` on exogeneous x's and $\hat{lprice}$

$lqu_i = \beta_0 + fuel_i\beta_1 + \hat{lprice}_i\beta_2 + year_i\beta_3 + weight_i\beta_4 + luxury_i\beta_5 + \epsilon_i$

```
#second stage 2SLS
X5_hat <- cbind(
  1, my_data$fuel, lpricehat, my_data$year, my_data$weight, my_data$luxury)

#E[xhat epsilon_i]=0
```

```r
b5_2sls <- solve(t(X5_hat) %*% X5_hat) %*% t(X5_hat) %*% y5
b5_2sls
```

```
##               [,1]
## [1,] 56.8578477528
## [2,] -0.1492998550
## [3,] -1.2018656766
## [4,] -0.0215690704
## [5,] -0.0006194225
## [6,]  0.9432700789
```

```r
# compute 2SLS standard errors
ehat <- y5 - X5%*%b5_2sls
# NOT e = wage - Xhat*b2sls
df5 <- nrow(X5) - ncol(X5)
S2_2sls <- as.numeric(t(ehat)%*%ehat)/df5

V_2sls <- solve(t(X5_hat)%*%X5_hat)*S2_2sls

se5_2sls <- sqrt(diag(V_2sls))
se5_2sls
```

```
## [1] 18.2541443126  0.0344762680  0.2745444076  0.0090466145  0.0005094664
## [6]  0.1390543259
```

The coefficient of `lprice` from 2SLS is the same as the `lprice` coefficient from IV, **-1.2018657**.

3. Control function

Estimate the first-stage regression and, in the second stage, use logprice (not the predicted education values of the first-stage regression as above) and also include the residuals from the first stage in the second stage, following a control function approach.

```r
# first stage
Z5_fs <- cbind(
  1, my_data$fuel, my_data$laverage_o1, my_data$year, my_data$weight,
  my_data$luxury)
a_fs <- solve(t(Z5_fs) %*% Z5_fs) %*% t(Z5_fs) %*% my_data$lprice
lpricehat <- Z5_fs %*% a_fs

# get first stage residuals
my_data$efs <- my_data$lprice - lpricehat

# add control function
Xcf <- cbind(
  1, my_data$fuel, my_data$lprice, my_data$efs, my_data$year, my_data$weight,
  my_data$luxury)
b5_cf<-solve(t(Xcf) %*% Xcf) %*% t(Xcf) %*% y5
b5_cf
```

```
##               [,1]
## [1,] 56.8578477581
## [2,] -0.1492998550
## [3,] -1.2018656767
## [4,] -0.4543582724
## [5,] -0.0215690704
```

```
## [6,] -0.0006194225
## [7,]  0.9432700790
```

4. Divide reduced-form regression coefficient by first-stage regression coefficient

The 2SLS coefficient can also be computed by dividing the reduced-form regression coefficient by the first-stage regression coefficient. Compute this ratio, as I did theoretically in the lecture.

```
# divide reduced-form regression coefficient by first-stage regression coefficient
coefratio5 <- b4[[6]] / a_fs[[3]]
```

The ratio of the reduced-form regression coefficient and the first-stage regression coefficient is **-1.2018657**.

5. Compare regression coefficients from different IV strategies

Confirm that the regression coefficients computed using the different IV strategies are basically equivalent, given that they all measure the same effect of logprice on log quantity in different instrumental variable fashions.

```
# compare IV and 2SLS coefficient
round(b5_iv[[3]], 6) == round(b5_2sls[[3]], 6)
```

```
## [1] TRUE
```

```
# compare CF and 2SLS coefficient
round(b5_cf[[3]], 6) == round(b5_2sls[[3]], 6)
```

```
## [1] TRUE
```

```
# compare ratio and 2SLS coefficient
round(coefratio5, 6) == round(b5_2sls[[3]], 6)
```

```
## [1] TRUE
```

6. Compare 2SLS to OLS

How does the 2SLS estimate of log price compare to the OLS estimate? Does the change make sense relative to factors you were worried about that would induce a bias, which could be in ϵ_i when you estimated equation (eq.1) by OLS? How do the standard errors compare, assuming homoskedasticity? Interpret these differences.

```
b2[[3]]
```

```
## [1] -1.441821
```

```
b5_2sls[[3]]
```

```
## [1] -1.201866
```

The 2SLS estimate of the `lprice` coefficient is slightly smaller in magnitude than the OLS estimate. This makes sense because it means that there are other factors that are correlated with high prices that were causing omitted variable bias, but price is still a very significant factor.

```
se2
```

```
## [1] 15.3219187656  0.0343595231  0.1885129871  0.0076250165  0.0003851746
## [6]  0.1345234330
```

```
se5_2sls
```

```
## [1] 18.2541443126  0.0344762680  0.2745444076  0.0090466145  0.0005094664
## [6]  0.1390543259
```

The standard errors for 2SLS coefficients are larger. This means that introducing `laverage_o1` instead of only using `lprice` increases the variance of the predictions and the standard error of the coefficients. This makes sense because we would not expect `laverage_o1` to be as correlated with `lqu` as `lprice` since it represents the prices in *other* countries.

7. Multiple instruments and Hausman test

Given that we get efficiency gains with more instruments, you consider now using both logAverageOther1 and logAverageOther2 (two average prices over a different set of European countries) as instruments for logprice. How would you test the null of the validity of both instruments? Perform the Hausman test of overidentifying restrictions assuming homoskedastic disturbances.

From Lecture 12: If homoskedasticity then $NR^2 \sim \chi^2_{Q_1}$ where $Q_1 = L_2 - G_2$ and R^2 is the R squared of the regression of the 2SLS residuals on the instruments that is, of the regression $e_{2SLS} = Z\rho + u$

```
# Take log of average_o2
my_data$laverage_o2 <- log(my_data$average_o2)

#first stage
Z5_fs <- cbind(
  1, my_data$fuel, my_data$laverage_o1, my_data$laverage_o2, my_data$year,
  my_data$weight, my_data$luxury)
a_fs <- solve(t(Z5_fs) %*% Z5_fs) %*% t(Z5_fs) %*% my_data$lprice
lpricehat <- Z5_fs %*% a_fs

#second stage 2SLS
X5_hat <- cbind(
  1, my_data$fuel, lpricehat, my_data$year, my_data$weight, my_data$luxury)

#E[xhat epsilon_i]=0
b5_2sls <- solve(t(X5_hat) %*% X5_hat) %*% t(X5_hat) %*% y5

# compute 2SLS standard errors
ehat <- y5 - X5%*%b5_2sls
# NOT e = wage - Xhat*b2sls
SSR <- t(ehat)%*%ehat
# construct demeaner
n <- length(y5)
i <- c(rep(1,n))
M0 <- diag(n)-i%*%t(i)*(1/n)
# demeaned y
M0y <- M0%*%y5
# total sum of squares
SST <- t(M0y)%*%M0y
# calculate R squared
Rsquared <- 1-(SSR/SST)
Rsquared
```

```
##            [,1]
## [1,] 0.2045674
```

```
# Hausman test
Hteststat <- pchisq(n*Rsquared, df=1, lower.tail=FALSE)
Hteststat
```

```
##                                                                                                        [,1]
```

```
## [1,] 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001532699
```

Since the R squared is **0.2045674** and the p-value of the χ^2 statistic for the Hausman test is **0**, we reject the null hypothesis. Therefore, we conclude that including both `laverage_o1` and `laverage_o2` overidentifies the number of restrictions.

Exercise 6: Limited Dependent Variable

1. Load and filter PBM data

Load the data pset_PBM_2024.dta

```
# Load data
pbm_data_raw <- read_dta(file.path(current_directory, "data", "pset4_PBM_2024.dta"))
```

Filter the respondents that chose noChoice==1 out of the data, keep only those that have noChoice==0. The remainder of the analysis is only done conditional on choosing one of the two options. Filter out the missing values for chosePBM, yourage, treat, relprice. You should get 262 observations now.

```
# Keep only noChoice==0
pbm_data <- filter(pbm_data_raw, noChoice==0)

# Filter out the missing values for chosePBM, yourage, treat, relprice.
pbm_data <- pbm_data %>% drop_na(chosePBM, yourage, treat, relprice)
```

2. Run linear probability model and correct standard errors for heteroskedasticity

Run the linear probability model of choosing the PBM option on yourage and a constant, treat, and relprice using the canned lm function in R and correct the standard errors for heteroskedasticity using the canned package also.

```
# Linear model
reg6 <- lm(chosePBM~yourage+treat+relprice, pbm_data)
summary(reg6)
```

```
##
## Call:
## lm(formula = chosePBM ~ yourage + treat + relprice, data = pbm_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44422 -0.27314 -0.05006  0.07199  0.76889
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  0.795512   0.197783   4.022            0.0000758 ***
## yourage     -0.007004   0.008351  -0.839             0.402423
## treat        0.654871   0.040036  16.357 < 0.0000000000000002 ***
## relprice    -0.272914   0.079603  -3.428             0.000706 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3235 on 258 degrees of freedom
## Multiple R-squared:  0.5237, Adjusted R-squared:  0.5182
## F-statistic: 94.56 on 3 and 258 DF,  p-value: < 0.00000000000000022
```

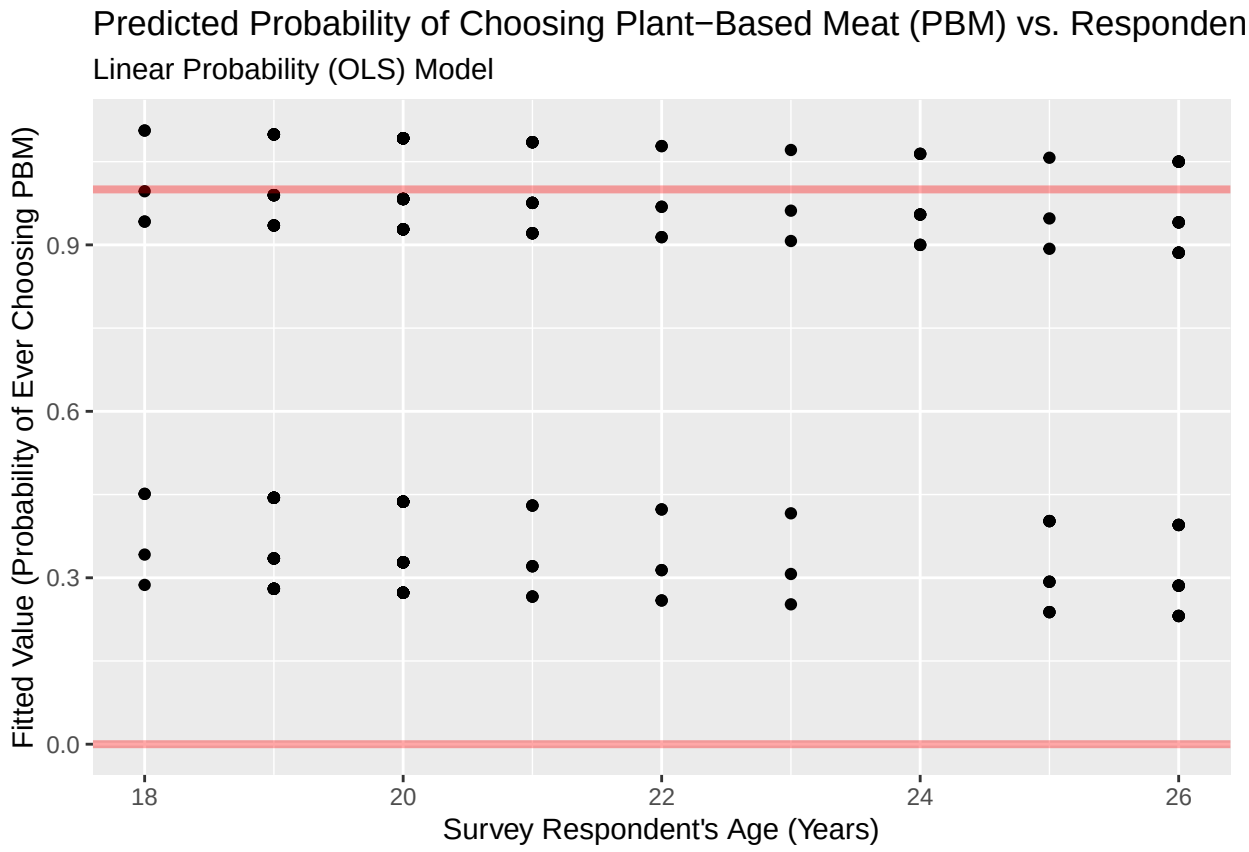
```r
# Correct standard errors for heteroskedasticity
# Use hc0, assume large sample
coeftest(reg6, vcov=hccm(reg6,type="hc0"))
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value              Pr(>|t|)
## (Intercept)  0.7955118  0.2203231  3.6107             0.0003669 ***
## yourage     -0.0070039  0.0092325 -0.7586             0.4487764
## treat        0.6548708  0.0414254 15.8084 < 0.00000000000000022 ***
## relprice    -0.2729136  0.0804100 -3.3940             0.0007971 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3. Plot predicted probabilities of choosing PBM against respondent's age

Like we did in Lecture 10, plot the predicted probabilities (on the y-axis) of choosing PBM against the respondent's age on the x-axis, and add a horizontal red line for y=0 and a horizontal red line for y=1.

```r
# Add fitted values to pbm_data
pbm_data$chosePBM_lmfit <- reg6$fitted.values

ggplot(pbm_data, aes(x = yourage, y = chosePBM_lmfit)) +
  # First add points, color determined by whether in or out of [0,1]
  geom_point() + # add points
  # add horizontal line at y=0
  geom_hline(yintercept=0, linewidth = 1.4, alpha = 0.35, color = "red") +
  # add horizontal line at y=1
  geom_hline(yintercept=1, linewidth = 1.4, alpha = 0.35, color = "red") +
  # generate labels
  labs(title = "Predicted Probability of Choosing Plant-Based Meat (PBM) vs. Respondent Age",
       subtitle = "Linear Probability (OLS) Model",
       x = "Survey Respondent's Age (Years)",
       y = "Fitted Value (Probability of Ever Choosing PBM)")
```



Is there a problem of predicting probabilities that fall out of the 0,1 range using the linear probability model?

Yes, there are predicted values for `chosePBM` that are greater than 1.

4. Logit Model

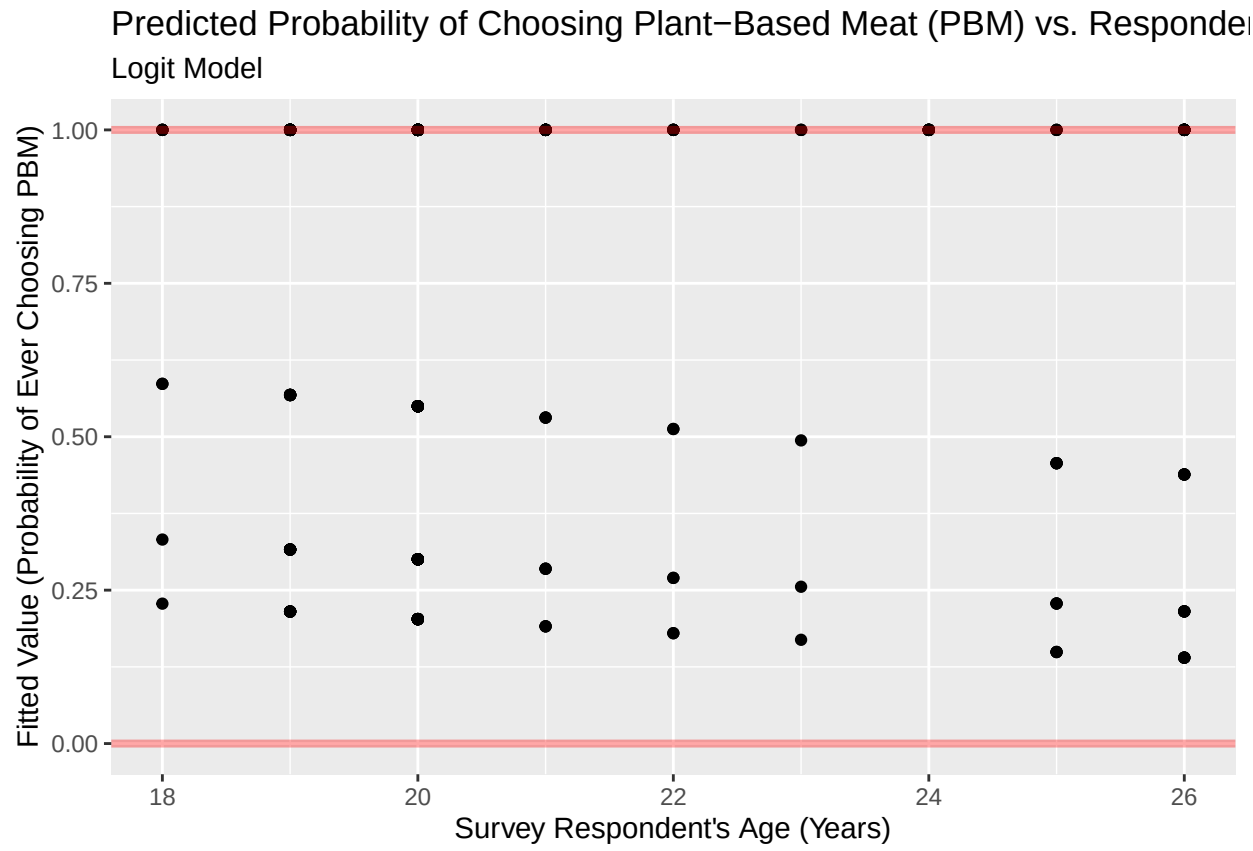
Estimate a logit model, using the R canned function (see lecture 10), of the probability of choosing PBM on the same covariates as in 2. Compute the fitted predicted probabilities and plot the scatter plot of the predicted probabilities on the y-axis and age on the horizontal axis.

```
# Create logit model
logit <- glm(chosePBM~yourage+treat+relprice, pbm_data, family=binomial(link="logit"))
summary(logit)
```

```
##
## Call:
## glm(formula = chosePBM ~ yourage + treat + relprice, family = binomial(link = "logit"),
##     data = pbm_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.77357    1.98666   1.899 0.057505 .
## yourage      -0.07444    0.08353  -0.891 0.372805
## treat        21.40387 1471.09693   0.015 0.988392
## relprice     -2.60801    0.78828  -3.308 0.000938 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 327.20  on 261  degrees of freedom
## Residual deviance: 149.36  on 258  degrees of freedom
## AIC: 157.36
##
## Number of Fisher Scoring iterations: 19
```

```r
# add in the logit fitted values
pbm_data <- mutate(pbm_data, chosePBM_logfit = logit$fitted.values)

# Create scatter plot
ggplot(pbm_data, aes(x = yourage, y = chosePBM_logfit)) +
  # First add points, color determined by whether in or out of [0,1]
  geom_point() + # add points
  # add horizontal line at y=0
  geom_hline(yintercept=0, linewidth = 1.4, alpha = 0.35, color = "red") +
  # add horizontal line at y=1
  geom_hline(yintercept=1, linewidth = 1.4, alpha = 0.35, color = "red") +
  # generate labels
  labs(title = "Predicted Probability of Choosing Plant-Based Meat (PBM) vs. Respondent Age",
       subtitle = "Logit Model",
       x = "Survey Respondent's Age (Years)",
       y = "Fitted Value (Probability of Ever Choosing PBM)")
```



Did the logit specification fix the problem in 3?

Yes, there are no longer fitted values that are greater than 1.

5. Logit marginal effects

Using the function of the margin computes the logit marginal effects. Then, create a data frame of the logit model's average values of the covariates. Compute the marginal effects at the mean values. Do these estimated marginal effects differ?

```
# calculate marginal effects at every observed value of X
# and average across the resulting effect estimates
margins <- margins(logit)
summary(margins)
```

```
##    factor     AME       SE       z      p     lower    upper
##  relprice -0.2544   0.0638 -3.9873 0.0001   -0.3795  -0.1294
##     treat  2.0879 143.4990  0.0145 0.9884 -279.1650 283.3407
##   yourage -0.0073   0.0081 -0.9007 0.3677   -0.0231   0.0085
```

```
# calculate the marginal effects of each variable at the means of the covariates

# create dataframe of mean data (i.e. one obs of X bar values)
meandata <- pbm_data %>%
  select(yourage, treat, relprice) %>%
  summarise_all(mean)

# compute marginal effects at mean values
meanmargins <- margins(logit, data = meandata)
summary(meanmargins)
```

```
##    factor     AME     SE       z      p   lower  upper
##  relprice -0.0001 0.0602 -0.0013 0.9990 -0.1180 0.1179
##     treat  0.0006 0.4519  0.0014 0.9989 -0.8851 0.8864
##   yourage -0.0000 0.0017 -0.0013 0.9989 -0.0034 0.0034
```

Yes, the average marginal effects and the marginal effects at means differ.

The average marginal effects are much greater in magnitude than the marginal effects at means. Even though the marginal effects at the mean values of the covariates are small, they are larger at other values. In particular, the marginal effect of `relprice` is statistically significant with a p-value <0.05.

6. Test significance of vegetarian and having had PBM before

Add being vegetarian dummy variable and having had PBM before as a covariate also and re-estimate the logit model.

```
# Add new variable, being vegetarian AND having had PBM before
pbm_data <- mutate(pbm_data, vegpbm = youvegetarian*pbm)

# Add new variable to logit model
logit2 <- glm(
  chosePBM~yourage+treat+relprice+vegpbm, pbm_data, family=binomial(link="logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logit2)
```

```
##
## Call:
## glm(formula = chosePBM ~ yourage + treat + relprice + vegpbm,
##     family = binomial(link = "logit"), data = pbm_data)
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.9803     3.3498   2.084  0.03718 *
## yourage       -0.1947     0.1466  -1.328  0.18423
## treat         23.3870  2212.9487   0.011  0.99157
## relprice      -4.1756     1.1880  -3.515  0.00044 ***
## vegpbm        22.6865  3218.4448   0.007  0.99438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 327.201  on 261  degrees of freedom
## Residual deviance:  81.458  on 257  degrees of freedom
## AIC: 91.458
##
## Number of Fisher Scoring iterations: 20
```

Test the null that the treatment does not matter in explaining the probability of choosing the PBM.

Unrestricted model:

$chosePBM_i = \frac{e_i^z}{1+e_i^z}$

where $z_i = \beta_1 + yourage_i\beta_2 + treat_i\beta_3 + relprice_i\beta_4 + vegpbm_i\beta_5 + \epsilon_i$

Null hypothesis: You should use the restricted model (no treatment).

Alternative hypothesis: You should use the unrestricted model.

```
# create model where treatment restricted
logit2_rtreat <- glm(
  chosePBM~yourage+relprice+vegpbm, pbm_data, family=binomial(link="logit"))
summary(logit2_rtreat)
```

```
##
## Call:
## glm(formula = chosePBM ~ yourage + relprice + vegpbm, family = binomial(link = "logit"),
##     data = pbm_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.72910    1.55961   3.032  0.00243 **
## yourage      -0.11910    0.06627  -1.797  0.07230 .
## relprice     -1.54761    0.58639  -2.639  0.00831 **
## vegpbm       18.26625  948.40435   0.019  0.98463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 327.20  on 261  degrees of freedom
## Residual deviance: 279.06  on 258  degrees of freedom
## AIC: 287.06
##
## Number of Fisher Scoring iterations: 17
```

```r
#get log likelihood value unrestricted
LLU <- logLik(logit2)
LLU
```

```
## 'log Lik.' -40.72881 (df=5)
```

```r
#get log likelihood value restricted
LLR <- logLik(logit2_rtreat)
LLR
```

```
## 'log Lik.' -139.5294 (df=4)
```

```r
# LR test
LRteststat <- -2 * (as.numeric(LLR)-as.numeric(LLU))
LRteststat
```

```
## [1] 197.6011
```

```r
# df = 5 - 4 = 1
LRpvalue <- pchisq(LRteststat, df = 1, lower.tail = FALSE)
LRpvalue
```

```
## [1] 0.00000000000000000000000000000000000000000000006971737
```

With a 1% significance level, we would reject the null hypothesis. This means we should use the unrestricted model. This suggests that treatment does matter.

Then test the null of whether being vegetarian and having had PBM before does not matter in explaining the probability of choosing PBM.

```r
# create model where vegpbm restricted
logit2_rvegpbm <- glm(
  chosePBM~yourage+treat+relprice, pbm_data, family=binomial(link="logit"))
summary(logit2_rvegpbm)
```

```
##
## Call:
## glm(formula = chosePBM ~ yourage + treat + relprice, family = binomial(link = "logit"),
##     data = pbm_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.77357    1.98666   1.899 0.057505 .
## yourage      -0.07444    0.08353  -0.891 0.372805
## treat        21.40387 1471.09693   0.015 0.988392
## relprice     -2.60801    0.78828  -3.308 0.000938 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 327.20  on 261  degrees of freedom
## Residual deviance: 149.36  on 258  degrees of freedom
## AIC: 157.36
##
## Number of Fisher Scoring iterations: 19
```

```r
#get log likelihood value unrestricted
LLU <- logLik(logit2)
```

```
LLU
```

```
## 'log Lik.' -40.72881 (df=5)
```

```
#get log likelihood value restricted
LLR <- logLik(logit2_rvegpbm)
LLR
```

```
## 'log Lik.' -74.68162 (df=4)
```

```
# LR test
LRteststat <- -2 * (as.numeric(LLR)-as.numeric(LLU))
LRteststat
```

```
## [1] 67.90563
```

```
# df = 5 - 4 = 1
LRpvalue <- pchisq(LRteststat, df = 1, lower.tail = FALSE)
LRpvalue
```

```
## [1] 0.00000000000000001715112
```

With a 1% significance level, we would reject the null hypothesis. This means we should use the unrestricted model. This suggests that being vegetarian and having had PBM before does matter.

7. Estimate log-likelihood function

Using the notes in lecture 10 write up the log-likelihood function for this case and estimate the parameters for the model with relprice and pbm and a constant as covariates. Compare the estimates of 7 with the canned function-based estimates. They should be the same.

```
# Define the negative log likelihood function
logl.logit <- function(theta,x,y){
  n <- nrow(x)
  y <- y
  x <- as.matrix(x)
  beta <- theta[1:ncol(x)]
  # Use the log-likelihood of the logit
  # where p is logit transformation of linear combination of predictors
  loglik <- sum(y*log(exp(x%*%beta)/(1+exp(x%*%beta))) +
                  (1-y)*log(1- exp(x%*%beta)/(1+exp(x%*%beta))))
  return(-loglik)
  }

y6ML <- pbm_data$chosePBM
X6ML <- cbind(1, pbm_data$relprice, pbm_data$pbm)

#estimate giving starting values
results6ML <- optim(
  c(-0.1,-0.1,-0.1),logl.logit,method="BFGS",hessian=T,x=X6ML,y=y6ML)

out <- list(
  beta=results6ML$par, vcov=solve(results6ML$hessian), ll=results6ML$value
  )
out
```

```
## $beta
## [1]  1.700731 -1.614912  1.326159
```

```
##
## $vcov
##             [,1]        [,2]        [,3]
## [1,]  0.47947318 -0.37786710 -0.03845193
## [2,] -0.37786710  0.33752532 -0.01606992
## [3,] -0.03845193 -0.01606992  0.08769388
##
## $ll
## [1] 149.4279
```

```
# compare with canned one
logit6ML <- glm(chosePBM~relprice+pbm, pbm_data, family=binomial(link="logit"))
summary(logit6ML)
```

```
##
## Call:
## glm(formula = chosePBM ~ relprice + pbm, family = binomial(link = "logit"),
##     data = pbm_data)
##
## Coefficients:
##             Estimate Std. Error z value   Pr(>|z|)
## (Intercept)   1.7007     0.6924   2.456    0.01404 *
## relprice     -1.6149     0.5810  -2.780    0.00544 **
## pbm           1.3261     0.2961   4.478 0.00000753 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 327.20  on 261  degrees of freedom
## Residual deviance: 298.86  on 259  degrees of freedom
## AIC: 304.86
##
## Number of Fisher Scoring iterations: 4
```

The estimates of β are the same.

8. Effect of having had PBM before

What is the marginal effect of having had pbm before on the probability of choosing PBM? What is the sample average of choosing PBM conditional of the sample that always chose one of the alternatives? What percentage of the mean is the estimated marginal effect? Answer: "Having had PBM before increases the probability of choosing PBM in the survey by XXX percent among those who choose one of the two alternatives" fill in the XXX based on the answer to the first questions in 8.

Assume that we use the same log-likelihood function used in Question 7.

Note, we filtered for noChoice==0 in Question 1.

```
# marginal effect of having had PBM before
marginsML <- margins(logit6ML)
marginsML
```

```
## Average marginal effects
```

```
## glm(formula = chosePBM ~ relprice + pbm, family = binomial(link = "logit"),     data = pbm_data)
```

```
##  relprice    pbm
```

```
##   -0.3114 0.2557
```

```
# sample average of chosing PBM
avgchosePBM <- mean(pbm_data$chosePBM)
avgchosePBM
```

```
## [1] 0.6832061
```

```
as.numeric(summary(marginsML)$AME[1])/avgchosePBM
```

```
## [1] 0.3742683
```

Having had PBM before increases the probability of choosing PBM in the survey by 37.4 percent among those who choose one of the two alternatives.

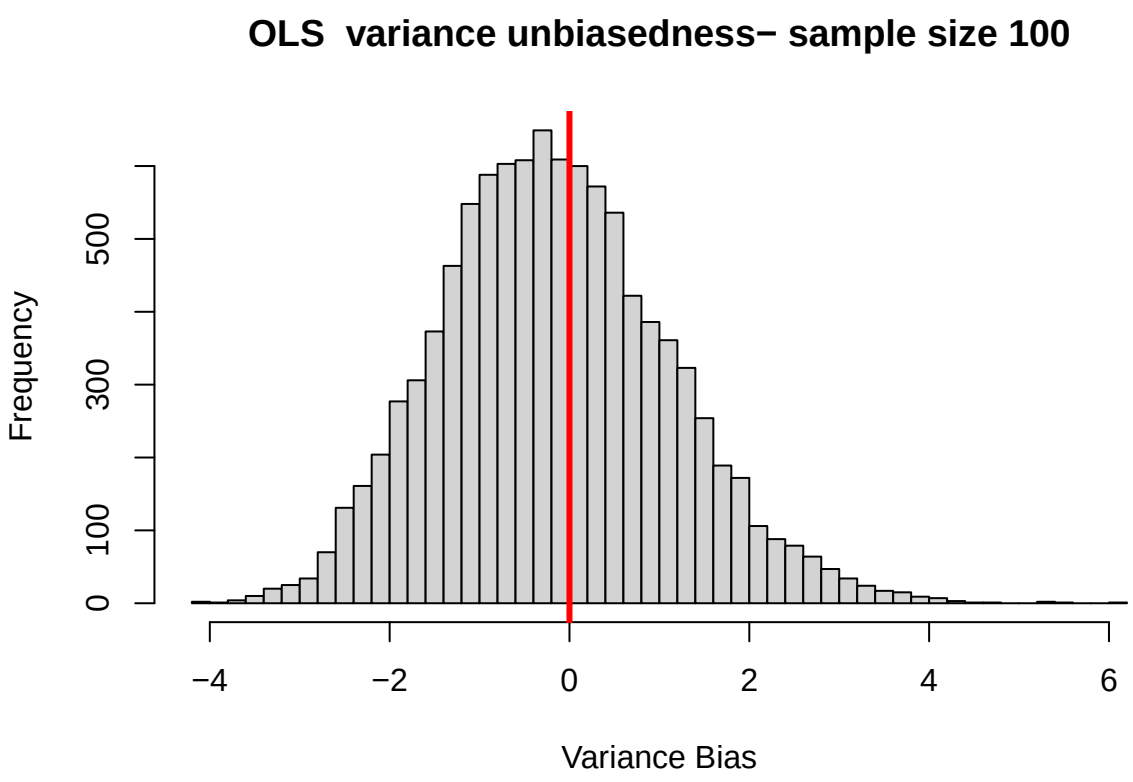
Exercise 7: Simulation

Let the linear model be given by $y_i = \beta_0 + x_i\beta_1 + \epsilon_i$ *where* x_i *is distributed as Normal with mean 0 and std 1, and where* ϵ_i *is distributed as normal, with mean 0 and standard error* $\sigma = 3$ *or variance* $\sigma^2 = 9$*. Let* Θ *be the vector of true parameters, where* $\Theta = \begin{bmatrix}\beta_0 & \beta_1 & \sigma^2\end{bmatrix}$*. Then, we define in R a vector of true parameters* `trueTheta <- c(1 , 5, 9)` *where the third* `trueTheta` *is the variance of* ϵ_i*, so std error is* `squareRoot( trueTheta[3,1] ) = 3` *to feed into the normal in the simulation.*

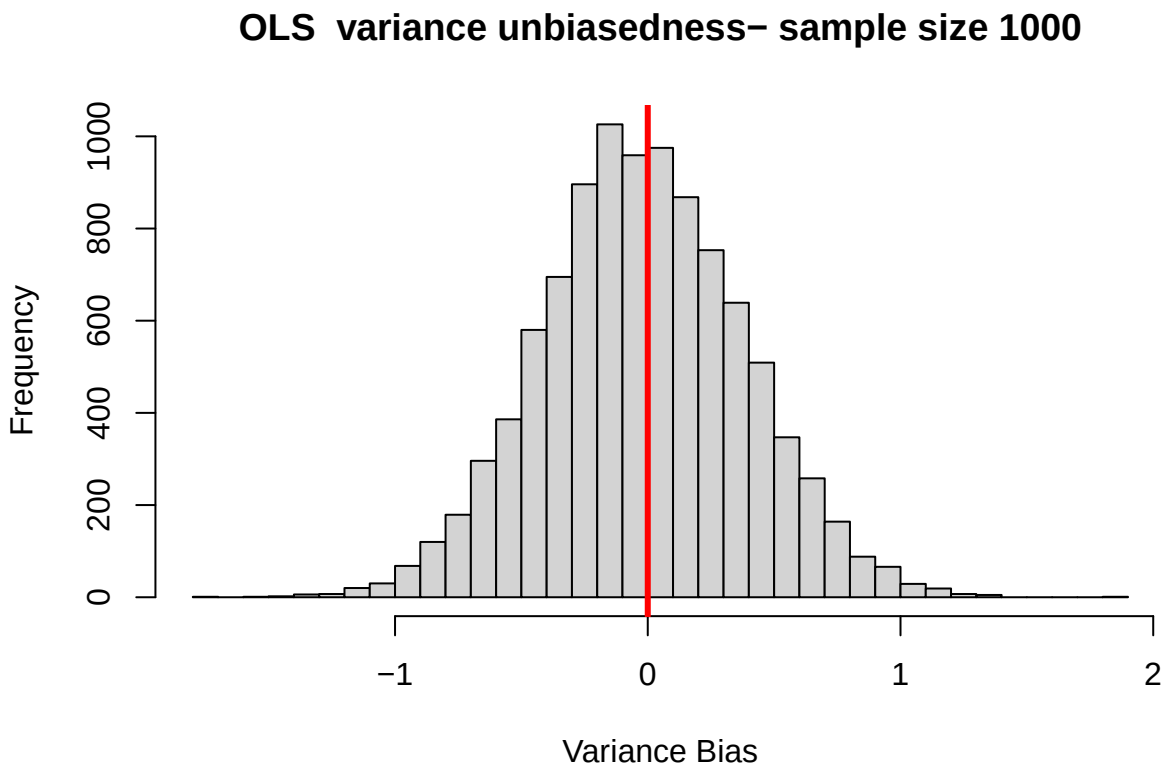
Please use R to create a simulation where you show what happens to the bias $\hat{\sigma}^2(N) - TrueTheta[3]$ *where* $\hat{\sigma}^2(N) = \frac{e'e}{N}$ *and e is the OLS residual, as the sample changes from* $N = 100$ *to* $N = 10000$ *and report the histogram of the simulated distribution of the variance estimator bias for both sample size based simulations. Use 10000 as the number of simulations you do, like in section 4's simulation.*

```
# A function to run the simulation
VarBiasSimulator <- function(simulationSize, sampleSize, trueTheta) {
  OLSVarBiasGenerator <- function(sampleSize, trueTheta) {
    # First generate x from N(0,1)
    x <- rnorm(n = sampleSize)
    # Now the error from N(0,1)
    e <- rnorm(n = sampleSize, sd=sqrt(trueTheta[3]))
    # Now combine trueTheta, x, and e to get y
    y <- trueTheta[1] + trueTheta[2] * x + e
    # Define the data matrix of independent vars.
    X <- cbind(1, x)
    # Force y to be a matrix
    y <- matrix(y, ncol = 1)
    # Calculate the OLS estimates
    theta.ols <- solve(t(X)%*%X) %*% t(X)%*%y
    # Calculate residuals
    ehat <- y - X%*%theta.ols
    # Calculate sigma-hat^2
    sigma2hat <- (t(ehat) %*% ehat)/sampleSize
    # Calculate difference between sigma-hat^2 and trueTheta[3]
    varbias <- as.numeric(sigma2hat-trueTheta[3]) %>% matrix(ncol = 1) %>%
      data.frame()
    # Set names
    names(varbias) <- c("varBias")
    # Return the bias
    return(varbias)
  }
```

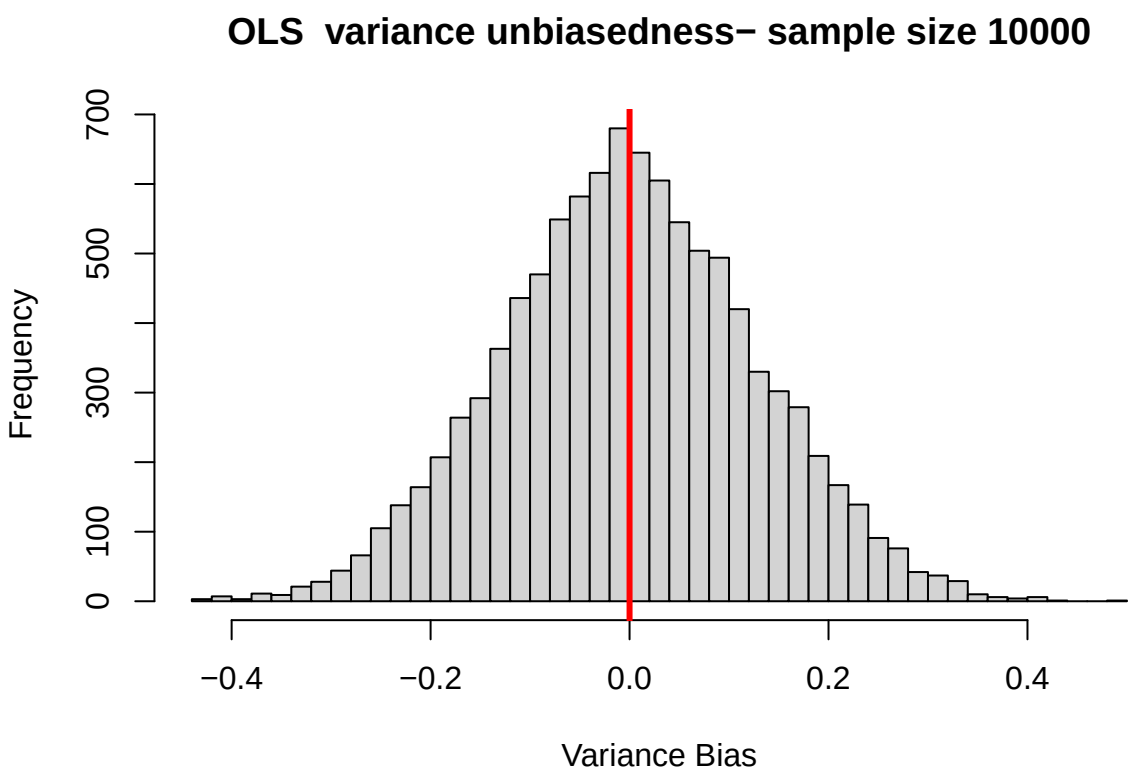
```r
  # Run OLSVarBiasGenerator simulationSize times with given parameters
  simulation.dt <- lapply(
    X = 1:simulationSize,
    FUN = function(i) OLSVarBiasGenerator(sampleSize, trueTheta)) %>%
    # Bind the rows together to output a nice data.frame
    bind_rows()

  # Return simulation.dt
  return(simulation.dt)
}

# Set seed
set.seed(12345)

# Run simulations
sim.dt100 <- VarBiasSimulator(simulationSize = 1e4, sampleSize = 100,
                              trueTheta = c(1, 5, 9))
sim.dt1000 <- VarBiasSimulator(simulationSize = 1e4, sampleSize = 1000,
                               trueTheta = c(1, 5, 9))
sim.dt10000 <- VarBiasSimulator(simulationSize = 1e4, sampleSize = 10000,
                                trueTheta = c(1, 5, 9))

# Plot histogram of variance bias for sample size = 100
hist(sim.dt100[,1], breaks=50,
     main = "OLS  variance unbiasedness- sample size 100",
     xlab = "Variance Bias")
# Emphasize zero line
abline(v = 0, col = "red", lwd = 3)
```




```
# Plot histogram of variance bias for sample size = 1000
hist(sim.dt1000[,1], breaks=50,
     main = "OLS  variance unbiasedness- sample size 1000",
     xlab = "Variance Bias")
# Emphasize zero line
abline(v = 0, col = "red", lwd = 3)
```




```
# Plot histogram of variance bias for sample size = 10000
hist(sim.dt10000[,1], breaks=50,
     main = "OLS  variance unbiasedness- sample size 10000",
     xlab = "Variance Bias")
# Emphasize zero line
abline(v = 0, col = "red", lwd = 3)
```


OLS variance unbiasedness− sample size 10000
Frequency
0
100
300
500
700
−0.4
−0.2
0.0
0.2
0.4
Variance Bias