ARE212_ps4_neri

Student:

2024-03-01

Install packages

```
### Define path/working directory and date
path<-'C:/Users/52554/Documents/GitHub/are_212/pset_4/code/'
path_data <- "C:/Users/52554/Documents/GitHub/are_212/pset_4/data/"
knitr::opts_chunk$set(setwd = path)
date <- Sys.Date()
print(date)
```

```
## [1] "2024-03-01"
```

```
### Function to install packages and call libraries
install <- function(packages){
  new.packages <- packages[!(packages %in% installed.packages()[, "Package"])]
  if (length(new.packages))
    install.packages(new.packages, dependencies = TRUE)
  sapply(packages, require, character.only = TRUE)
}
required.packages <- c("readr", "haven", "dplyr", "estimatr", "devtools",
                       "rdrobust", "rdd", "ggplot2", "tidyverse", "pacman", "psych",
                       "stargazer", "tinytex")
install(required.packages)
```

```
## Loading required package: readr

## Loading required package: haven

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
## Loading required package: estimatr

## Warning: package 'estimatr' was built under R version 4.3.2

## Loading required package: devtools

## Loading required package: usethis

##
## Attaching package: 'devtools'

## The following object is masked _by_ '.GlobalEnv':
##
##     install

## Loading required package: rdrobust

## Loading required package: rdd

## Loading required package: sandwich

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: AER

## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## Loading required package: survival

## Loading required package: Formula
## Loading required package: ggplot2

## Loading required package: tidyverse

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x car::recode()   masks dplyr::recode()
## x purrr::some()   masks car::some()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Loading required package: pacman

## Warning: package 'pacman' was built under R version 4.3.2

## Loading required package: psych
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
##
## The following object is masked from 'package:car':
##
##     logit
##
## Loading required package: stargazer
##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
##
## Loading required package: tinytex

##     readr     haven     dplyr  estimatr  devtools  rdrobust       rdd   ggplot2
##      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
## tidyverse    pacman     psych stargazer   tinytex
##      TRUE      TRUE      TRUE      TRUE      TRUE
```

```
p_load(dplyr, haven, readr, knitr, psych, ggplot2,stats4, stargazer, lmSupport, magrittr, qwraps2, Jmis
```

```
## Installing package into 'C:/Users/52554/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## Warning: package 'lmSupport' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
## Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib
##   no fue posible abrir la URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.3/PACKAGES'

## Warning in p_install(package, character.only = TRUE, ...):

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'lmSupport'

## Warning in p_load(dplyr, haven, readr, knitr, psych, ggplot2, stats4, stargazer, : Failed to install/
## lmSupport
```

```
# Define the function
b_ols <- function(data, y, X) {
  # Require the 'dplyr' package
  require(dplyr)

  # Select y variable data from 'data'
  y_data <- select(data, .dots = y)
  # Convert y_data to matrices
  y_data <- as.matrix(y_data)

  # Select X variable data from 'data'
  X_data <- select(data, .dots = X)
  # Convert X_data to matrices
  X_data <- as.matrix(X_data)

  # Add a constant inside the OLS fucntion, a column of ones to X_data
  X_data <- cbind(1, X_data)
  colnames(X_data)[1] <- "ones"

  # Calculate beta hat
  beta_hat <- solve(t(X_data) %*% X_data) %*% t(X_data) %*% y_data
  rownames(beta_hat) <- c("intercept", as.list(X))
  #Calculate degree of freedom
  n<-nrow(data)
  k<-ncol(X_data)
  dfree=n-k

    #Predicted values
  P <- (X_data %*% solve(t(X_data)%*%X_data) %*% t(X_data))
  Y_hat <- P%*%y_data

  # Change the name of 'ones' to 'intercept'
  rownames(beta_hat) <- c("intercept", X)
  e<-y_data-X_data%*%beta_hat
  S2_e<-t(e) %*% e
  S2_e<-as.numeric(S2_e/dfree)

  #Stand errors
  Vb_ols<-solve(t(X_data) %*% X_data) * S2_e
  se_ols<-sqrt(diag(Vb_ols))

  #T values
  t_value <-beta_hat/se_ols

  #Errores
  # SSR <- t(e) %*% e
  # SST <- t(y_data)%*%y_data
  # SSE <- t(beta_hat) %*% t(X_data)  %*% X_data  %*% beta_hat
  # R2<-1-(SSR/SST)

  return(list("beta_hat"=beta_hat,"df"=df, "e"=e,"S2_e"=S2_e, "Vb_ols"=Vb_ols,"se_ols"=se_ols,"X_data"=X
}
```

```
data_or <- read_dta(paste0(path_data,"pset4_2024.dta"))
ls(data_or)
```

```
##  [1] "average_o1"  "average_o2"  "brand"       "co"          "country"
##  [6] "country1"    "country2"    "country3"    "country4"    "country5"
## [11] "domestic"    "firm"        "fuel"        "height"      "horsepower"
## [16] "loc"         "luxury"      "ngdp"        "ngdpe"       "pop"
## [21] "pr"          "price"       "princ"       "qu"          "segment"
## [26] "type"        "weight"      "width"       "year"        "yearsquared"
```

```
data <- data_or %>%
  mutate(logquantity=log(qu)) %>%
  mutate(logprice=log(price)) %>%
  mutate(logaverage_o1=log(average_o1)) %>%
  mutate(logaverage_o2=log(average_o2))
```

Exercise 1.

###Estimate the model.

Equation 1

$$\text{logquantity}_i = \beta_1 + \beta_2 * \text{fuel}_i + \beta_3 * \text{logprice}_i + \beta_4 * \text{weight}_i + \epsilon_i$$

```
reg1<-b_ols(data = data, y = "logquantity", X = c("fuel", "logprice","weight"))
print(reg1$beta_hat, sep = "\n")
```

```
        .dots
```

intercept 13.0213 fuel -0.0920 logprice -0.7975 weight -0.0011

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

H_0: Homoscedasticity is present (the residuals are distributed with equal variance)

H_a: Heteroscedasticity is present (the residuals are not distributed with equal variance)

```
reglm_1<-lm(logquantity~fuel+logprice+weight,data)
bptest(reglm_1)
```

```
studentized Breusch-Pagan test
```

data: reglm_1 BP = 57, df = 3, p-value = 0.000000000003

```
#We see the p-value is less than 1%, so we reject the H0, this means the model have a problem of Hetero
```

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

```
e1<-reg1$e
X_data<-reg1$X_data
Xe<-cbind(e1, data$fuel*e1, data$logprice*e1, data$weight*e1)

Vb_whiteRobust<-solve(t(X_data) %*% X_data) %*% t(Xe) %*%  Xe %*% solve(t(X_data) %*% X_data)
seWhite<-sqrt(diag(Vb_whiteRobust))
se_ols <-reg1$se_ols

#Compare
seWhite
```

ones .dots1 .dots2 .dots3 0.22388 0.02324 0.14857 0.00025

```
se_ols
```

ones .dots1 .dots2 .dots3 0.22335 0.02562 0.14889 0.00024

```
#With package
reglm_1<-lm(logquantity~fuel+logprice+weight,data)
coeftest(reglm_1)
```

t test of coefficients:

```
        Estimate Std. Error t value             Pr(>|t|)
```

(Intercept) 13.02129 0.22335 58.30 < 0.0000000000000002 ***fuel -0.09195 0.02562 -3.59 0.00034*** logprice -0.79754 0.14889 -5.36 0.000000095 ***weight -0.00107 0.00024 -4.48 0.000008082*** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

```
coeftest(reglm_1,vcov=hccm(reglm_1,type="hc0"))
```

t test of coefficients:

```
        Estimate Std. Error t value             Pr(>|t|)
```

(Intercept) 13.02129 0.22387 58.16 < 0.0000000000000002 ***fuel -0.09195 0.02324 -3.96 0.000079023*** logprice -0.79754 0.14857 -5.37 0.000000089 ***weight -0.00107 0.00025 -4.30 0.000017618*** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

```
yum<-sandwich(reglm_1)
se_yum<-sqrt(diag(yum))
se_yum
```

(Intercept) fuel logprice weight 0.22388 0.02324 0.14857 0.00025

```
#We calculate the Robust Standard Errors, noticing all SE for relevant coefficients are smaller than th
```

1.c) Suppose that there is a model where a structural parameter of interest Y is defined as

$$\text{Y}_i = log(\beta_2 + 2)(\beta_3 + 3\beta_4)$$

### Using the OLS estimation results from eq. 1, calculate Y and its white standard error (hint: think Delta Method).

```
re1 <- lm(logquantity~fuel+logprice+weight, data)

beta_1=reg1$beta_hat[1]
beta_2=reg1$beta_hat[2]
beta_3=reg1$beta_hat[3]
beta_4=reg1$beta_hat[4]

SE_beta1=sqrt(diag(vcov(re1)))[1]
SE_beta2=sqrt(diag(vcov(re1)))[2]
SE_beta3=sqrt(diag(vcov(re1)))[3]
SE_beta4=sqrt(diag(vcov(re1)))[4]

parderiV_beta2 <- ((beta_3+3*beta_4)/(beta_2+2))
parderiV_beta3 <- log(beta_2+2)
parderiV_beta4 <- 3*log(beta_2+2)

y=log(beta_2+2)*(beta_3+3*beta_4)
SE_y = sqrt( (parderiV_beta2*SE_beta2)^2 + (parderiV_beta3*SE_beta3)^2 + (parderiV_beta4*SE_beta4)^2 )

y
```

[1] -0.52

```
SE_y
```

fuel 0.097

Exercise 2.

Equation 2

$$\text{log quantity}_i = \beta_0 + \beta_1 * \text{fuel}_i + \beta_2 * \text{lprice}_i + \beta_3 * \text{year}_i + \beta_4 * \text{weight}_i + \beta_5 * \text{luxury}_i + \epsilon_i$$

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

```r
reg2<-b_ols(data = data, y = "logquantity", X = c("fuel","year","weight","luxury","logprice"))
print( list("coeff",reg2$beta_hat, "t value",reg2$t_value))
```

[[1]] [1] “coeff”

[[2]] .dots intercept 68.77895 fuel -0.15249 year -0.02742 weight -0.00022 luxury 0.98534 logprice -1.44182

[[3]] [1] “t value”

[[4]] .dots intercept 4.49 fuel -4.44 year -3.60 weight -0.57 luxury 7.32 logprice -7.65

```r
# reglm_2<-lm(logquantity~fuel+logprice+year+weight+luxury,data)
# summary(reglm_2)
cor(data$fuel, data$logprice)
```

[1] 0.63 ### The coef for logprice is -1.44, because both are log, it implies an elasticity. When prices increases in aprox 1%, the quantity decreses in aprox 1.44%. This is in line with law demand, a higher price reduces quantity demanded (negative correlation).

2.2 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. 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 theOLS approach.

In my opinion, the most important variable correlated with higher prices are materials or quality of the product, for example an electric car could be more expensive and during gasoline shoks more people want to buy it. A second factor could be mileage, newer vehicles are more expensive. If these factors are affecting price and relevant for the equation, then we will have ommited variables and our coef for price would be biased.

2.3 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?

The coef for fuel is -0.15 and is statistacally significant, this means if we ommited it we would have biases in some coefficients. The correlation between both variables is 0.634 meaning that a better fuel eficiency is correlated with a higher price. If fuel was ommited the price coefficient would have a possitive bias.

3. Eq. 3 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.

Equation 3

$$\text{logprice}_i = \alpha_0 + \alpha_1 * \text{fuel}_i + \alpha_2 * \text{year}_i + \alpha_3 * \text{weight}_i + \alpha_4 * \text{luxury}_i + \alpha_5 * \text{logaverage\_o1}_i + \epsilon_i$$

```r
reg3<-b_ols(data = data, y = "logprice", X = c("fuel","year","weight","luxury","logaverage_o1"))
print( list(reg3$beta_hat, reg3$t_value))
```

```
[[1]] .dots intercept 25.52811 fuel -0.00261 year -0.01270 weight 0.00046 luxury 0.00534 logaverage_o1 0.77200

[[2]] .dots intercept 20.73 fuel -0.85 year -20.72 weight 12.86 luxury 0.43 logaverage_o1 40.90
```

```r
#reglm_3<-lm(logprice~fuel+year+weight+luxury+average_o1,data)
#summary(reglm_3)
#stargazer(reglm_3)
```

Notice both var are in log, then the coeff implies an elasticity, an increase of 1% in foreign prices is correlated with 0.77% increased in domestic prices Notice, the coef is statistically significant.

4. Specify an (eq.3) 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

Equation 4

$$\text{logquantity}_i = \delta_0 + \delta_1 * \text{fuel}_i + \delta_2 * \text{year}_i + \delta_3 * \text{weight}_i + \delta_4 * \text{luxury}_i + \delta_5 * \text{logaverage\_o1}_i + \epsilon_i$$

```r
reg4<-b_ols(data = data, y = "logquantity", X = c("fuel","year","weight","luxury","logaverage_o1"))
print( list(reg4$beta_hat, reg4$t_value))
```

```
[[1]] .dots intercept 26.1765 fuel -0.1462 year -0.0063 weight -0.0012 luxury 0.9369 logaverage_o1 -0.9278

[[2]] .dots intercept 1.87 fuel -4.21 year -0.91 weight -2.90 luxury 6.69 logaverage_o1 -4.33
```

```r
# reglm_4<-lm(logquantity~fuel+year+weight+luxury+average_o1,data)
# summary(reglm_4)
#stargazer(reglm_4)
```

The logaverage_o1 coeficiente indicates that an increase of 1% in foreign prices is correlated with a decrease of 0.92% in domestic qunatity demanded. This coef is statistically significant. Remember that we got a negative relation between domestic prices and quantity, then if foreign prices is negatively correlated with domestic prices, it is logic negatively correlated with quantity.

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 E_i.

5.1 Estimate eq.2 by Instrumental variables using the variable LogAverageOther1 as an instrument for logprice. Please interpret the IV estimate of the logprice coefficient.

Equation 2

$$\text{log quantity}_i = \beta_0 + \beta_1 * \text{fuel}_i + \beta_2 * \text{lprice}_i + \beta_3 * \text{year}_i + \beta_4 * \text{weight}_i + \beta_5 * \text{luxury}_i + \epsilon_i$$

```
Y <- data$logquantity
X <- cbind(1, data$fuel, data$logprice,      data$year, data$weight, data$luxury)
Z <- cbind(1, data$fuel, data$logaverage_o1, data$year, data$weight, data$luxury)


beta_IV<- solve(t(Z)%*%X)%*%t(Z)%*%Y
rownames(beta_IV) <- c("intercept","fuel","IV","year","weight","luxury")
beta_IV
```

```
          [,1]
```

intercept 56.85785 fuel -0.14930 IV -1.20187 year -0.02157 weight -0.00062 luxury 0.94327 ### 5.2 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.

```
reg521<-b_ols(data = data, y = "logprice",    X = c("fuel", "logaverage_o1","year","weight","luxury"))
reg521$beta_hat
```

```
              .dots
```

intercept 25.52811 fuel -0.00261 logaverage_o1 0.77200 year -0.01270 weight 0.00046 luxury 0.00534

```
data$Yhat51=reg521$Y_hat

reg522<-b_ols(data = data, y = "logquantity", X = c("fuel", "Yhat51","year","weight","luxury"))
reg522$beta_hat
```

```
          .dots
```

intercept 56.85785 fuel -0.14930 Yhat51 -1.20187 year -0.02157 weight -0.00062 luxury 0.94327

5.3 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.

```
reg531<-b_ols(data = data, y = "logprice", X = c("fuel", "logaverage_o1","year","weight","luxury"))
reg531$beta_hat
```

```
              .dots
```

intercept 25.52811 fuel -0.00261 logaverage_o1 0.77200 year -0.01270 weight 0.00046 luxury 0.00534

```
data$e531 <- reg531$e

reg532<-b_ols(data = data, y = "logquantity", X = c("fuel","logprice", "e531","year","weight","luxury")
reg532$beta_hat
```

```
          .dots
```

intercept 56.85785 fuel -0.14930 logprice -1.20187 e531 -0.45436 year -0.02157 weight -0.00062 luxury 0.94327

5.4 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.

```
reg4$beta_hat[[5]]
```

[1] 0.94

```
reg521$beta_hat[[3]]
```

[1] 0.77

```
coef_m <- reg4$beta_hat[[5]]/reg521$beta_hat[[3]]
print(coef_m, round(3))
```

[1] 1.21

##5.5 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. ### They are all the same

##5.6 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 Ei when you estimated equation (eq.1) by OLS? How do the standard errors compare, assuming homoskedasticity? Interpret these differences.

##5.7 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

```
#Lecture 12
reg551<-b_ols(data = data, y = "logprice", X = c("fuel", "logaverage_o1","logaverage_o2","year","weight
data$Yhat551=reg551$Y_hat

reg522<-b_ols(data = data, y = "logquantity", X = c("fuel", "Yhat551","year","weight","luxury"))
reg522$beta_hat
```

```
          .dots
```

intercept 76.023683 fuel -0.154421 Yhat551 -1.587647 year -0.030972 weight 0.000025 luxury 1.010915

6.1 Load the data pset__PBM__2024.dta

6.2 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.

```
data_or <- read_dta(paste0(path_data, "pset4_PBM_2024.dta"))
ls(data_or)
```

[1] “choseMeat” “chosePBM” “dprice” “noChoice”
[5] “pbm” “pmeat” “pPBM” “relprice”
[9] “treat” “type” “yourage” “youvegetarian”

```
data <- data_or %>%
  filter(noChoice==0) %>%
  filter(!is.na(chosePBM)) %>%
  filter(!is.na(yourage)) %>%
  filter(!is.na(treat)) %>%
  filter(!is.na(relprice))


reg62<-lm(chosePBM~yourage+treat+relprice,data)
summary(reg62)
```

Call: lm(formula = chosePBM ~ yourage + treat + relprice, data = data)

Residuals: Min 1Q Median 3Q Max -0.4442 -0.2731 -0.0501 0.0720 0.7689

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.79551 0.19778 4.02 0.000076 ***yourage -0.00700 0.00835 -0.84 0.40242
treat 0.65487 0.04004 16.36 < 0.0000000000000002*** relprice -0.27291 0.07960 -3.43 0.00071 *** —
Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.32 on 258 degrees of freedom Multiple R-squared: 0.524, Adjusted R-squared: 0.518 F-statistic: 94.6 on 3 and 258 DF, p-value: <0.0000000000000002

```
yhat_reg62 <-reg62$fitted.values

coeftest(reg62, vcov=hccm(reg62,type="hc0"))
```

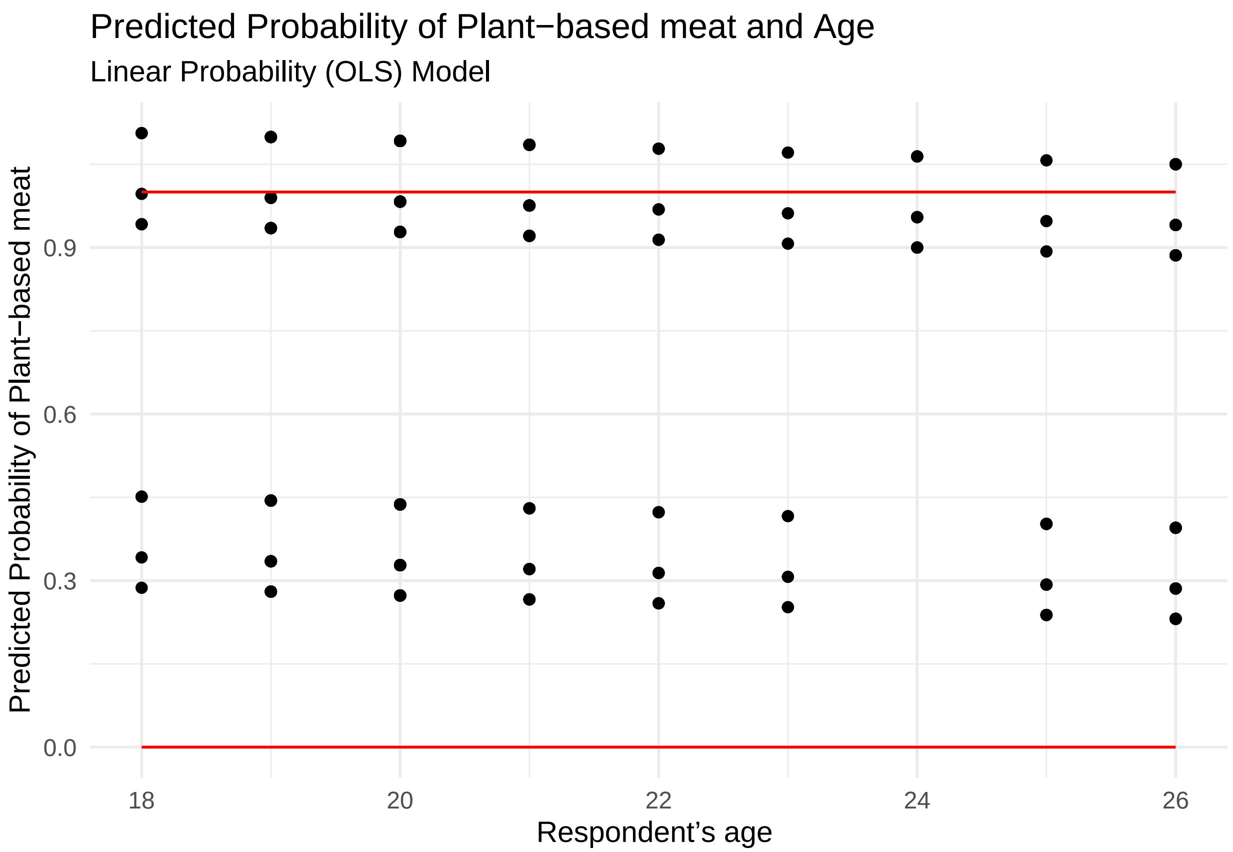
t test of coefficients:

```
        Estimate Std. Error t value             Pr(>|t|)
```

(Intercept) 0.79551 0.22032 3.61 0.00037 ***yourage -0.00700 0.00923 -0.76 0.44878
treat 0.65487 0.04143 15.81 < 0.0000000000000002*** relprice -0.27291 0.08041 -3.39 0.00080 *** —
Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

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

```
ggplot(data, aes(x=yourage, y=yhat_reg62)) +
  geom_point(color = "black", fill = "grey") + theme_minimal() +
  geom_line(aes(y = 1), col = "red") +
  geom_line(aes(y = 0), col = "red") +
  labs(x="Respondent's age ", y="Predicted Probability of Plant-based meat", subtitle="Linear Probabili
```



###The problem of LM model is probabilities bigger than 1, which do not have sense.

###6.4 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. Did the logit specification fix the problem in 3?

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

Call: glm(formula = chosePBM ~ yourage + treat + relprice, family = binomial(link = “logit”), data = data)

Coefficients: Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.7736 1.9867 1.90 0.05751 .
yourage -0.0744 0.0835 -0.89 0.37281
treat 21.4039 1471.0969 0.01 0.98839
relprice -2.6080 0.7883 -3.31 0.00094 *** — Signif. codes: 0 ‘*’ ***0.001*** ’*’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

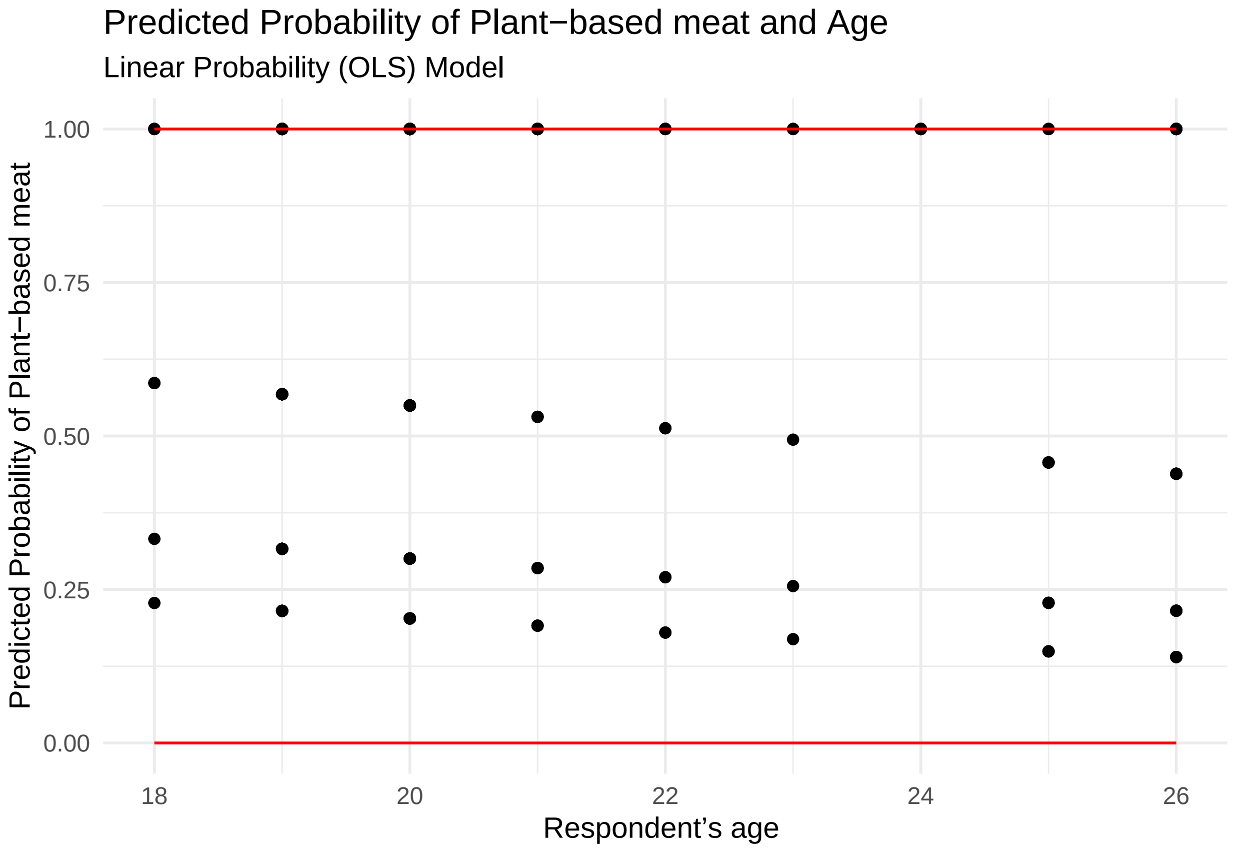
(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 327.20  on 261  degrees of freedom
```

Residual deviance: 149.36 on 258 degrees of freedom AIC: 157.4

Number of Fisher Scoring iterations: 19

```
logit_predic <-logit$fitted.values

ggplot(data, aes(x=yourage, y=logit_predic)) +
  geom_point(color = "black", fill = "grey") + theme_minimal() +
  geom_line(aes(y = 1), col = "red") +
  geom_line(aes(y = 0), col = "red") +
  labs(x="Respondent’s age ", y="Predicted Probability of Plant-based meat", subtitle="Linear Probabili
```



6.5 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?

```
# create dataframe of mean data (i.e. one obs of X bar values)
# replicate R's margins, dydx(*) command:
margins <- margins(logit)
meandata <- data %>%
  select(yourage,treat,relprice) %>%
  summarise_all(mean)
margins_atmean <- margins(logit, data = meandata)

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

```
summary(margins_atmean)
```

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 ### Both are different. Margin at mean is smaller than marginal effects.

6.6 Add being vegetarian dummy variable and having had PBM before as a co variate also and re-estimate the logit model. Test the null that the treatment does not matter in explaining the probability of choosing the PBM. Then test the null of whether being vegetarian and having had PBM before does not matter in explaining the probability of choosing PBM.

```
data <- data  %>%
  mutate(vegpbm=youvegetarian*pbm)

logit <- glm(chosePBM~yourage+treat+relprice+vegpbm, data, family = binomial(link = "logit"))
summary(logit)
```

Call: glm(formula = chosePBM ~ yourage + treat + relprice + vegpbm, family = binomial(link = “logit”), data = data)

Coefficients: Estimate Std. Error z value Pr(>|z|)
(Intercept) 6.980 3.350 2.08 0.03718 *
yourage -0.195 0.147 -1.33 0.18423
treat 23.387 2212.949 0.01 0.99157
relprice -4.176 1.188 -3.51 0.00044 *** vegpbm 22.687 3218.445 0.01 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: 91.46

Number of Fisher Scoring iterations: 20

```
logit_predic <-logit$fitted.values
```

###We can know if some variable is relevant for the model with z value. treat and vegpbm variables are not relevant for being likely of choosuing PBM.

6.7 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, Author : Sofia Villas-Boas
  logl.logit <- function(theta,x,y){
    n<-nrow(x)
    y <- y
    x <- as.matrix(x)
    beta <- theta[1:ncol(x)]
```

```r
    # Use the log-likelihood of the logit, where p is
    # defined as the logit transformation of a linear combination
    # of predictors, according to logit(p)=e( X beta)/(1+e(X beta)) see slide 41 of powerpoint
    loglik <-sum(y*log(exp(x%*%beta)/(1+exp(x%*%beta))) + (1-y)*log(1- exp(x%*%beta)/(1+exp(x%*%beta))))
    return(-loglik)
  }

y <- data$chosePBM
x <- cbind(1, data$relprice, data$pbm)

#estimate giving starting values
resultsML<-optim(c(0,0,0),logl.logit,method="BFGS",hessian=T,x=x,y=y)
resultsML
```

$par [1] 1.7 -1.6 1.3

$value [1] 149

$counts function gradient 23 7

$convergence [1] 0

$message NULL

$hessian [,1] [,2] [,3] [1,] 51 58 33 [2,] 58 70 38 [3,] 33 38 33

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

logit_canned <- glm(chosePBM~relprice+pbm, data, family = binomial(link = "logit"))

out
```

$beta [1] 1.7 -1.6 1.3

$vcov [,1] [,2] [,3] [1,] 0.479 -0.378 -0.038 [2,] -0.378 0.338 -0.016 [3,] -0.038 -0.016 0.088

$ll [1] 149

```r
summary(logit_canned)
```

Call: glm(formula = chosePBM ~ relprice + pbm, family = binomial(link = “logit”), data = data)

Coefficients: Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.701 0.692 2.46 0.0140 *
relprice -1.615 0.581 -2.78 0.0054 ** pbm 1.326 0.296 4.48 0.0000075 *** — 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: 304.9

Number of Fisher Scoring iterations: 4 ###We notice both betas are the same.

6.8 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

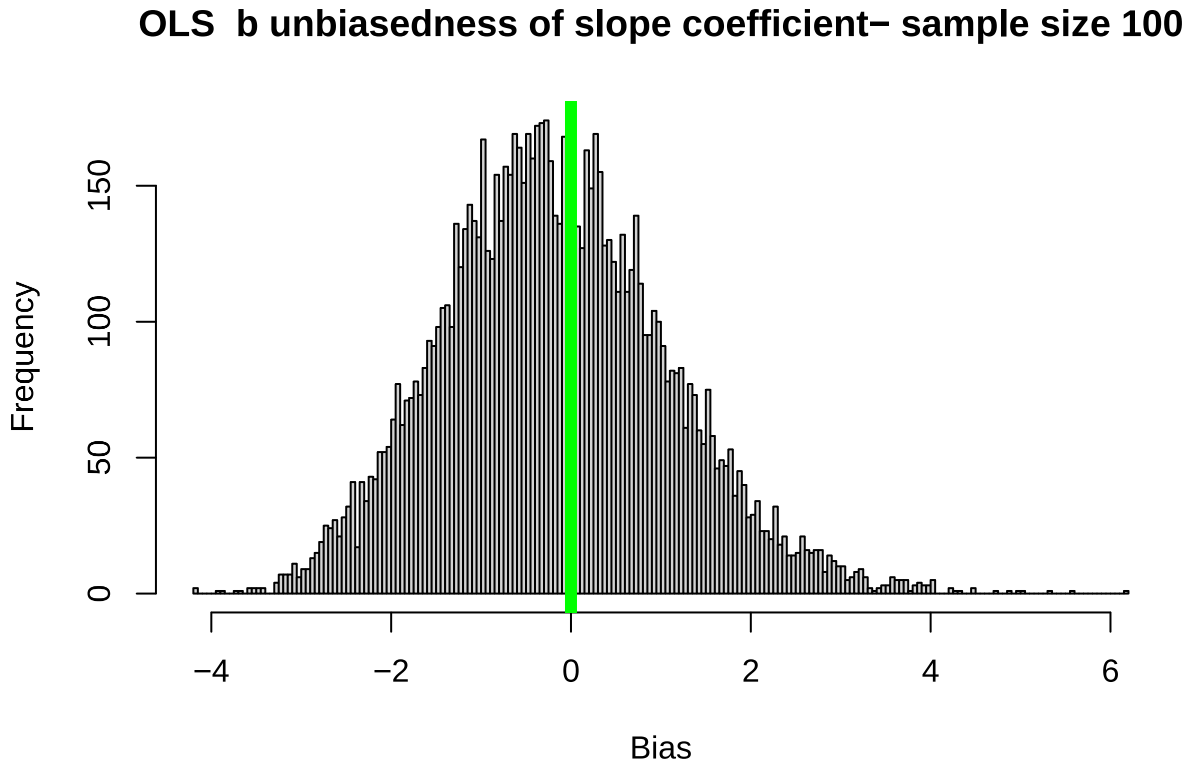
###Answer: "Having had PBM before increases the probability of choosing PBM in the survey by XXX percent among those who choose one of the two alternatives"

Exercise 7 – Simulation. Let the linear model be given by

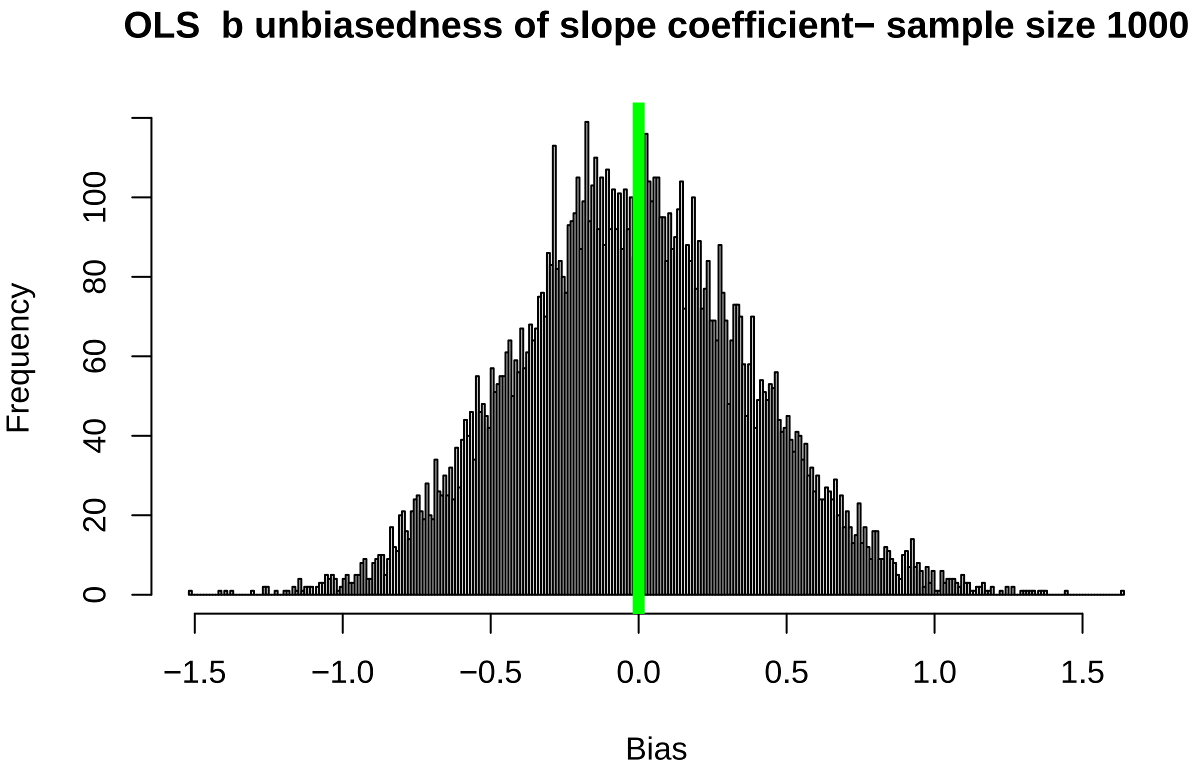
$$\mathrm{y}_i = \beta_0 + \beta_1 * \mathrm{x}_i + \epsilon_i$$

```r
BiasSimulator <- function(simulationSize, sampleSize, trueBeta) {

   OLSBiasGenerator <- function(sampleSize, trueBeta) {

    #trueBeta <- c(1 , 5, 9)
    x <- rnorm(n = sampleSize, mean = 0, sd=1)
    e <- rnorm(n = sampleSize, mean = 0, sd=3)
    y <- trueBeta[1] + trueBeta[2] * x + e
    X <- cbind(1, x)
    y <- matrix(y, ncol = 1) # Force y to be a matrix

    # Calculate the OLS estimates
    b.ols <- solve(t(X) %*% X) %*% t(X) %*% y
    b.ols %>% as.vector()
    e_hat <- y - X%*%b.ols
    sigma_hat <- (t(e_hat) %*% e_hat)/sampleSize

    biasBeta <- (sigma_hat-trueBeta[3]) %>% matrix(ncol = 1) %>%  data.frame()
      # Set names
   #names(biasBeta) <- c("interceptBias", "regressorBias")
      # Return the bias
    return(biasBeta)
  }
    # Run OLSBiasGenerator simulationSize times with given parameters
    simulation.dt <- lapply(
      X = 1:simulationSize,
      FUN = function(i) OLSBiasGenerator(sampleSize, trueBeta)) %>%
      # Bind the rows together to output a nice data.frame
      bind_rows()

    # Return simulation.dt
    return(simulation.dt)
    #simulation dt is a Simulationsize by number of true parametres matrix
  }

set.seed(1234)
sim.dt100 <- BiasSimulator(simulationSize = 1e4, sampleSize = 100, trueBeta = c(1 , 5, 9))
sim.dt1000 <- BiasSimulator(simulationSize = 1e4, sampleSize = 1000, trueBeta = c(1 , 5, 9))
sim.dt10000 <- BiasSimulator(simulationSize = 1e4, sampleSize = 10000, trueBeta = c(1 , 5, 9))
```

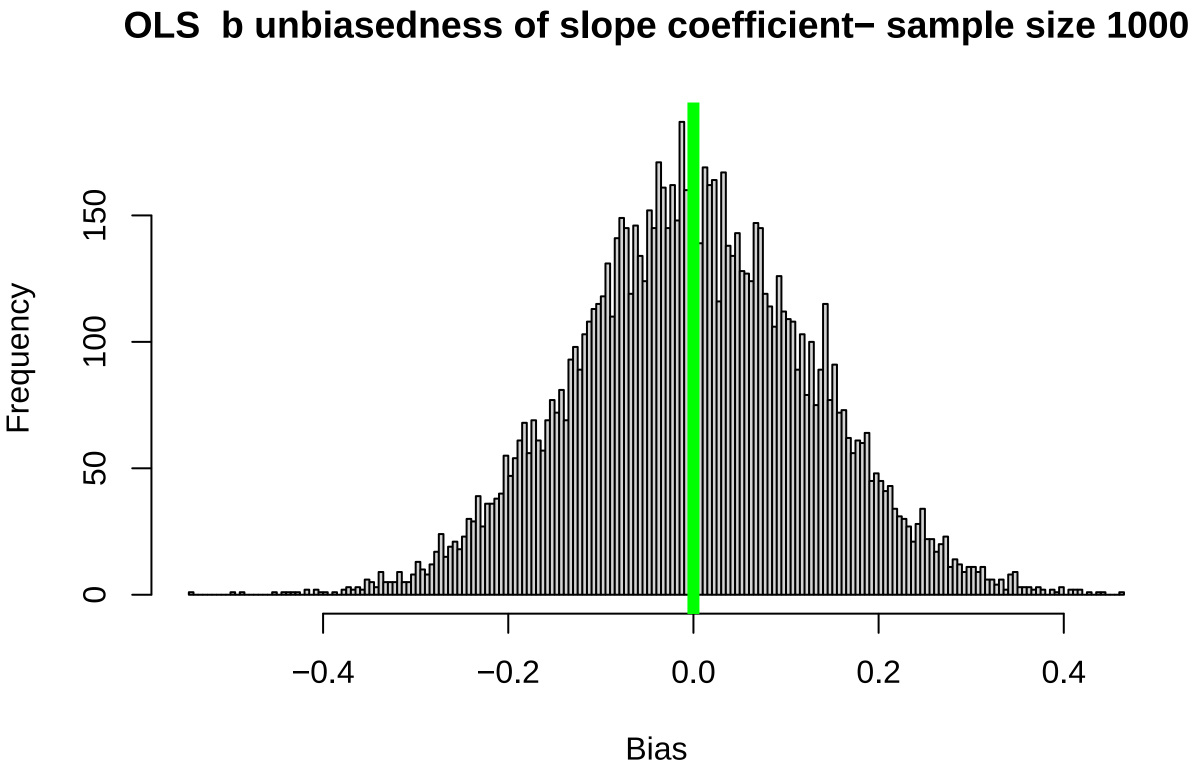
```
hist(sim.dt100[,1], breaks = 300, main = "OLS  b unbiasedness of slope coefficient- sample size 100",
       xlab = "Bias")
abline(v = 0, col = "green", lwd = 6)
```



```
hist(sim.dt1000[,1], breaks = 300, main = "OLS  b unbiasedness of slope coefficient- sample size 1000",
       xlab = "Bias")
abline(v = 0, col = "green", lwd = 6)
```



```
hist(sim.dt10000[,1], breaks = 300, main = "OLS  b unbiasedness of slope coefficient- sample size 1000"
       xlab = "Bias")
abline(v = 0, col = "green", lwd = 6)
```


OLS b unbiasedness of slope coefficient− sample size 1000
Frequency
0
50
100
150
−0.4
−0.2
0.0
0.2
0.4
Bias