Geneva Graduate Institute, Econometrics I

Problem Set 6 Solutions

Francesco Casalena

Fall 2024

Problem 1

The dataset `dat_SalesCustomers.csv` contains data on sales of shopping malls in Instanbul. It includes the following variables: *invoice_no* (identifier of transaction/invoice), *customer_id* (identifier of customer), *category* (type of goods sold), *price* (in TRY, Turkish Lira), *invoice_date*, *shopping_mall*, *gender*, *age* and *payment_method* (cash- vs. credit-card- vs. debit-card-payment).

You are interested in shedding light on the determinants of cash- vs card-payment. For this purpose, you set up a probit model:

$$y_i^* = x_i'\beta + u_i \ , \quad u_i|x_i \sim N(0,1) \ , \tag{1}$$

whereby we observe $y_i = \mathbf{1}\{y_i^* \geqslant 0\}$, a dummy variable for cash payment. Recall that the Maximum Likelihood (ML) estimator for β solves

$$\hat{\beta} = \arg\min_{\beta} Q_n(\beta; Z_n) \quad \text{for} \quad Q_n(\beta; Z_n) = -\frac{1}{n}\ell(\beta; Z_n) \ , \tag{2}$$

where

$$\ell(\beta; Z_n) = \sum_{i=1}^{n} y_i \log(\Phi(x_i'\beta)) + (1 - y_i)\log(\Phi(-x_i'\beta))$$

is the log-likelihood and $Z_n = \{y_i, x_i\}_{i=1}^n$ comprises all of the data you have available (outcome-variables and covariates for the n observations in your sample).

1. Are there missing values in your data? Delete all observations with a missing value in the variables *category*, *price*, *gender*, *age* and *payment_method*. How many observations do you have left?

Solution:

```r
library(tidyverse)

rm(list = ls())
data <- read.csv("dat_SalesCustomers.csv", header = TRUE)

colSums(is.na(data))
```

```
##     invoice_no    customer_id       category       quantity          price
##              0              0              0              0              0
##   invoice_date  shopping_mall         gender            age payment_method
##              0              0              0            119              0
```

```r
data <- data %>% drop_na()

nrow(data)
```

```
## [1] 99338
```

There are 119 observations missing for age. After we delete these, we are left with a dataset with 99338 observations.

2. Based on the variable *payment_method*, generate a dummy variable for cash payment and call it *paid_in_cash*. Also, based on *gender*, create a dummy for males, *male*. What fraction of transactions were carried out in cash? What fraction of the overall sales (in TRY) were carried out in cash?

Solution:

```r
# Generate dummies
data <- data %>%
  mutate(male = as.numeric(gender == "Male"),
```

```
       paid_in_cash = as.numeric(payment_method == "Cash"))

# Relative frequencies of cash transactions
table(data$paid_in_cash) %>% prop.table()
```

```
##
##         0         1
## 0.5530713 0.4469287
```

```
# Relative frequencies of sales paid in cash
data %>%
  group_by(payment_method) %>%
  summarise(total_sales = sum(price)) %>%
  mutate(
    proportion = total_sales / sum(total_sales))
```

```
## # A tibble: 3 x 3
##   payment_method total_sales proportion
##   <chr>                <dbl>      <dbl>
## 1 Cash             30669094.      0.448
## 2 Credit Card      24027939.      0.351
## 3 Debit Card       13776341.      0.201
```

44.7% of the transactions were settled in cash, 44.8% of the sales were paid in cash.

3. Based on the variable *category*, create a dummy for each of the following four categories: i) clothes and shoes, ii) cosmetics, iii) food, iv) technology. In this way, we divide the categories into five groups, whereby the fifth is made up by the rest, i.e. goods that do not belong to either of the four categories. How are the transactions split across these five categories? How are the sales split across these five categories?

Solution:

```
data <- data[1:1000,]

# Sum up categories
data$category2 <- "rest"
data$category2[data$category == "Clothing" | data$category == "Shoes"] <- "clothingshoes"
data$category2[data$category == "Cosmetics"] <- "cosmetics"
data$category2[data$category == "Food & Beverage"] <- "food"
data$category2[data$category == "Technology"] <- "technology"

# Generate dummies
data <- data %>%
  mutate(clothingshoes = as.numeric(category2 == "clothingshoes"),
         cosmetics = as.numeric(category2 == "cosmetics"),
         food = as.numeric(category2 == "food"),
         technology = as.numeric(category2 == "technology"))

# Relative frequencies of transactions by goods
table(data$category2) %>% prop.table()
```

```
##
## clothingshoes     cosmetics          food          rest    technology
##         0.438         0.148         0.140         0.224         0.050
```

```
# Relative frequencies of sales by goods
(mSalesProportions <- data %>%
  group_by(category2) %>%
  summarise(total_sales = sum(price)) %>%
  mutate(
    proportion = total_sales / sum(total_sales)))
```

```
## # A tibble: 5 x 3
##   category2     total_sales proportion
##   <chr>               <dbl>      <dbl>
## 1 clothingshoes     474429.    0.706
## 2 cosmetics          18297.    0.0272
## 3 food                2123.    0.00316
## 4 rest               16718.    0.0249
## 5 technology        160650     0.239
```

43.8% of transactions (70.6% of sales) regarded clothing and shoes, 14.8% of transactions (2.7% of sales) regarded cosmetics, 14% of transactions (0.3% of sales) regarded food, 5% of transactions (23.9% of sales) regarded technology and 22.4% of transactions (2.5% of sales) regarded the rest.

4. Taking *paid_in_cash* as your outcome variable y_i and *price*, *male*, *age* and aand the four category-dummies as your covariates x_i, use a numerical optimization-command from the software of your choice to solve the optimization problem in (2) and obtain $\hat{\beta}$ for your sample.[1]

Hint: instead of computing first $\Phi(x)$ using a software-command for the cdf of a $N(0,1)$ RV (`pnorm(x)` in R) and then taking logs, it's better to directly use a software-command for the log of the cdf of a $N(0,1)$ RV (`pnorm(x,log.p=TRUE)` in R). This way, you avoid having to compute the log of a number very close to zero, which can result in `-Inf`.[2]
Hint: to ensure convergence, you might want to supply a the gradient of your objective function.}

Solution:

```
# Define dependent and explanatory variable
Y_ <- data$paid_in_cash
X_ <- cbind(rep(1), data$price, data$male, data$age,
        data$clothingshoes,data$cosmetics, data$food,data$technology)
X_ <- as.matrix(X_)
k <- ncol(X_)
n <- nrow(X_)

# Construct function to optimise and its gradient:
fLik_i <- function(beta,yi,xi){
    if(yi==1){
        arg <- as.numeric(t(xi)%*%beta)
    }else{
        arg <- - as.numeric(t(xi)%*%beta)
    }
    return(pnorm(arg,log.p=TRUE))
}
```

[1] As part of your derivations for exercise (f), you have to find the score and the Hessian of the objective function in (2),

$$S_n(\beta) \equiv Q_n^{(1)}(\beta; Z_n) \equiv \frac{\partial Q_n(\beta; Z_n)}{\partial \beta} \quad \text{and} \quad H_n(\beta) \equiv Q_n^{(2)}(\beta; Z_n) \equiv \frac{\partial^2 Q_n(\beta; Z_n)}{\partial \beta \partial \beta'} = \frac{\partial S_n(\beta)}{\partial \beta'} .$$

You can also use them to construct your own numerical optimization algorithm to find $\hat{\beta}$.

[2] The alternative is to do manual adjustments, coding -Inf as a very large negative number, but this can be imprecise.

```r
fScore_i <- function(beta,yi,xi){
    if(yi==1){
        arg1 <- as.numeric(t(xi)%*%beta)
        arg2 <- xi
    }else{
        arg1 <- - as.numeric(t(xi)%*%beta)
        arg2 <- - xi
    }
    pdfcdfratio <- exp(dnorm(arg1,log=TRUE)-pnorm(arg1,log.p=TRUE))
    return(pdfcdfratio*arg2)
}

Q_n <- function(beta,Y,X){
    n <- length(Y)
    vLogLiks <- rep(0,n)
    for (i in 1:n){
        vLogLiks[i] <- fLik_i(beta,Y[i],X[i,])
    }
    return(-1/n * sum(vLogLiks))
}

S_n <- function(beta, Y, X) {
    n <- length(Y)
    k <- ncol(X)
    mScores <- matrix(0,n,k)
    for (i in 1:n){
        mScores[i,] <- fScore_i(beta,Y[i],X[i,])
    }
    vScore <- apply(mScores,2,sum)
    return(-1/n * vScore)
}

# Optimise Q_n function
beta_init <- c(rep(0,k)) # initial guess
estimates <- optim(par = beta_init,
                   fn = Q_n,
                   gr = S_n,
                   Y = Y_,
                   X = X_,
                   method = "BFGS",
                   control = list(reltol = 1e-12,maxit=10000,trace=0)
                   )

# Display coefficient estimates
beta_hat <- estimates$par
beta_hat
```

```
## [1]  0.0682050610  0.0001120891 -0.0501665783 -0.0018299908 -0.2878447117
## [6] -0.1195016680  0.0639850350 -0.4194913123
```

```r
# Comparison with built-in GLM function
probit <- glm(paid_in_cash ~ price + male + age
              + clothingshoes + cosmetics + food
              + technology,
```

```
    family = binomial(link = "probit"), data = data)
probit$coefficients
```

```
##    (Intercept)          price           male            age clothingshoes
##   0.0682123408  0.0001120901 -0.0501658057 -0.0018300223 -0.2878515604
##      cosmetics           food     technology
## -0.1195170778  0.0639803682 -0.4195008012
```

5. Based on your estimate, compute the effect of age doubling on the expected probability of using cash for a 30 year-old male who bought clothes/shoes for 500 TRY, i.e. for an observation with $x_i = x_i^* \equiv [1, 500, 1, 30, 1, 0, 0, 0]$. Put differently, this is the difference in expected probabilities of cash payment between a 60 year-old and a 30 year-old male who bought clothes/shoes for 500 TRY. We will call this quantity $\gamma_1(\hat{\beta})$. Also, compute the same effect without conditioning on the category of goods sold in two steps: (i) compute the effect for each of the five categories and (ii) take a weighted average of them, with weights given by the proportions of these goods-categories in overall sales (see your answer to (c)). We will call this quantity $\gamma_2(\hat{\beta})$.

Solution:

Use the formula for the computation of marginal effects in the probit model when the explanatory variable x_i changes from x_1 to x_2 by $\Delta x_i = x_2 - x_1$:

$$\mathbb{E}[y_i|x_i = x_2] - \mathbb{E}[y_i|x_i = x_2] = \Phi(x_2'\beta) - \Phi(x_1'\beta) \ ,$$

where $x_1 = [1, 500, 1, 30, 1, 0, 0, 0]'$ and $x_2 = [1, 500, 1, 60, 1, 0, 0, 0]'$ in our case. We estimate this difference (effect) by replacing β with our estimate $\hat{\beta}$.

```
## Compute marginal effect conditionally on clothes
# Generate x1
x1 <- c(1, 500, 1, 30, 1, 0, 0, 0)

# Generate x2
x2 <- c(1, 500, 1, 60, 1, 0, 0, 0)

# Compute marginal effect
gamma1 <- pnorm(t(x2) %*% beta_hat) - pnorm(t(x1) %*% beta_hat)
gamma1
```

```
##              [,1]
## [1,] -0.02095997
```

```
## Compute unconditional marginal effect
# Generate x1 for all categories
vsCategories <- c("rest","clothesshoes", "cosmetics", "food", "technology")
nC <- length(vsCategories)
vGammas <- rep(0,nC)
mHelp <- matrix(0,k,nC)
mHelp[(k-nC+2):k,2:nC] <- diag(rep(1,nC-1))

for (i in 1:nC){
    x1here <- x1 + mHelp[,i]
    x2here <- x2 + mHelp[,i]
    vGammas[i] <- pnorm(t(x2here) %*% beta_hat) - pnorm(t(x1here) %*% beta_hat)
}

vSalesProportions <- rep(0,nC)
```

```
vSalesProportions[1] <- mSalesProportions$proportion[mSalesProportions$category2=="rest"]
vSalesProportions[2] <- mSalesProportions$proportion[mSalesProportions$category2=="clothingshoes"]
vSalesProportions[3] <- mSalesProportions$proportion[mSalesProportions$category2=="cosmetics"]
vSalesProportions[4] <- mSalesProportions$proportion[mSalesProportions$category2=="food"]
vSalesProportions[5] <- mSalesProportions$proportion[mSalesProportions$category2=="technology"]

gamma2 <- vGammas %*% vSalesProportions
gamma2
```

```
##             [,1]
## [1,] -0.01822036
```

Our estimates suggest that a 60 year old male buying clothes for 500 TRY is 2 percentage points less likely to pay cash than a 30 year old male buying clothes for 500 TRY. The same difference without conditioning on the types of goods bought is a bit smaller, namely 1.8 pp.

6. Suppose that your probit model in (1) is correctly specified. Is your estimator $\hat{\beta}$ consistent? Use the simplified version of the extremum estimation theory we discussed in class to answer this question.

Solution:

Essentially, three things are needed for consistency: i) our sample objective function Q_n converges uniformly in probability to a population objective function Q, ii) this Q is continuous, and iii) this Q is uniquely minimized by β_0.

In our simplified version of the extremum estimation theory, we simply argue that the sample objective function Q_n converges to a population objective function Q, in which we replace the $\frac{1}{n}\sum$ with the expectation operator:

$$
\begin{aligned}
\mathcal{Q}_n(\beta) \equiv \frac{1}{n}\sum_{i=1}^{n} -y_i \ln\left(\Phi(x_i'\beta)\right) - (1-y_i)\ln\left(\Phi(-x_i'\beta)\right) \\
\xrightarrow{p} \mathbb{E}\left[-y_i \ln\left(\Phi(x_i'\beta)\right) - (1-y_i)\ln\left(\Phi(x_i'\beta)\right)\right] \equiv \mathcal{Q}(\beta)
\end{aligned}
$$

(To be precise, this holds because of the Uniform Law of Large Numbers (ULLN).)

The function $\mathcal{Q}(\beta)$ is clearly continuous. The only remaining thing to show is that it is uniquely minimized by β_0. This we can show by showing that the vector of first derivatives of Q – its score – is equal to zero only at β_0. The score is

$$
\begin{aligned}
S(\beta) \equiv \frac{\partial \mathcal{Q}(\beta)}{\partial \beta} &= \mathbb{E}\left[-\frac{y_i}{\Phi(x_i'\beta)}\phi(x_i'\beta)x_i - \frac{1-y_i}{\Phi(-x_i'\beta)}\phi(-x_i'\beta)(-x_i)\right] = \\
&= -\mathbb{E}\left[\left[\frac{y_i}{\Phi(x_i'\beta)} - \frac{1-y_i}{\Phi(-x_i'\beta)}\right]\phi(x_i'\beta)x_i\right] = \\
&= -\mathbb{E}\left[\frac{y_i(1-\Phi(x_i'\beta)) - (1-y_i)\Phi(x_i'\beta)}{\Phi(x_i'\beta)\Phi(-x_i'\beta)}\phi(x_i'\beta)x_i\right] = \\
&= -\mathbb{E}\left[\frac{y_i - \Phi(x_i'\beta)}{\Phi(x_i'\beta)\Phi(-x_i'\beta)}\phi(x_i'\beta)x_i\right] = \\
&= -\mathbb{E}\left[\frac{\mathbb{E}[y_i|x_i] - \Phi(x_i'\beta)}{\Phi(x_i'\beta)\Phi(-x_i'\beta)}\phi(x_i'\beta)x_i\right].
\end{aligned}
$$

Because $\mathbb{E}[y_i|x_i] = \Phi(x_i'\beta_0)$, we have $S(\beta_0) = 0$ and $S(\beta) > 0$ for all $\beta \neq \beta_0$. This proves consistency.

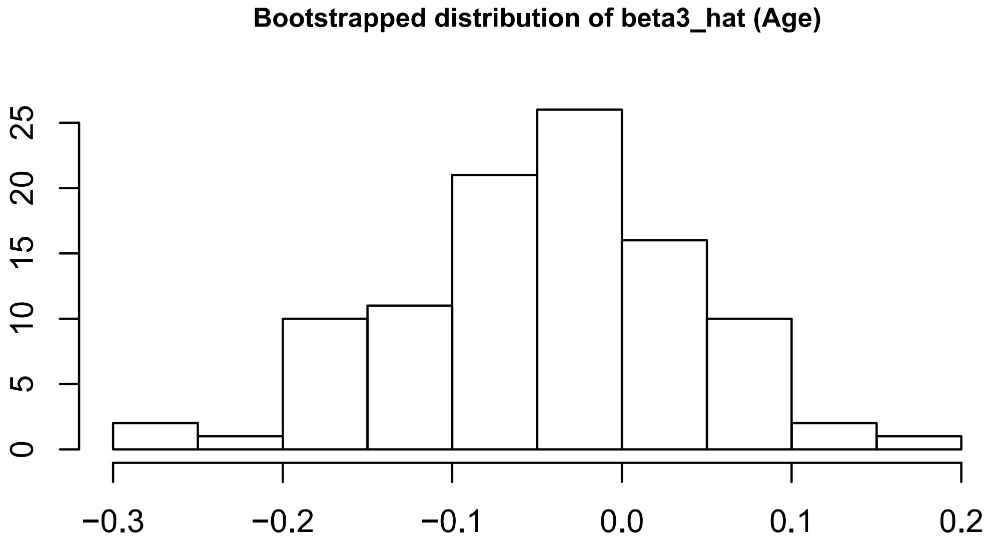
7. Use bootstrapping to find a numerical approximation of the finite sample distribution of $\hat{\beta}$ as well as the two marginal effects $\gamma_1(\hat{\beta})$ and $\gamma_2(\hat{\beta})$: draw $M = 100$ different samples of n observations with

replacement from your dataset and compute (numerically) $\hat{\beta}$, $\gamma_1(\hat{\beta})$ and $\gamma_2(\hat{\beta})$ for each of them. Plot the finite sample distributions you obtained (regarding $\hat{\beta}$, you can limit yourself to the coefficient on *age*).

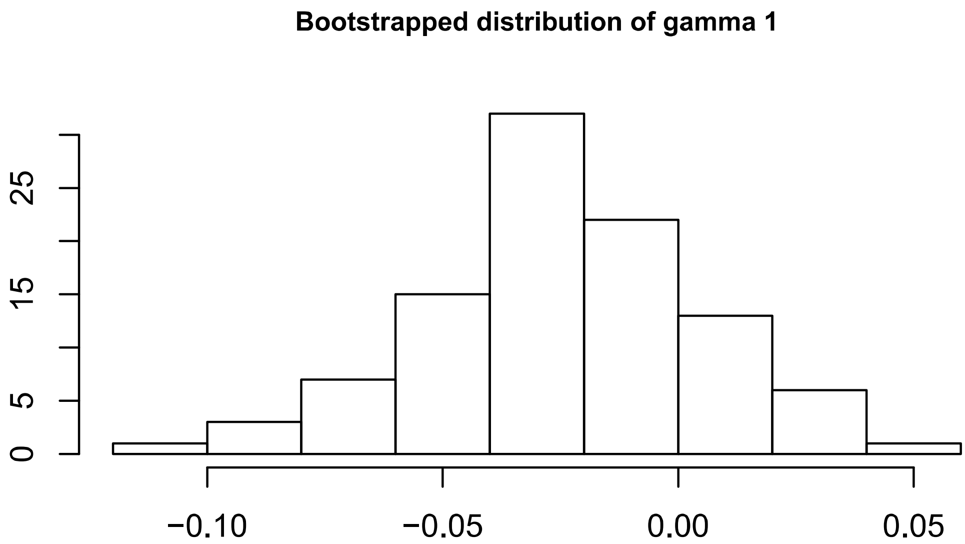
Solution:

```
set.seed(2024)

# Set number of total bootstrap replications
M <- 100

# Here we store the bootstrap replications
mBetaHatsBS <- matrix(NA,nrow=M,ncol=k)
vGamma1BS <- numeric(length=M)
mGammasBS <- matrix(NA,nrow=M,ncol=nC)
vGamma2BS <- numeric(length=M)

for (i in 1:M) {

      # Draw bootstrap sample
      boot_indices <- sample(1:n, size = n, replace = TRUE)
      boot_X <- X_[boot_indices, ]
      boot_Y <- Y_[boot_indices]

      # Optimise Q_n function to find probit estimator:
    estimates <- optim(par = beta_init,
                   fn = Q_n,
                       gr = S_n,
                   Y = boot_Y,
                   X = boot_X,
                   method = "BFGS", # Use BFGS algorithm
                   control = list(reltol = 1e-12,maxit=10000,trace=0)
                   )

    # Store bootstrapped estimates of beta_hat
    mBetaHatsBS[i,1:k] <- estimates$par[1:k]

    # Compute gamma1
    vGamma1BS[i] <- pnorm(t(x2) %*% mBetaHatsBS[i, ]) -
      pnorm(t(x1) %*% mBetaHatsBS[i, ])

    # Compute gamma for each category
    mHelp <- matrix(0,k,nC)
    mHelp[(k-nC+2):k,2:nC] <- diag(rep(1,nC-1))

    for (j in 1:nC) {
        x1here <- x1 + mHelp[,j]
        x2here <- x2 + mHelp[,j]
        mGammasBS[i,j] <- pnorm(t(x2here) %*% mBetaHatsBS[i, ]) -
            pnorm(t(x1here) %*% mBetaHatsBS[i, ])
    }

    # Compute gamma2
    vGamma2BS[i] <- mGammasBS[i, ] %*% vSalesProportions
}
```

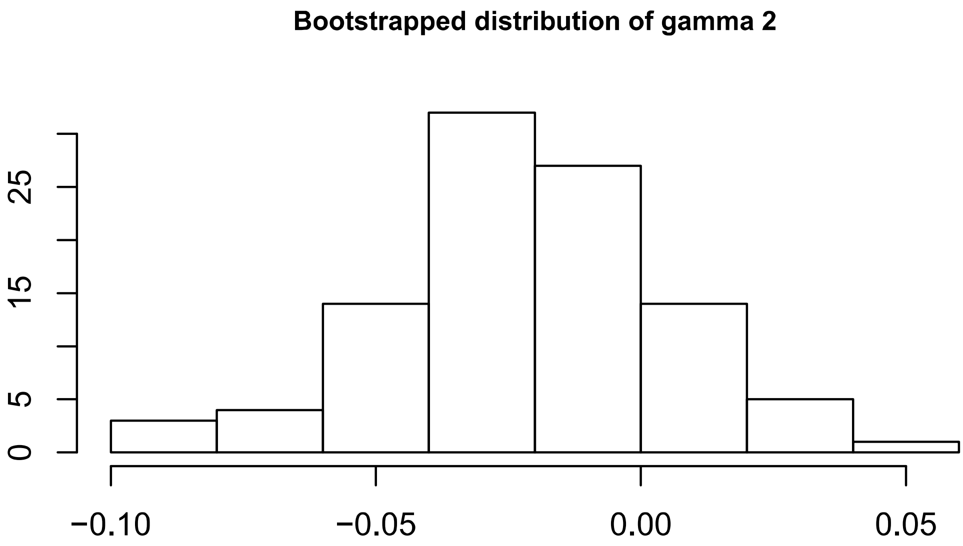
```r
# Plot results
hist_beta3_bs <- hist(mBetaHatsBS[ ,3], plot = FALSE)
plot(hist_beta3_bs,
     xlab="",ylab="",
     main = "Bootstrapped distribution of beta3_hat (Age)",
     cex.main = 0.8)
```




```r
hist_gamma1_bs <- hist(vGamma1BS, plot = FALSE)
plot(hist_gamma1_bs,
     xlab="",ylab="",
     main = "Bootstrapped distribution of gamma 1",
     cex.main = 0.8)
```




```r
hist_gamma2_bs <- hist(vGamma2BS, plot = FALSE)
plot(hist_gamma2_bs,
     xlab="",ylab="",
     main = "Bootstrapped distribution of gamma 2",
     cex.main = 0.8)
```



Simulating M true random samples would allow us to compute numerically the distribution of $\hat{\beta}$ in repeated sampling. Given a single random sample of size n from the population, drawing M different sub-samples with replacement and computing the estimate for each of them gives an approximation of the true finite sample distribution, as long as the sample consists of observations that are equally likely draws from the underlying population. The approximation is better the more representative this single random sample is of the whole population, which is why the approximation generally gets better as the sample size increases.

8. Another approach to approximate the finite sample distribution of $\hat{\beta}$ and functions of it like the marginal effects is to use their asymptotic distribution. Use the simplified version of the extremum estimation theory we discussed in class to show that the asymptotic distribution of $\hat{\beta}$ is given by

$$\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N(0, H^{-1}) \quad \text{with} \quad H = \mathbb{E}[\frac{\phi(x_i'\beta_0)^2}{\Phi(x_i'\beta_0)\Phi(-x_i'\beta_0)} x_i x_i'] \ . \tag{3}$$

   Then, use the asymptotic distribution in (3) to approximate the finite sample distribution of $\hat{\beta}$ in your sample. How does this approximate finite sample distribution of the estimated coefficient on *age* compare to the one obtained via bootstrapping?

Hint: The numerator and the hadenominator in the fraction that appears in H are often both very close to zero. Rather than computing it as-is, first compute the log of it and then take the exponential, i.e. compute

$$\frac{\phi(x_i'\beta_0)^2}{\Phi(x_i'\beta_0)\Phi(-x_i'\beta_0)} \quad as \quad exp\,\{2log\phi(x_i'\beta_0) - log\Phi(x_i'\beta_0) - log\Phi(-x_i'\beta_0)\} \ .$$

To compute log $\phi(x)$ and log $\Phi(x)$, as before in exercise (b), it's better practice to use the log-pdf/cdf software-commands than to compute first the pdf/cdf and then take logs manually (i.e. in R, use `dnorm(x,log=TRUE)` and `pnorm(x,log.p=TRUE)`).

Solution:

In our simplified extermum estimation theory, under the following three conditions, we have that:

$$\sqrt{n}(\hat{\beta}_n - \beta_0) \xrightarrow{d} \mathcal{N}(0, H^{-1}MH^{-1}) \ .$$

First, β_0 is interior (which we just assume). Second, the vector of first derivatives of the sample objective function – its score – when standardized by $\sqrt{n}$ and evaluated at β_0 converges in distribution:

$$\sqrt{n}\mathcal{Q}_n^{(1)}(\beta_0) \xrightarrow{d} \mathcal{N}(0, M) \ .$$

Third, the matrix of second derivatives of the sample objective function – its Hessian –, evaluated at β_0, converges in probability:

$$\mathcal{Q}_n^{(1)}(\beta_0) \xrightarrow{p} H \ .$$

Finally, because we are dealing with a maximum likelihood optimization, we know that $M = H$ will hold, so that we have

$\sqrt{n}(\hat{\beta}_n - \beta_0) \xrightarrow{d} \mathcal{N}(0, H^{-1})$.

Above, we computed the score of the population objective function, $s(\beta) = \mathcal{Q}^{(1)}(\beta)$. It is easy to see that the score of the sample objective function, $S_n(\beta) = \mathcal{Q}_n^{(1)}(\beta)$ is given by the same expression, where we just replace the expectation operator with $1/n\sum_{i=1}^n$:

$$S_n(\beta) = \frac{1}{n}\sum_{i=1}^n -\frac{y_i - \Phi(x_i'\beta)}{\Phi(x_i'\beta)\Phi(-x_i'\beta)}\phi(x_i'\beta)x_i \ .$$

By the CLT, we have:

$$\sqrt{n}S_n(\beta_0) = \frac{1}{\sqrt{n}}\sum_{i=1}^n -\frac{y_i - \Phi(x_i'\beta_0)}{\Phi(x_i'\beta_0)\Phi(-x_i'\beta_0)}\phi(x_i'\beta_0)x_i \xrightarrow{d} N(0, M) \ ,$$

because $\mathbb{E}[-\frac{y_i - \Phi(x_i'\beta_0)}{\Phi(x_i'\beta_0)\Phi(-x_i'\beta_0)}\phi(x_i'\beta_0)x_i] = s(\beta_0) = 0$ (as we established in exercise 6). Thereby,

$$M = \mathbb{E}[\frac{(y_i - \Phi(x_i'\beta_0))^2}{\Phi(x_i'\beta_0)^2\Phi(-x_i'\beta_0)^2}\phi(x_i'\beta_0)^2 x_i x_i'] \ .$$

Based on $S_n(\beta)$, we can compute the Hessian:

$$\begin{aligned}
H_n(\beta) \equiv & \frac{\partial^2 \mathcal{Q}_n(\beta)}{\partial\beta\partial\beta'} = \frac{\partial S_n(\beta)}{\partial\beta'} = \\
= & -\frac{1}{n}\sum_{i=1}^n \left\{ \frac{-\phi(x_i'\beta)\Phi(x_i'\beta)\Phi(-x_i'\beta) - [y_i - \Phi(x_i'\beta)][\phi(x_i'\beta)\Phi(-x_i'\beta)\Phi(x_i'\beta)\phi(-x_i'\beta)]}{\Phi(x_i'\beta)^2\Phi(-x_i'\beta)^2}\phi(x_i'\beta)x_i x_i' + \right. \\
& \left. + x_i\frac{\partial\phi(x_i'\beta)}{\partial\beta'}\frac{y_i - \Phi(x_i'\beta)}{\Phi(x_i'\beta)\Phi(-x_i'\beta)} \right\} = \\
= & -\frac{1}{n}\sum_{i=1}^n \left\{ \left(\frac{1}{\Phi(x_i'\beta)\Phi(-x_i'\beta)} - \frac{[y_i - \Phi(x_i'\beta)][\Phi(-x_i'\beta) - \Phi(x_i'\beta)]}{\Phi(x_i'\beta)^2\Phi(-x_i'\beta)^2} \right)\phi(x_i'\beta)^2 x_i x_i' - \right. \\
& \left. - \frac{y_i - \Phi(x_i'\beta)}{\Phi(x_i'\beta)\Phi(-x_i'\beta)}\phi(x_i'\beta)x_i'\beta x_i x_i' \right\} \ ,
\end{aligned}$$

where

$$\frac{\partial\phi(x_i'\beta)}{\partial\beta'} = \frac{\partial(2\pi)^{-(1/2)}\exp\{-\frac{1}{2}(x_i'\beta)^2\}}{\partial\beta'} = \exp\{-\frac{1}{2}(x_i'\beta)^2\}(2\pi)^{-(1/2)}(-x_i'\beta)x_i' = -\phi(x_i'\beta)(x_i'\beta)x_i' \ .$$

By the WLLN, this sample Hessian evaluated at β_0 converges in probability to the population Hessian H evaluated at β_0:

$$H_n(\beta_0) \equiv \mathcal{Q}_n^{(2)}(\beta_0, Y_n) \xrightarrow{p} H(\beta_0) = \mathbb{E}\left[\frac{\phi(x_i'\beta_0)^2}{\Phi(x_i'\beta_0)\Phi(-x_i'\beta_0)}x_i x_i'\right] \ .$$

The expression for $H(\beta_0)$ does not include the other three terms from $H_n(\beta)$ above because they drop out. This follows from the LIE and because $\mathbb{E}[y_i|x_i] = \Phi(x_i'\beta_0)$. (This is analogous to the reason why $s(\beta_0) = 0$; see exercise 6.) Note that for $H(\beta_0)$ to be positive definite, the matrix $\mathbb{E}[x_i x_i']$ must be positive definite.

By the information matrix equality, we know that $M(\beta_0) = H(\beta_0)$. Overall, then, we get that, by Extremum Estimation Theory, $\hat{\beta}$ converges in distribution to a Normal with mean β_0 and variance $H(\beta_0)^{-1}$:

$$\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N(0, H(\beta_0)^{-1}) \ .$$

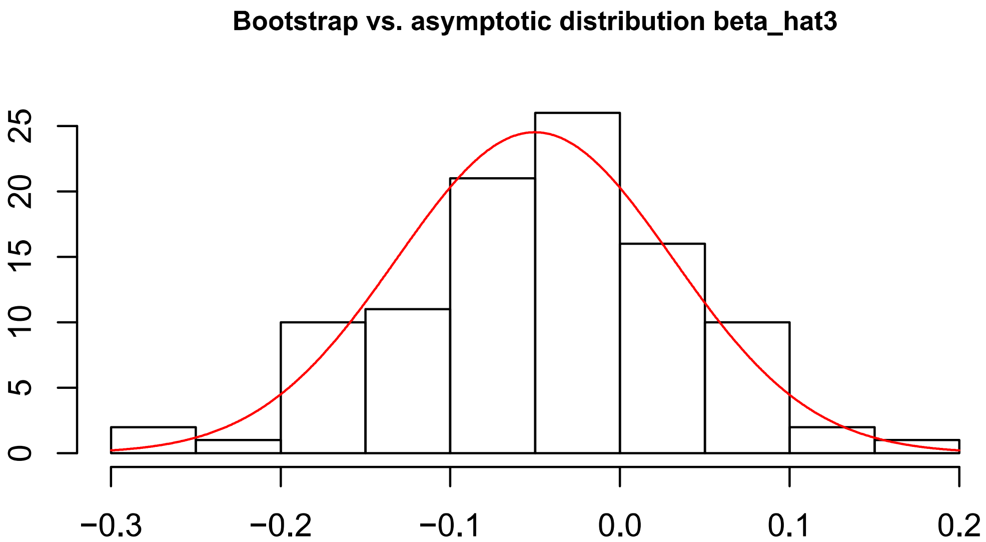
Now, given a consistent estimator for $\hat{H}$, the finite sample distribution of $\hat{\beta}$ can be approximated as:

$$\hat{\beta} \sim N\left(\beta_0, \frac{1}{n}\hat{H}^{-1}\right) \ ,$$

where:

$$\hat{H} = H_n(\hat{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\phi(x_i'\hat{\beta})^2}{\Phi(x_i'\hat{\beta})\Phi(-x_i'\hat{\beta})}x_i x_i'\right] .$$

```r
# Compute the Hessian
mHessian = matrix(0,k,k)
for (ii in 1:n){
    arg = X_[ii,] %*% beta_hat
#to avoid having a very low number in the numerator as well as in the
#denominator, resulting in zero, we compute the log of the first term
#and then exponentiate it:
    logFirstTerm = 2 * dnorm(arg,log=TRUE) -
        pnorm(arg,log.p=TRUE) - pnorm(-arg,log.p=TRUE)

    mHessian <- mHessian + as.numeric(1/n * exp(logFirstTerm)) * X_[ii,] %*% t(X_[ii,])
}

# Obtain the asymptotic VCov matrix of beta_hat
mAVarBetaHat = solve(mHessian)/n

## Plot the comparison with the BS distribution
# First plot histogram of BS distribution
plot(hist_beta3_bs,
     xlab = "", ylab = "",
     main = "Bootstrap vs. asymptotic distribution beta_hat3",
     cex.main = 0.8)

# Adjust scale of asymptotic distribution
y_adj <- length(mBetaHatsBS[,3]) * diff(hist_beta3_bs$breaks)[1]

# Then overlay the asymptotic distribution plot
vXaxis = seq(-0.3,0.2,length.out=1000)
lines(vXaxis,
      y_adj * dnorm(vXaxis, beta_hat[3],
      sd=sqrt(mAVarBetaHat[3,3])),
      col = "red")
```



Overall, the bootstrapped distribution of $\hat{\beta}_3$ resembles the Normal distribution, although it seems to be slightly skewed to the right and has slightly taller tails.

9. Use the asymptotic distribution of $\hat{\beta}$ from (3) and the Delta method to find the asymptotic distribution of $\gamma_1(\hat{\beta})$. Then, use it to approximate the finite sample distribution of $\gamma_1(\hat{\beta})$ in your sample. How does this approximate finite sample distribution compare to the one obtained via bootstrapping?

Solution:

Our object of interest, $\gamma_1(\beta) = \Phi(x_2'\beta) - \Phi(x_1'\beta)$ is a continuous function of β. The Delta method tells us then that our estimator of it, $\gamma_1(\hat{\beta})$, is asymptotically Normally distributed:

$$\sqrt{n}(\gamma_1(\hat{\beta}) - \gamma_1(\beta_0)) \xrightarrow{d} N(0, V) \ ,$$

with:

$$V = \left(\frac{\partial \gamma_1(\beta_0)}{\partial \beta_0'}\right) H^{-1} \left(\frac{\partial \gamma_1(\beta_0)}{\partial \beta_0'}\right)' \ ,$$

whereby $\frac{\partial \gamma_1(\beta)}{\partial \beta} = \phi(x_2'\beta)x_2 - \phi(x_1'\beta)x_1$ in our case.

From this, we conclude that:

$$\gamma_1(\hat{\beta}) \overset{approx.}{\sim} N\left(\gamma_1(\beta_0), \frac{1}{n}\hat{V}\right)$$

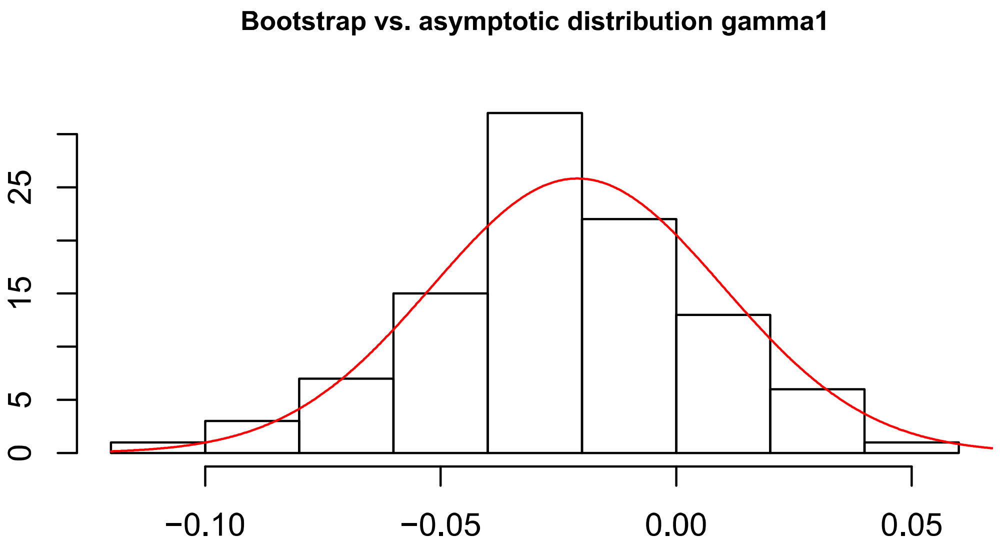
with:

$$\hat{V} = \left(\frac{\partial \gamma_1(\hat{\beta})}{\partial \hat{\beta}'}\right) \hat{H}^{-1} \left(\frac{\partial \gamma_1(\hat{\beta})}{\partial \hat{\beta}'}\right)' \ .$$

We can now compare this approximate finite sample distribution with the one we obtained with the bootstrap:

```
# Compute partial derivative
p1 <- x1 %*% beta_hat
p2 <- x2 %*% beta_hat
partial_gamma <- dnorm(p2) * x2 - dnorm(p1) * x1
```

```r
# Compute asymptotic variance
partial_gamma <- as.numeric(partial_gamma)
mPartialGamma <- matrix(partial_gamma, nrow = 1)
AVargamma <- mPartialGamma %*% mAVarBetaHat %*% t(mPartialGamma)

# Plot comparison with BS distribution
plot(hist_gamma1_bs,
     xlab = "", ylab = "",
     main = "Bootstrap vs. asymptotic distribution gamma1",
     cex.main = 0.8)

y_adj <- length(vGamma1BS) * diff(hist_gamma1_bs$breaks)[1]

vXaxis = seq(-0.12,0.07,length.out=1000)
lines(vXaxis,
      y_adj * dnorm(vXaxis, gamma1, sd=sqrt(AVargamma)),
      col = "red")
```



Also in this case, the bootstrapped distribution of γ_1 is fairly close to the Normal, although it displays a slight skewness to the left.

10. Now let's test whether the true partial effect $\gamma_1(\hat{\beta})$ is significantly different from 0 at the $\alpha = 0.05$ level:

$$\mathcal{H}_0 : \gamma_1(\beta) = 0 \quad \text{vs.} \quad \mathcal{H}_1 : \gamma_1(\beta) \neq 0 \, .$$

(In other words, we are testing whether the expected probabilities of cash payment for a 30 year-old and a 60 year-old male buying clothes for 500 TRY are different.) One approach to do so uses the finite sample distribution of $\gamma_1(\hat{\beta})$ approximated via its asymptotic distribution, which you found in the exercise before:

$$\gamma_1(\hat{\beta}) \overset{approx.}{\sim} N(\gamma_1(\beta), \frac{1}{n}\hat{V}) \, ,$$

for some $\hat{V}$ you had to find. Use this expression to construct a t-test. What do you conclude? Also, use the above expression to construct a 95% confidence interval for $\gamma_1(\beta)$.[3] (If you couldn't find $\hat{V}$, just state the test statistic and critical value for a general $\hat{V}$.)

[3]Note that in general, we would use the Wald-test. Here we can use the t-test because we are testing a single thing, i.e. our testing function $g(\beta) = \gamma_1(\beta) = 0$ is a scalar. Our t-test will give the same result as the Wald test, because the asymptotic distribution of the Wald-test-statistic is derived in the same way as that of our t-test statistic here (i.e. it also uses the Delta method), except that it squares things in the end to go from a Normal to a Chi-Squared distribution.

Solution:

Under $\mathcal{H}_0$,

$$\gamma_1(\hat{\beta}) \overset{approx.}{\sim} N\left(0, \frac{1}{n}\hat{V}\right) \ ,$$

and hence, also under $\mathcal{H}_0$, for the t-test statistic we have:

$$t = \left| \frac{\gamma_1(\hat{\beta}) - 0}{\sqrt{\frac{1}{n}\hat{V}}} \right| \sim N(0,1) \ .$$

```r
# Compute t-stat
tstat <- abs(gamma1/sqrt(AVargamma))
tstat
```

```
##           [,1]
## [1,] 0.6783944
```

Since the test statistic $t = 0.68$ is below the critical value $c_{\alpha=0.05} = 1.96$, we accept (or fail to reject) the null hypothesis that the marginal effect is zero at the 5% significance level.

We now compute the 95% CI by inverting the t-statistic:

$$CI_{95\%} := \left[\gamma_1(\hat{\beta}) - 1.96 \times \sqrt{\hat{V}/n}; \ \ \gamma_1(\hat{\beta}) + 1.96 \times \sqrt{\hat{V}/n}\right]$$

```r
# Compute bounds for CI
lower_bound <- gamma1 - 1.96*sqrt(AVargamma)
upper_bound <- gamma1 + 1.96*sqrt(AVargamma)

CI95 <- c(lower_bound, upper_bound)
CI95
```

```
## [1] -0.08151698  0.03959704
```

Hence,

$$CI_{95\%} := [-0.08; \ \ 0.04]$$