

NIFB22001U Introduction to Econometrics

Exam 2022/2023 - R Code Attachment

Abdullah Faqih Al Mubarak

```
#load libraries
library(tidyverse)
library(stargazer)
library(lmtest)
library(car)
library(sandwich)
library(AER)
library(ggplot2)
library(patchwork)
library(corrplot)

#load data
data_01 <- get(load("examdata2023_01.RData"))
data_02 <- get(load("examdata2023_02.RData"))
```

Part 1: The Returns to Education

1. Below code is used for estimating

$$wage = \alpha + \beta educ + u \quad (1)$$

```
model_01_pred <- lm(data = data_01,
                    formula = wage ~ education)

#Variance
```

```
var(data_01$education)
```

```
stargazer(model_01_pred, type = "text")
```

2. Covariance check between education and libraries

```
cov(data_01$education, data_01$libraries)
```

3. 2SLS regression of model (1)

```
model_01_iv_pred <- ivreg(wage ~ education | libraries,  
                           data = data_01,  
                           model = TRUE)
```

```
stargazer(model_01_pred, model_01_iv_pred, type = "text")
```

Part 2: Global Food Consumption Patterns

2.1 Data description and preliminary analysis

1. Descriptive statistic of supplied data

```
stargazer(data_02, type = "text", median = TRUE, mean.sd = TRUE, digits  
↪ = 2)
```

2. Consumption proportion vs GDP Per capita

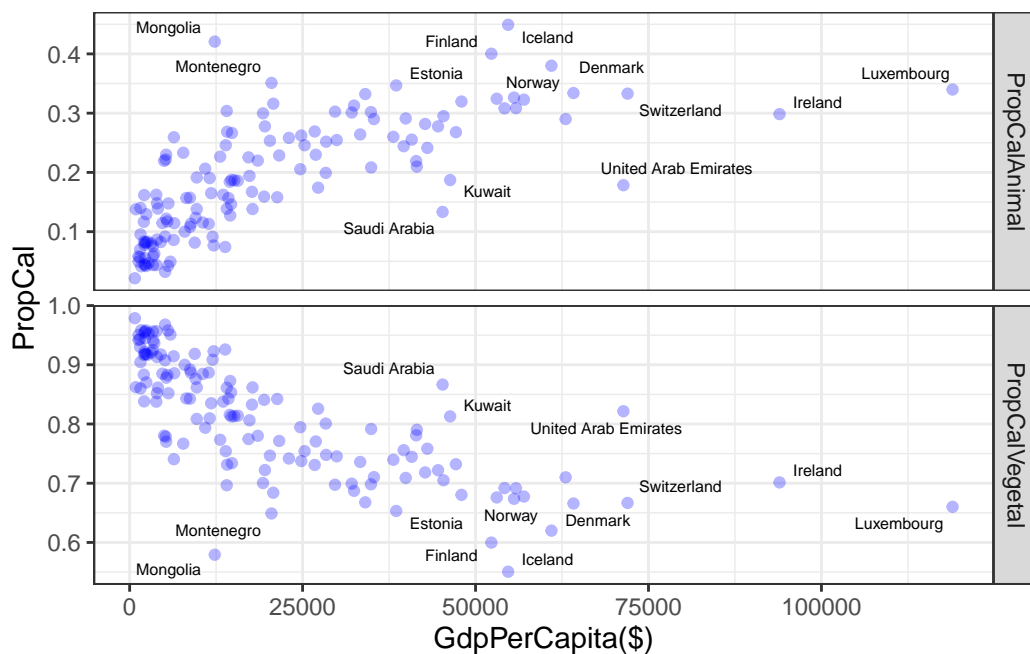
```
#|warning : false  
#|eval: false  
propcal_gdp <- data_02 |>  
  select(Country, PropCalAnimal, PropCalVegetal, GdpPerCapita) |>  
  pivot_longer(cols = starts_with("PropCal"),  
               names_to = "ConsumType", values_to = "PropCal")  
  
ggplot(data = propcal_gdp,  
       mapping = aes(y = PropCal, x = GdpPerCapita)) +
```

```
geom_point(color = "blue", alpha = 0.3) +
geom_text_repel(size = 2, verbose = FALSE,
  mapping = aes(label = Country)) +
facet_grid(rows = vars(ConsumType), scales = "free") +
labs(x = "GdpPerCapita($)") +
theme_bw()
```

Warning: Removed 38 rows containing missing values (`geom_point()`).

Warning: Removed 38 rows containing missing values (`geom_text_repel()`).

Warning: ggrepel: 127 unlabeled data points (too many overlaps). Consider increasing max.overlaps
ggrepel: 127 unlabeled data points (too many overlaps). Consider increasing max.overlaps



2. PropCalAnimal and log(PropCalAnimal) distribution

```
#|warning : false
#|eval: false

# Proceeding without missing values
```

```

data_02 <- data_02 |>
  filter(!is.na(PropCalAnimal))

PropCalAnimal_dist <- data_02 |>
  select(PropCalAnimal) |>
  mutate("log(PropCalAnimal)" = log(PropCalAnimal))

p1 <- ggplot(data = PropCalAnimal_dist,
             mapping = aes(x = PropCalAnimal)) +
  geom_histogram(mapping = aes(y = ..density..),
                fill = "lightblue", color = "lightblue",
                alpha = 0.75) +
  geom_density(color = "orange") +
  theme_bw()

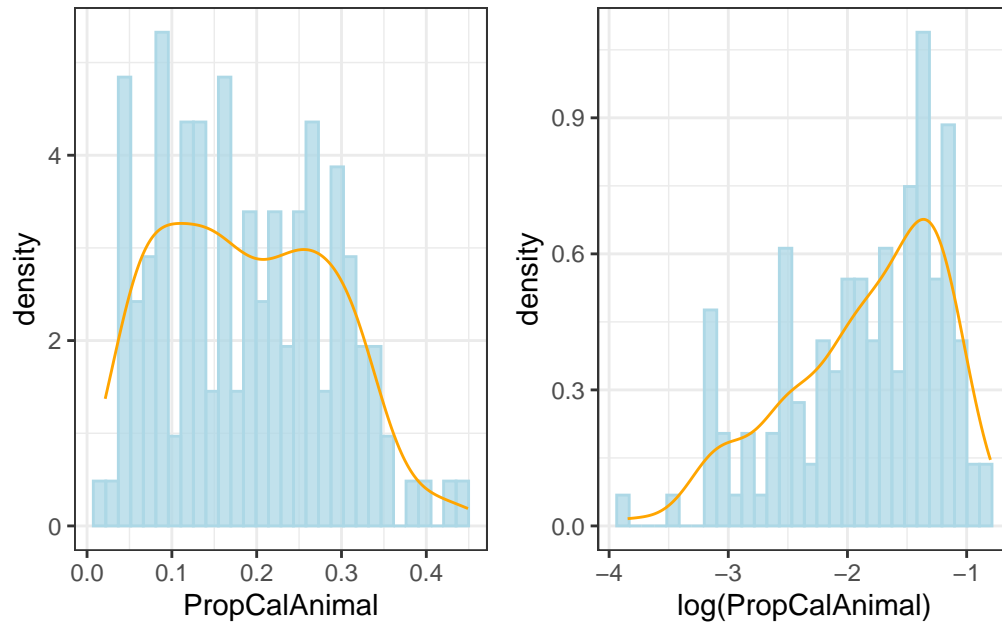
p2 <- ggplot(data = PropCalAnimal_dist,
             mapping = aes(x = `log(PropCalAnimal)`)) +
  geom_histogram(mapping = aes(y = ..density..),
                fill = "lightblue", color = "lightblue",
                alpha = 0.75) +
  geom_density(color = "orange") +
  theme_bw()

p1+p2

```

Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
 i Please use `after_stat(density)` instead.

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



3. Estimation result, Correlation of explanatory variables, and Homoskedasticity check of model (2):

```
expl_vars_cor <- cor((data_02 |> select(AgLandShare, AgGdpShare,
  ↪ ArableLandShare, GdpPerCapita, HCI, LifeExp, PopMiddle, PopOld,
  ↪ PopFemale, PopUrban)))

corrplot(expl_vars_cor,method = 'color',type = 'upper', addCoef.col =
  ↪ 'black',
  title = "Correlation of Explanatory Variables",
  number.digits = 1, mar=c(0,0,1,0)#Fix the title position
)

model_02 <- lm(data = data_02,
  formula = PropCalAnimal ~ AgLandShare + AgGdpShare +
  ArableLandShare + GdpPerCapita + HCI + LifeExp +
  ↪ PopMiddle +
  PopOld + PopFemale + PopUrban)

stargazer(model_02, type = "text", font.size = "footnotesize",
  column.sep.width = "1pt", single.row = TRUE,
  digits=2, no.space = TRUE)
```

```
bptest(model_02)
```

3. Estimation result and Homoskedasticity check of model (3):

```
model_03 <- lm(data = data_02,  
               formula = log(PropCalAnimal) ~ AgLandShare + AgGdpShare +  
               ArableLandShare + GdpPerCapita + HCI + LifeExp +  
               ↪ PopMiddle +  
               PopOld + PopFemale + PopUrban)  
  
stargazer(model_03, type = "text", font.size = "footnotesize",  
           column.sep.width = "1pt", single.row = TRUE,  
           digits=2, no.space = TRUE)  
  
bptest(model_03)
```

4. Residual distribution of model (2) and model (3):

```
res_model_02 <- ggplot(data = data.frame(value = residuals(model_02)),  
                      mapping = aes(x = value)) +  
  geom_histogram(mapping = aes(y = ..density..),  
                fill = "lightblue", color = "lightblue",  
                alpha = 0.75) +  
  geom_density(color = "orange") +  
  xlab("Residuals of model (2)") +  
  theme_bw()  
  
res_model_03 <- ggplot(data = data.frame(value = residuals(model_03)),  
                      mapping = aes(x = value)) +  
  geom_histogram(mapping = aes(y = ..density..),  
                fill = "lightblue", color = "lightblue",  
                alpha = 0.75) +  
  geom_density(color = "orange") +  
  xlab("Residuals of model (3)") +  
  theme_bw()  
  
res_model_02 + res_model_03
```

5. Restricted F-test for model(2) by removing AgLandShare and AgGdpShare

```

model_02_restrict <- lm(data = data_02,
                        formula = PropCalAnimal ~ ArableLandShare +
                        GdpPerCapita + HCI + LifeExp +
                        PopMiddle + PopOld + PopFemale + PopUrban)

anova(model_02_restrict, model_02)

```

6. Estimation result of model (4):

```

model_04 <- lm(data = data_02,
               formula = PropCalAnimal ~ ArableLandShare +
               ↪ log(GdpPerCapita)
               + HCI + LifeExp + PopMiddle + PopOld + PopFemale +
               ↪ PopUrban)

stargazer(model_04, type = "text", font.size = "footnotesize",
           column.sep.width = "1pt", single.row = TRUE,
           digits=2, no.space = TRUE)

```

7. Estimation result of model (4) with Quadratic HCI:

```

model_04_hci_square <- lm(data = data_02,
                          formula = PropCalAnimal ~ ArableLandShare +
                          ↪ log(GdpPerCapita)
                          + HCI + I(HCI^2) + LifeExp + PopMiddle + PopOld +
                          ↪ PopFemale + PopUrban)

stargazer(model_04_hci_square, type = "text", font.size =
  ↪ "footnotesize",
          column.sep.width = "1pt", single.row = TRUE,
          digits=4, no.space = TRUE)

```

8. Complete comparison of models

```

bp_test_model_02 <- bptest(model_02)
bp_test2_model_03 <- bptest(model_03)
bp_test2_model_04 <- bptest(model_04)

```

```

reset_test_model_02 <- resettest(model_02, power=2, type="regressor",
  ↪ vcov. = vcovHC(model_02, "HC1"))
reset_test_model_03 <- resettest(model_03, power=2, type="regressor",
  ↪ vcov. = vcovHC(model_03, "HC1"))
reset_test_model_04 <- resettest(model_04, power=2, type="regressor",
  ↪ vcov. = vcovHC(model_04, "HC1"))

stargazer(model_02, model_03, model_04, type="text",
  align=TRUE,
  add.lines=list(
    c("BP Test P-value", sprintf("%.3f",
      ↪ bp_test_model_02$p.value),
      sprintf("%.3f", bp_test2_model_03$p.value),
      sprintf("%.3f", bp_test2_model_04$p.value)),
    c("RESET (power=2) Test P-value", sprintf("%.3f",
      ↪ reset_test_model_02$p.value),
      sprintf("%.3f", reset_test_model_03$p.value),
      sprintf("%.3f", reset_test_model_04$p.value))
  ),
  column.sep.width = "1pt",
  font.size = "footnotesize"
)

```

9. Model(3) summary with robust-heteroskedasticity standard error

```

cov1 = vcovHC(model_03, type = "HC1")
robust_se = sqrt(diag(cov1))
# Report in stargazer
stargazer(model_03, model_03,
  type = "text",
  se = list(NULL, robust_se), # setting to robust se
  column.sep.width = "1pt",
  font.size = "footnotesize",
  omit.stat = c("ser", "F", "adj.rsq", "rsq")
)

```