

# EDS241: Assignment 4

Wylie Hampson

03/10/2022

This question will ask you to estimate the price elasticity of demand for fresh sardines across 56 ports located in 4 European countries with monthly data from 2013 to 2019. The data are contained in the file EU\_sardines.csv, which is available on Gauchospace.

Each row in the data file is a combination of port location (where the fish is landed and sold) in a given year and month. You can ignore the fact that the sample is not balanced (the number of monthly observations varies across ports).

For the assignment, you will need the following variables: year, month, country, port (port where sardines are landed and sold), price\_euro\_kg (price per kg in €), and volume\_sold\_kg (quantity of sardines sold in kg). In the questions below, I use  $\log()$  to denote the natural logarithm.

**Read in the data:**

```
sardines <- read.csv(here("data", "EU_sardines.csv")) %>%  
  mutate(log_price_euro_kg = log(price_euro_kg), log_volume_sold_kg = log(volume_sold_kg))
```

**Question a:** Estimate a bivariate regression of  $\log(\text{volume\_sold\_kg})$  on  $\log(\text{price\_euro\_kg})$ . What is the price elasticity of demand for sardines? Test the null hypothesis that the price elasticity is equal to -1.

```
model_a <- lm_robust(data = sardines, log_volume_sold_kg ~ log_price_euro_kg)  
  
model_a_table <- broom::tidy(model_a) %>%  
  dplyr::select(term, estimate, std.error, p.value,) %>%  
  knitr::kable()  
  
model_a_table
```

term	estimate	std.error	p.value
(Intercept)	7.759061	0.0430246	0
log_price_euro_kg	-1.545335	0.0781254	0

```
linearHypothesis(model_a, c("log_price_euro_kg = -1"), white.adjust = "hc2")
```

```
## Linear hypothesis test  
##  
## Hypothesis:  
## log_price_euro_kg = - 1  
##
```

The price elasticity of sardines is -1.55. Because our linear hypothesis gives us a very low p-value under 0.001, we can reject the null hypothesis that the price elasticity is equal to -1.

```
model_b <- lm_robust(data = sardines, formula = log_price_euro_kg ~ wind_m_s)

model_b_table <- broom::tidy(model_b) %>%
  dplyr::select(term, estimate, std.error, p.value) %>%
  knitr::kable()

model_b_table
```

term	estimate	std.error	p.value
(Intercept)	-0.3048875	0.0273093	0
wind_m_s	0.0673459	0.0055995	0

[illegible]

2

that we would expect to see because higher windspeed would make fishing more difficult, which would raise the price of fish. The F-statistic for wind speed is 144.6526487, so it is a relevant and non-weak instrument.

**Question c:** Estimate the TSLS estimator of the price elasticity of demand for sardines using `wind_m_s` as an instrument for `log(price_euro_kg)`. What is the estimated price elasticity of demand for sardines?

```
tsls <- ivreg(log_volume_sold_kg ~ log_price_euro_kg | wind_m_s, data = sardines)
summary(tsls)
```

```
##
## Call:
## ivreg(formula = log_volume_sold_kg ~ log_price_euro_kg | wind_m_s,
##       data = sardines)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.8626 -1.9790 -0.2333  2.0950  6.2354
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept)    7.75534    0.04331  179.08 <0.0000000000000002 ***
## log_price_euro_kg -1.08802    0.37003   -2.94      0.0033 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.728 on 3986 degrees of freedom
## Multiple R-Squared:  0.09529, Adjusted R-squared:  0.09506
## Wald test: 8.646 on 1 and 3986 DF,  p-value: 0.003297
```

```
tsls_table <- broom::tidy(tsls) %>%
  dplyr::select(term, estimate, std.error, p.value) %>%
  knitr::kable()
```

```
tsls_table
```

term	estimate	std.error	p.value
(Intercept)	7.755341	0.0433068	0.0000000
log_price_euro_kg	-1.088015	0.3700295	0.0032973

The 2 stage least square estimator for price elasticity of demand for sardines, using wind speed as an instrument for `log(price_euro_kg)`, is -1.0880152.

**Question d:** Repeat the exercise in (c), but include fixed effects for each year, month, and country. [Hint: you can use the command “`as.factor(country) + as.factor(year) + as.factor(month)`” to the `ivreg` function in R]. Report the estimated price elasticity of demand and the F-statistic testing for relevant and non-weak instruments.

```
tsls_d <- ivreg(log_volume_sold_kg ~ log_price_euro_kg +
  as.factor(country) +
  as.factor(year) +
  as.factor(month) | wind_m_s +
```

```

        as.factor(country) +
        as.factor(year) +
        as.factor(month),
        data = sardines)

summary(tsls_d)

##
## Call:
## ivreg(formula = log_volume_sold_kg ~ log_price_euro_kg + as.factor(country) +
##       as.factor(year) + as.factor(month) | wind_m_s + as.factor(country) +
##       as.factor(year) + as.factor(month), data = sardines)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.2940 -1.8317 -0.1353  1.9969  6.2894
##
## Coefficients:
##                                Estimate Std. Error t value
## (Intercept)                   7.33742    0.20781  35.309
## log_price_euro_kg              -1.25004    0.46393  -2.694
## as.factor(country)Italy        -0.68925    0.12970  -5.314
## as.factor(country)Portugal     1.71563    0.34614   4.957
## as.factor(country)United Kingdom -0.07422    0.31428  -0.236
## as.factor(year)2014            0.14610    0.15281   0.956
## as.factor(year)2015            0.18487    0.15221   1.215
## as.factor(year)2016            0.21335    0.15320   1.393
## as.factor(year)2017            0.07400    0.15224   0.486
## as.factor(year)2018           -0.09137    0.15508  -0.589
## as.factor(year)2019            0.03602    0.19688   0.183
## as.factor(month)2              0.06866    0.20972   0.327
## as.factor(month)3              0.51583    0.20489   2.518
## as.factor(month)4              0.91433    0.20297   4.505
## as.factor(month)5              1.14887    0.20370   5.640
## as.factor(month)6              1.14474    0.20164   5.677
## as.factor(month)7              1.40047    0.21047   6.654
## as.factor(month)8              1.26382    0.21692   5.826
## as.factor(month)9              1.31072    0.21298   6.154
## as.factor(month)10             0.72059    0.22958   3.139
## as.factor(month)11             0.48128    0.22575   2.132
## as.factor(month)12             0.06683    0.21920   0.305
##                                Pr(>|t|)
## (Intercept)                   < 0.0000000000000002 ***
## log_price_euro_kg              0.00708 **
## as.factor(country)Italy        0.0000001129314 ***
## as.factor(country)Portugal     0.0000007476568 ***
## as.factor(country)United Kingdom 0.81332
## as.factor(year)2014            0.33909
## as.factor(year)2015            0.22461
## as.factor(year)2016            0.16382
## as.factor(year)2017            0.62692
## as.factor(year)2018            0.55580
## as.factor(year)2019            0.85483

```

```
## as.factor(month)2          0.74339
## as.factor(month)3          0.01185 *
## as.factor(month)4          0.0000068372477 ***
## as.factor(month)5          0.0000000181902 ***
## as.factor(month)6          0.0000000146801 ***
## as.factor(month)7          0.0000000000324 ***
## as.factor(month)8          0.0000000061221 ***
## as.factor(month)9          0.0000000008287 ***
## as.factor(month)10         0.00171 **
## as.factor(month)11         0.03308 *
## as.factor(month)12         0.76049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.648 on 3966 degrees of freedom
## Multiple R-Squared:  0.1522, Adjusted R-squared:  0.1477
## Wald test: 9.963 on 21 and 3966 DF, p-value: < 0.00000000000000022
```

```
tsls_d_table <- broom::tidy(tsls_d) %>%
  dplyr::select(term, estimate, std.error, p.value) %>%
  knitr::kable()
```

```
tsls_d_table
```

term	estimate	std.error	p.value
(Intercept)	7.3374236	0.2078052	0.0000000
log_price_euro_kg	-1.2500413	0.4639288	0.0070798
as.factor(country)Italy	-0.6892477	0.1296958	0.0000001
as.factor(country)Portugal	1.7156252	0.3461351	0.0000007
as.factor(country)United Kingdom	-0.0742223	0.3142812	0.8133168
as.factor(year)2014	0.1461004	0.1528131	0.3390939
as.factor(year)2015	0.1848675	0.1522107	0.2246103
as.factor(year)2016	0.2133455	0.1532014	0.1638245
as.factor(year)2017	0.0740035	0.1522378	0.6269200
as.factor(year)2018	-0.0913677	0.1550847	0.5557960
as.factor(year)2019	0.0360239	0.1968818	0.8548292
as.factor(month)2	0.0686605	0.2097237	0.7433936
as.factor(month)3	0.5158251	0.2048850	0.0118537
as.factor(month)4	0.9143275	0.2029709	0.0000068
as.factor(month)5	1.1488658	0.2036995	0.0000000
as.factor(month)6	1.1447382	0.2016413	0.0000000
as.factor(month)7	1.4004650	0.2104651	0.0000000
as.factor(month)8	1.2638172	0.2169211	0.0000000
as.factor(month)9	1.3107231	0.2129754	0.0000000
as.factor(month)10	0.7205877	0.2295819	0.0017095
as.factor(month)11	0.4812754	0.2257501	0.0330768
as.factor(month)12	0.0668261	0.2191991	0.7604851

```
# To find the F-statistic
model_d <- lm_robust(log_price_euro_kg ~ wind_m_s +
  as.factor(country) +
  as.factor(year) +
```

```

        as.factor(month),
        data = sardines)

model_d_test <- linearHypothesis(model_d, "wind_m_s = 0", white.adjust = "hc2")

model_d_test

## Linear hypothesis test
##
## Hypothesis:
## wind_m_s = 0
##
## Model 1: restricted model
## Model 2: log_price_euro_kg ~ wind_m_s + as.factor(country) + as.factor(year) +
##          as.factor(month)
##
##      Res.Df Df    Chisq          Pr(>Chisq)
## 1      3967
## 2      3966  1  77.658 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

*Using fixed effects for each year, country, and month, the estimated price elasticity of demand is -1.25. The F-statistic is 77.66, meaning these are not a weak instruments.*