Aplicación 1.3: Curva de Engel para el gasto en alimentos

# Packages  
library(readr)  
library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

library(ggplot2)  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

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

library(splines)  
#  
engel <- read\_delim("ENGEL\_ALIM\_1.csv", ";", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## GALIM = col\_double(),  
## RENTA = col\_double()  
## )

# Estadísticos descriptivos  
head(engel)

## # A tibble: 6 x 2  
## GALIM RENTA  
## <dbl> <dbl>  
## 1 256. 420.  
## 2 311. 541.  
## 3 486. 901.  
## 4 403. 639.  
## 5 496. 751.  
## 6 634. 946.

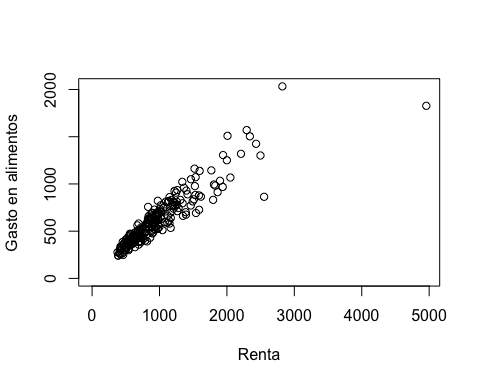
summary(engel)

## GALIM RENTA   
## Min. : 242.3 Min. : 377.1   
## 1st Qu.: 429.7 1st Qu.: 638.9   
## Median : 582.5 Median : 884.0   
## Mean : 624.2 Mean : 982.5   
## 3rd Qu.: 743.9 3rd Qu.:1164.0   
## Max. :2032.7 Max. :4957.8

dim(engel)

## [1] 235 2

# Diagrama de puntos (scatter plot) de las variables x e y:  
plot(engel$RENTA, engel$GALIM,   
 ylim=c(0, max(engel$GALIM)),   
 xlim=c(0, max(engel$RENTA)),   
 xlab="Renta",   
 ylab="Gasto en alimentos",  
 type = "p")



# Estimación de un modelo por MCO -OLS en inglés- y guardar los resultados en el objeto "ols":  
ols <- lm(GALIM ~ RENTA, data = engel)  
class(ols)

## [1] "lm"

names(ols)

## [1] "coefficients" "residuals" "effects" "rank"   
## [5] "fitted.values" "assign" "qr" "df.residual"   
## [9] "xlevels" "call" "terms" "model"

str(ols)

## List of 12  
## $ coefficients : Named num [1:2] 147.475 0.485  
## ..- attr(\*, "names")= chr [1:2] "(Intercept)" "RENTA"  
## $ residuals : Named num [1:235] -95.5 -99.2 -99 -54.5 -16.2 ...  
## ..- attr(\*, "names")= chr [1:235] "1" "2" "3" "4" ...  
## $ effects : Named num [1:235] -9568.04 3853.63 -92.65 -45.24 -8.17 ...  
## ..- attr(\*, "names")= chr [1:235] "(Intercept)" "RENTA" "" "" ...  
## $ rank : int 2  
## $ fitted.values: Named num [1:235] 351 410 585 458 512 ...  
## ..- attr(\*, "names")= chr [1:235] "1" "2" "3" "4" ...  
## $ assign : int [1:2] 0 1  
## $ qr :List of 5  
## ..$ qr : num [1:235, 1:2] -15.3297 0.0652 0.0652 0.0652 0.0652 ...  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:235] "1" "2" "3" "4" ...  
## .. .. ..$ : chr [1:2] "(Intercept)" "RENTA"  
## .. ..- attr(\*, "assign")= int [1:2] 0 1  
## ..$ qraux: num [1:2] 1.07 1.05  
## ..$ pivot: int [1:2] 1 2  
## ..$ tol : num 1e-07  
## ..$ rank : int 2  
## ..- attr(\*, "class")= chr "qr"  
## $ df.residual : int 233  
## $ xlevels : Named list()  
## $ call : language lm(formula = GALIM ~ RENTA, data = engel)  
## $ terms :Classes 'terms', 'formula' language GALIM ~ RENTA  
## .. ..- attr(\*, "variables")= language list(GALIM, RENTA)  
## .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. ..$ : chr [1:2] "GALIM" "RENTA"  
## .. .. .. ..$ : chr "RENTA"  
## .. ..- attr(\*, "term.labels")= chr "RENTA"  
## .. ..- attr(\*, "order")= int 1  
## .. ..- attr(\*, "intercept")= int 1  
## .. ..- attr(\*, "response")= int 1  
## .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. ..- attr(\*, "predvars")= language list(GALIM, RENTA)  
## .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "numeric"  
## .. .. ..- attr(\*, "names")= chr [1:2] "GALIM" "RENTA"  
## $ model :'data.frame': 235 obs. of 2 variables:  
## ..$ GALIM: num [1:235] 256 311 486 403 496 ...  
## ..$ RENTA: num [1:235] 420 541 901 639 751 ...  
## ..- attr(\*, "terms")=Classes 'terms', 'formula' language GALIM ~ RENTA  
## .. .. ..- attr(\*, "variables")= language list(GALIM, RENTA)  
## .. .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. .. ..$ : chr [1:2] "GALIM" "RENTA"  
## .. .. .. .. ..$ : chr "RENTA"  
## .. .. ..- attr(\*, "term.labels")= chr "RENTA"  
## .. .. ..- attr(\*, "order")= int 1  
## .. .. ..- attr(\*, "intercept")= int 1  
## .. .. ..- attr(\*, "response")= int 1  
## .. .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. .. ..- attr(\*, "predvars")= language list(GALIM, RENTA)  
## .. .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "numeric"  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "GALIM" "RENTA"  
## - attr(\*, "class")= chr "lm"

#  
# Resumen de resultados:  
summary(ols)

##   
## Call:  
## lm(formula = GALIM ~ RENTA, data = engel)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -725.70 -60.24 -4.32 53.41 515.77   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 147.47539 15.95708 9.242 <2e-16 \*\*\*  
## RENTA 0.48518 0.01437 33.772 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 114.1 on 233 degrees of freedom  
## Multiple R-squared: 0.8304, Adjusted R-squared: 0.8296   
## F-statistic: 1141 on 1 and 233 DF, p-value: < 2.2e-16

anova(ols)

## Analysis of Variance Table  
##   
## Response: GALIM  
## Df Sum Sq Mean Sq F value Pr(>F)   
## RENTA 1 14850458 14850458 1140.5 < 2.2e-16 \*\*\*  
## Residuals 233 3033805 13021   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

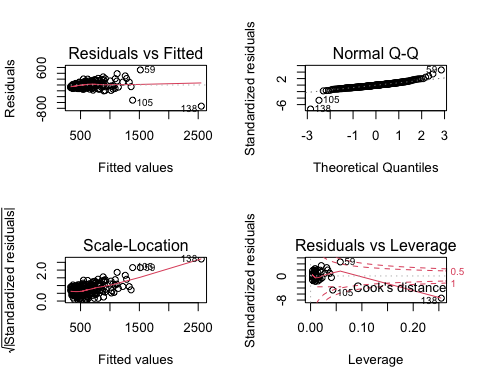
#  
coef(ols)

## (Intercept) RENTA   
## 147.4753885 0.4851784

confint(ols)

## 2.5 % 97.5 %  
## (Intercept) 116.0367916 178.913985  
## RENTA 0.4568738 0.513483

# Gráficas de diagnóstico:  
par(mfrow=c(2,2))  
plot(ols)



par(mfrow=c(1,1))  
#  
ols$coefficients

## (Intercept) RENTA   
## 147.4753885 0.4851784

(b1 <- coef(ols)[[1]])

## [1] 147.4754

(b2 <- coef(ols)[[2]])

## [1] 0.4851784

vcov(ols)

## (Intercept) RENTA  
## (Intercept) 254.6283412 -0.2027754824  
## RENTA -0.2027755 0.0002063929

(varb1 <- vcov(ols)[1, 1])

## [1] 254.6283

(varb2 <- vcov(ols)[2, 2])

## [1] 0.0002063929

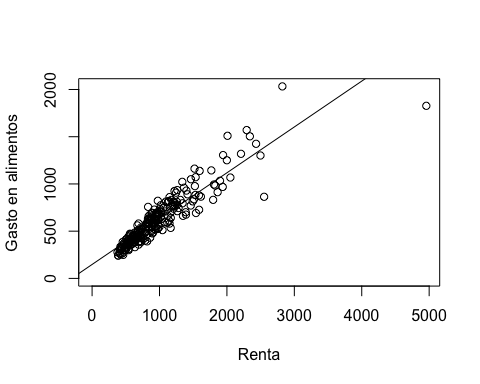
(covb1b2 <- vcov(ols)[1,2])

## [1] -0.2027755

#  
# Resultados formateados:  
stargazer(ols, type = "text", title = "Resultados de la regresión")

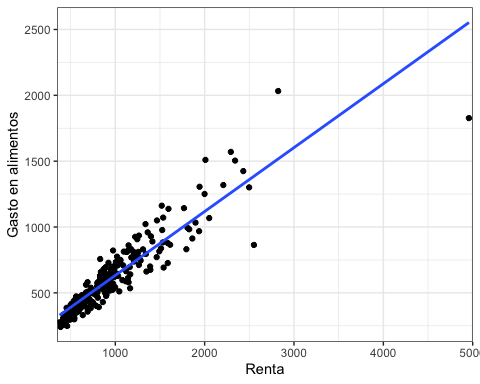
##   
## Resultados de la regresión  
## ===============================================  
## Dependent variable:   
## ---------------------------  
## GALIM   
## -----------------------------------------------  
## RENTA 0.485\*\*\*   
## (0.014)   
##   
## Constant 147.475\*\*\*   
## (15.957)   
##   
## -----------------------------------------------  
## Observations 235   
## R2 0.830   
## Adjusted R2 0.830   
## Residual Std. Error 114.108 (df = 233)   
## F Statistic 1,140.534\*\*\* (df = 1; 233)   
## ===============================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

# Diagrama de puntos y línea de regresión:  
# Método estándar  
#  
plot(engel$RENTA, engel$GALIM,  
 ylim=c(0, max(engel$GALIM)),  
 xlim=c(0, max(engel$RENTA)),  
 xlab="Renta",  
 ylab="Gasto en alimentos",  
 type = "p")  
abline(ols)



# También podría usarse:  
# b1 <- coef(ols)[[1]]  
# b2 <- coef(ols)[[2]]  
# abline(b1,b2)  
# Método ggplot  
#  
ggplot(engel, aes(x = RENTA, y = GALIM)) + geom\_point() + geom\_smooth(method = "lm", se = FALSE) + scale\_x\_continuous(limits = c(350, 5000), expand = c(0, 0)) + theme\_bw() + labs(x = "Renta", y = "Gasto en alimentos")

## `geom\_smooth()` using formula 'y ~ x'



#  
# Predicción con el modelo de regresión simple:  
new\_RENTA <- data.frame(RENTA=c(400, 2000, 4500))  
pred\_GALIM <- predict(ols, new\_RENTA)  
names(pred\_GALIM) <-c("Renta = 400", "2000", "4500")  
pred\_GALIM

## Renta = 400 2000 4500   
## 341.5468 1117.8322 2330.7783

# Modelo cuadrático en la variable RENTA  
waldtest(ols, . ~ . + I(RENTA^2))

## Wald test  
##   
## Model 1: GALIM ~ RENTA  
## Model 2: GALIM ~ RENTA + I(RENTA^2)  
## Res.Df Df F Pr(>F)   
## 1 233   
## 2 232 1 64.16 5.546e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

ols2 <- lm(GALIM ~ RENTA + I(RENTA^2), data = engel)  
summary(ols2)

##   
## Call:  
## lm(formula = GALIM ~ RENTA + I(RENTA^2), data = engel)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -525.99 -48.31 4.05 50.69 546.56   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.006e+00 2.244e+01 0.357 0.722   
## RENTA 7.100e-01 3.083e-02 23.031 < 2e-16 \*\*\*  
## I(RENTA^2) -6.603e-05 8.243e-06 -8.010 5.55e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 101.2 on 232 degrees of freedom  
## Multiple R-squared: 0.8671, Adjusted R-squared: 0.866   
## F-statistic: 756.9 on 2 and 232 DF, p-value: < 2.2e-16

# Modelo semiparamétrico  
plm <- lm(GALIM ~ bs(RENTA, df = 5) , data = engel)  
summary(plm)

##   
## Call:  
## lm(formula = GALIM ~ bs(RENTA, df = 5), data = engel)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -604.30 -46.06 2.03 49.76 434.65   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 270.56 41.76 6.479 5.58e-10 \*\*\*  
## bs(RENTA, df = 5)1 55.91 66.58 0.840 0.402   
## bs(RENTA, df = 5)2 235.58 42.05 5.603 6.03e-08 \*\*\*  
## bs(RENTA, df = 5)3 942.79 97.04 9.715 < 2e-16 \*\*\*  
## bs(RENTA, df = 5)4 2060.05 198.08 10.400 < 2e-16 \*\*\*  
## bs(RENTA, df = 5)5 1568.16 108.90 14.401 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 100.4 on 229 degrees of freedom  
## Multiple R-squared: 0.8708, Adjusted R-squared: 0.868   
## F-statistic: 308.8 on 5 and 229 DF, p-value: < 2.2e-16

# Elección de df  
# bs <- lapply(3:10, function(i) lm(GALIM ~ bs(RENTA, df = i) , data = engel))  
# structure(sapply(bs, AIC, k = log(nrow(engel))), .Names = 3:10)  
#  
par(mar = c(5, 5, 2, 4))  
engel\_sim <- data.frame(RENTA = 350:5000)  
engel\_sim$GALIMhat1 <- predict(ols2, newdata = engel\_sim)  
engel\_sim$GALIMhat2 <- predict(plm, newdata = engel\_sim)

## Warning in bs(RENTA, degree = 3L, knots = c(`33.33333%` = 715.3700777,  
## `66.66667%` = 1014.153953: some 'x' values beyond boundary knots may cause ill-  
## conditioned bases

plot(GALIM ~ jitter(RENTA, factor = 3), pch = 19, cex = 1.5, col = rgb(0.5, 0.5, 0.5, alpha = 0.02), data = engel)  
lines(GALIMhat1 ~ RENTA, data = engel\_sim, lty = 2)  
lines(GALIMhat2 ~ RENTA, data = engel\_sim)  
legend("topleft", c("Regresión cuadrática", "Regresión spline"),  
 lty = c(2, 1), bty = "n")

