Regression in R

Justin Smith

2023-10-30

Background

- OLS regression is the workhorse in econometrics
- Even when more advanced techniques are used, OLS is often included as a benchmark
- In this tutorial we will learn
  - How to estimate parameters by OLS
  - Export them in a readable format

Population Regression Model

- Suppose the population regression is

$$y = \mathbf{x}\boldsymbol{\beta} + u$$

- Where
  - y is the outcome variable
  - $\mathbf{x}$ is a vector of independent variables
  - $\boldsymbol{\beta}$ is the corresponding vector of slopes
  - u is the population residual
- Remember that the population regression slope vector is

$$\boldsymbol{\beta} = (\mathbf{E}[\mathbf{x}'\mathbf{x}])^{-1}\mathbf{E}[\mathbf{x}'y]$$

Ordinary Least Squares

- Suppose we collect a random sample of n people on all variables
- Collect the values of the dependent variable into a column vector $\mathbf{y}$
- Arrange similar column vectors for each x into a matrix $\mathbf{X}$
- The OLS estimator replaces the population values with consistent estimates from this sample
- We saw that this is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'y$$

Estimating $\hat{\boldsymbol{\beta}}$ in R

- The main function to estimate $\hat{\boldsymbol{\beta}}$ is `lm()` from the `stats` package
- As an example, we can load the mtcars data and regress miles per gallon on weight

```
cardata <- mtcars
lm(mpg ~wt, data = cardata)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = cardata)
##
## Coefficients:
## (Intercept)           wt
##      37.285       -5.344
```

- This returns some very basic information including only the parameter estimates
- But the command can save significantly more information
- To see, save the regression as its own object

```
cardata <- mtcars
reg<-lm(mpg ~wt, data = cardata)
str(reg)
```

```
## List of 12
##  $ coefficients : Named num [1:2] 37.29 -5.34
##   ..- attr(*, "names")= chr [1:2] "(Intercept)" "wt"
##  $ residuals    : Named num [1:32] -2.28 -0.92 -2.09 1.3 -0.2 ...
##   ..- attr(*, "names")= chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##  $ effects      : Named num [1:32] -113.65 -29.116 -1.661 1.631 0.111 ...
##   ..- attr(*, "names")= chr [1:32] "(Intercept)" "wt" "" "" ...
##  $ rank         : int 2
##  $ fitted.values: Named num [1:32] 23.3 21.9 24.9 20.1 18.9 ...
##   ..- attr(*, "names")= chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##  $ assign       : int [1:2] 0 1
##  $ qr           :List of 5
##   ..$ qr   : num [1:32, 1:2] -5.657 0.177 0.177 0.177 0.177 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##   .. .. ..$ : chr [1:2] "(Intercept)" "wt"
##   .. ..- attr(*, "assign")= int [1:2] 0 1
##   ..$ qraux: num [1:2] 1.18 1.05
##   ..$ pivot: int [1:2] 1 2
##   ..$ tol  : num 1e-07
##   ..$ rank : int 2
##   ..- attr(*, "class")= chr "qr"
##  $ df.residual  : int 30
##  $ xlevels      : Named list()
##  $ call         : language lm(formula = mpg ~ wt, data = cardata)
##  $ terms        :Classes 'terms', 'formula'  language mpg ~ wt
##   .. ..- attr(*, "variables")= language list(mpg, wt)
##   .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "mpg" "wt"
##   .. .. .. ..$ : chr "wt"
##   .. ..- attr(*, "term.labels")= chr "wt"
##   .. ..- attr(*, "order")= int 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(mpg, wt)
##   .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. ..- attr(*, "names")= chr [1:2] "mpg" "wt"
##  $ model        :'data.frame':   32 obs. of  2 variables:
##   ..$ mpg: num [1:32] 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##   ..$ wt : num [1:32] 2.62 2.88 2.32 3.21 3.44 ...
##   ..- attr(*, "terms")=Classes 'terms', 'formula'  language mpg ~ wt
##   .. .. ..- attr(*, "variables")= language list(mpg, wt)
##   .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:2] "mpg" "wt"
##   .. .. .. .. ..$ : chr "wt"
##   .. .. ..- attr(*, "term.labels")= chr "wt"
##   .. .. ..- attr(*, "order")= int 1
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. .. ..- attr(*, "predvars")= language list(mpg, wt)
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. .. ..- attr(*, "names")= chr [1:2] "mpg" "wt"
##  - attr(*, "class")= chr "lm"
```

- This object stores a list of 12 things including
    - Coefficients
    - Residuals
    - Fitted values
- But there are things missing, like
    - Standard errors
    - Measures of fit
- To get measures of fit, we can apply the `summary()` command to our regression

```r
cardata <- mtcars
reg<-lm(mpg ~wt, data = cardata)
sumreg<-summary(reg)
str(sumreg)
```

```
## List of 11
##  $ call         : language lm(formula = mpg ~ wt, data = cardata)
##  $ terms        :Classes 'terms', 'formula'  language mpg ~ wt
##   .. ..- attr(*, "variables")= language list(mpg, wt)
##   .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "mpg" "wt"
##   .. .. .. ..$ : chr "wt"
##   .. ..- attr(*, "term.labels")= chr "wt"
##   .. ..- attr(*, "order")= int 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(mpg, wt)
##   .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. ..- attr(*, "names")= chr [1:2] "mpg" "wt"
##  $ residuals    : Named num [1:32] -2.28 -0.92 -2.09 1.3 -0.2 ...
##   ..- attr(*, "names")= chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##  $ coefficients : num [1:2, 1:4] 37.285 -5.344 1.878 0.559 19.858 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "(Intercept)" "wt"
##   .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"
##  $ aliased      : Named logi [1:2] FALSE FALSE
##   ..- attr(*, "names")= chr [1:2] "(Intercept)" "wt"
##  $ sigma        : num 3.05
##  $ df           : int [1:3] 2 30 2
##  $ r.squared    : num 0.753
##  $ adj.r.squared: num 0.745
##  $ fstatistic   : Named num [1:3] 91.4 1 30
##   ..- attr(*, "names")= chr [1:3] "value" "numdf" "dendf"
##  $ cov.unscaled : num [1:2, 1:2] 0.38 -0.1084 -0.1084 0.0337
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "(Intercept)" "wt"
##   .. ..$ : chr [1:2] "(Intercept)" "wt"
##  - attr(*, "class")= chr "summary.lm"
```

- This new object saves several more things, including
  - Coefficients
  - Residuals
  - Standard errors
  - R^2
- You can access these things directly if necessary
- For example, if I wanted the R^2 I could type

```
cardata <- mtcars
reg<-lm(mpg ~wt, data = cardata)
sumreg<-summary(reg)
sumreg$r.squared
```

```
## [1] 0.7528328
```

- Note that $ is a way to subset dataframes or lists (as an alternative to `select()`)

Stargazer

- Mostly you will not access elements of the regression individually
- There are packages to output nice looking tables
- The main one is **stargazer**
- The example below outputs a basic text table

```
cardata <- mtcars
reg<-lm(mpg ~wt, data = cardata)
stargazer(reg, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                 mpg
## -----------------------------------------------
## wt                           -5.344***
##                               (0.559)
##
## Constant                     37.285***
##                               (1.878)
```

```
##
## -----------------------------------------------
## Observations                    32
## R2                             0.753
## Adjusted R2                    0.745
## Residual Std. Error       3.046 (df = 30)
## F Statistic           91.375*** (df = 1; 30)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

- This outputs the coefficients and some summary statistics for the regression
- You can customize what appears in the table
- The following removes the dependent variable caption, variable labels, keeps only the number of observations and R^2, and gives a title

```r
cardata <- mtcars
reg<-lm(mpg ~wt, data = cardata)
stargazer(reg, type = "text", dep.var.caption = "", covariate.labels = c("Intercept", "Weight"),keep.sta
```

```
##
## Regression of MPG on WT
## ========================================
##                           mpg
## ----------------------------------------
## Intercept             -5.344***
##                        (0.559)
##
## Weight                37.285***
##                        (1.878)
##
## ----------------------------------------
## Observations             32
## R2                      0.753
## ========================================
## Note:        *p<0.1; **p<0.05; ***p<0.01
```

- For many applications, you do not want a text output format
- In .qmd documents you will likely want **latex** or **html**
- To change that, just change the type

```r
cardata <- mtcars
reg<-lm(mpg ~wt, data = cardata)
stargazer(reg, type = "latex", dep.var.caption = "", covariate.labels = c("Intercept", "Weight"),keep.s
```

```
##
## % Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac a
## % Date and time: Tue, Oct 31, 2023 - 10:07:09
## \begin{table}[!htbp] \centering
##   \caption{Regression of MPG on WT}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## \\[-1.8ex] & mpg \\
## \hline \\[-1.8ex]
##  Intercept & $-$5.344$^{***}$ \\
##   & (0.559) \\
##   & \\
##  Weight & 37.285$^{***}$ \\
##   & (1.878) \\
##   & \\
## \hline \\[-1.8ex]
## Observations & 32 \\
## R$^{2}$ & 0.753 \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{1}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

- This looks ugly, but is easily interpreted by markdown in your document
- Finally you can pick a specific style to taylor your output to a particular journal
- Suppose we want to output in the Quarterly Journal of Economics style

```
cardata <- mtcars
reg<-lm(mpg ~wt, data = cardata)
stargazer(reg, type = "text", style = "qje", dep.var.caption = "", covariate.labels = c("Intercept", "We
```

```
##
## Regression of MPG on WT
## ================================================
##                            mpg
## ------------------------------------------------
## Intercept               -5.344***
##                          (0.559)
##
## Weight                  37.285***
##                          (1.878)
##
## N                          32
## R2                        0.753
## ================================================
## Notes:    ***Significant at the 1 percent level.
##            **Significant at the 5 percent level.
##            *Significant at the 10 percent level.
```