# Linear and IV Regression in R

Aileen Brown Cuevas
Thomas Wiemann
*University of Chicago*

July 30, 2021

# Linear Regression in R

R provides a convenient built-in function for linear regression: `lm` (short for linear model, which is a term from statistics).

Suppose we want to run a linear regression of the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + U, \tag{1}$$

where $\beta_j, j \in \{0, 1, 2, 3\}$ are the unknown (fixed) coefficients of interest.

Suppose further that our data $\{(Y_i, X_{i1}, X_{i2}, X_{i3})\}_{i=1}^N$ is stored in a `dataframe` in R called `df`. We may then run the linear regression by calling `lm`. To obtain coefficient estimates, corresponding standard errors, as well as the $R^2$ and adjusted $R^2$, call `summary` on the `lm`-object.

Example: `lm`

```
fit_lm <- lm(y ~ 1 + X.1 + X.2 + X.3, data = df)
summary(fit_lm) # get coefficient values and standard errors
```

# Linear Regression in R

## Sample `lm` Output

```
fit_lm <- lm(y ~ 1 + X.1 + X.2 + X.3, data = df)
summary(fit_lm) # get coefficient values and standard errors

#> Call:
#> lm(formula = y ~ 1 + X.1 + X.2 + X.3, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -7.0028 -1.5431  0.2806  1.6243  6.4738
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  6.15992    0.13734  44.852   <2e-16 ***
#> X.1          1.98620    0.02628  75.588   <2e-16 ***
#> X.2          1.01089    0.03989  25.344   <2e-16 ***
#> X.3          0.01939    0.04063   0.477    0.633
#> ---
#> Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#>
#> Residual standard error: 2.32 on 1616 degrees of freedom
#> Multiple R-squared:  0.7997, Adjusted R-squared:  0.7994
#> F-statistic:  2151 on 3 and 1616 DF,  p-value: < 2.2e-16
```

# Linear Regression in R

Note that the built-in `lm` and `summary` only provide standard errors under the homosekedasticity assumption. In practice, we rarely think that assumption is plausible. To calculate heteroskedasticity-robust standard errors, we need to load additional packages: `sandwich` and `lmtest`.

Given the `lm`-object we computed above (called `fit_lm` here), we can the calculate "robust" standard errors in a straightforward manner using the `coeftest` function.

Heteroskedasticity Robust Standard Errors

```
# Install packages only the first time.
install.packages(c("sandwich", "lmtest"))

# Load packages
library(sandwich)
library(lmtest)

# Compute HC-robust standard errors
coeftest(fit_lm, vcov = vcovHC(fit_lm, type = "HC1"))
```

# Linear Regression in R (Contd.)

We have uploaded a quick R script that illustrates how to run linear regression in R.

Check it out: `https://thomaswiemann.github.io/assets/teaching/Spring2020-Econ-21020/linear_regression.R`

Bonus: The script also illustrates how one may code up a linear regression procedure from scratch and calculate heteroskedasticity-robust standard errors without using any packages. May be interesting for the enthusiastic programmers among you.

# Two Stage Least Squares in R

R does not provide a built-in function for two stage least squares (TSLS). This leaves us with two options: 1) code it up ourselves or 2) install a package.

Coding TSLS in R is excellent practice and ensures that you have fully understood the method yourself. You're highly encouraged to give it a try. Send an email should you get stuck!

A less time and effort-intensive approach is opting for an R package with an out-of-the-box TSLS implementation. An excellent choice here is the `AER` package (short for "Applied Econometrics with R").

To calculate heteroskedasticity robust standard errors, we can make use of the `sandwich` and `lmtest` packages discussed last time.

# Two Stage Least Squares in R (Contd.)

Suppose we are interested in the following instrument variable specification

$$\begin{aligned} Y &= \beta_0 + \tau D + \beta_x X + U, \\ D &= \alpha_0 + \alpha_z Z + \alpha_x X + V, \end{aligned} \tag{2}$$

where $D$ is the endogenous variable of interest, $X$ is an exogenous variable (included instrument), and $Z$ is the instrument (excluded instrument). Suppose further that our data $\{(Y_i, D_i, X_i, Z_i)\}_{i=1}^N$ is stored in a `dataframe` in R called `df`.

Heteroskedasticity Robust Standard Errors for TSLS

```
# Install packages only the first time.
install.packages(c("sandwich", "lmtest", "AER"))

# Load packages
library(sandwich); library(lmtest); library(AER)

# Estimate TSLS
tsls_fit <- ivreg(y ~ 1 + D + X | X + Z, data = df)

# Compute HC-robust standard errors
coeftest(tsls_fit, vcov = vcovHC(tsls_fit, type = "HC1"))
```