

# Economía Aplicada

## Manejo de Datos en gret1

---

Departamento de Economía  
Universidad Carlos III de Madrid

# Outline

- 1 Presentación resumida
- 2 MCO en gretl

# Comandos básicos y funciones

## Comandos básicos y funciones para estimaciones MCO

- `ols`: computa MCO
- `$coeff`: vector columna que contiene los coeficientes estimados en el último modelo
- `$yhat`: función que computa los valores predichos de la variable dependiente.
- `$uhat`: función que computa los residuos de la última regresión.
- `$sample`: función que identifica las observaciones usadas en la estimación.
- `omit/add`: contrastes de significación.
- `restrict`: contrastes de restricciones más generales.

*ols depvar indepvars* `--robust` `--simple-print`  
`--print-final`

- no hay que olvidarse de la constante *const* en la lista de regresores
- `--robust`: errores estándar correctos bajo nuestros supuestos (NO OLVIDARSE!); si no, computa errores bajo homoscedasticidad
- `--simple-print`: muestra solamente el  $R^2$  luego de la tabla principal
- `--print-final`: (con loops) el resultado se muestra en la iteración final
- Puede ser muy útil usar listas con *list*: hacer listas es útil para que los scripts sean más claros y que las modificaciones sean más fáciles.

$$wages = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 expersq + u$$

- `ols wages const educ exper expersq --robust --simple-print`
- ejemplo usando "list"
  - `list indepvar = educ exper expersq`
  - `ols wage const indepvar --robust`

## Usando los coeficientes estimados con MCO

Después de *ols*, podemos trabajar en nuestro script con los coeficientes estimados

- `$coeff`: devuelve un vector columna que contiene los coeficientes estimados en el último modelo.
- `$coeff(x1)`: el estimador para la pendiente de la variable  $x_1$ .
- si guardamos estos valores en un vector, podremos acceder luego usando notación matricial.

### un ejemplo simple

- `ols wage const educ exper expersq --robust`
- `b = $coeff`
- `genr ypredicho = b[1]+b[2]*educ+b[3]*exper + b[4]*expersq`
- `genr residuos = wages - ypredicho`

## Más opciones después de MCO

Después de *ols*, una serie de variables pueden recuperarse

- `$yhat`: valores predichos usando la última estimación.
- `$uhat`: residuos de la última estimación.
- `$sample`: 1: usado , 0: en la muestra, pero no usado, NA: fuera de la muestra.

ejemplo usando “list”

- `list indepvar = educ exper expertsq`
- `ols wage const indepvar --robust`
- `genr olssmpl = $sample`
- `genr uhat = $uhat`

## omit *varlist* --wald --quiet

- debe seguir a un comando de estimación como *ols*.
- contrasta la significación conjunta de *varlist* usando por defecto un test F.
- el modelo restringido reemplaza al original como modelo activo.
- --wald: estadístico de Wald ( $\chi^2$ )
- --quiet: solamente muestra el resultado del test
- los resultados pueden recuperarse usando \$test y \$pvalue

ols wage const educ exper expersq --robust

- omit exper expersq --wald

## add *varlist* `--quiet`

- debe seguir a un comando de estimación como *ols*.
- *varlist* se agrega al modelo anterior y un nuevo modelo es estimado.
- computa un test F (y su valor p) para la significación de *varlist*
- `--quiet`: solamente muestra el estadístico y su valor p.

## `ols wage const educ --robust`

- `add exper expersq`

## restrict --quiet --bootstrap

```
restrict --quiet --bootstrap
ecuaciones correspondientes a las restricciones
end restrict
```

- después de *ols*, computa un test de Wald (a menos que usemos `--bootstrap`)
- cada restricción lineal se escribe como una ecuación separada.
- los parámetros se referencian usando `b[j]` o `b[varname]`
- `--quiet`: solamente muestra el estadístico y su valor p.

- restrict
- $b[3] - 2*b[4] = 0$
- end restrict