



# Regresión Lineal Múltiple

**Sesión N° 8**

13 octubre 2021

**Análisis de datos estadísticos en R**

**Profesora** Valentina Andrade de la Horra

**Ayudantes** Dafne Jaime y Nicolás Godoy

# Contenidos Sesión 8



**Construcción modelo regresión lineal con lm**

**Modelamiento con srvyr y glm**

**Predictores categóricos**

**Transformaciones funcionales**

# Contenidos Sesión 8



**Creación exploratoria**

**Extraer información**

**Representación gráfica**

*Tablas*

*Gráficos*

# Modelo de regresión lineal



# Modelo de regresión lineal



- Herramienta de análisis poderosa en ciencias sociales
- Permite generalizar relaciones entre variables

--

$Y$	$X$
Explicada	Explicativa
Dependiente	Independiente
Respuesta	Control
Predicha	Predictora

# Modelo de regresión lineal



## Algunas propiedades básicas y clave

1.  $Y$  de carácter continuo
  2.  $X_i$  relación lineal con  $Y$  (*Supuesto N°2 Teorema Gauss Markow*)
  3. Muestreo aleatorio para recolección de datos
- Otros: *colinealidad imperfecta, homocedasticidad, esperanza condicional de residuos cero, i.i.d.*

# RLM en R con lm()



- `lm()` es del paquete base

```
lm(formula = ,  
    data = ,  
    weights = )
```

- Por lo general, **crearemos objetos** para luego manipularlos

```
modelo1 <- lm(formula = ,  
               data = ,  
               weights = )
```

# RLM en R con lm()



## Modelo sin predictores

```
modelo1 <- lm(y ~ 1,  
  data = datos,  
  weights = ponderador)
```

## Modelo simple

```
modelo1 <- lm(y ~ x1,  
  data = datos,  
  weights = ponderador)
```



# RLM en R con lm()



## Paso a paso

1. Un buen procesamiento de datos
2. Identificar variable *dependiente* e *independientes*
3. Seguir la formula  $y \sim x_1 + x_2 + \dots x_k$
4. No olvidar otros objetos *básicos* para construir modelo: base y pesos muestrales (veremos los problemas asociados a no considerarlos)
5. Crear objeto

# Modelamiento con glm y svyglm





# ¿Y no bastaba con eso?

No, y ya entenderás la razón

# RLM en R con glm()



- Muchas veces encontrarás tutoriales donde **no se incorporan los pesos muestrales en los modelos**
- Como sabemos, eso trae errores de estimación y se pasa por alto uno de los supuestos de Gauss Markow y de la FRP (*función de regresión poblacional*)

# glm() para regresiones lineales



- Un punto antes de revisar la construcción con survey es que podemos construir un modelo lineal ( $\text{lm}()$ ) con un modelo lineal generalizado ( $\text{glm}()$ )
- $\text{glm}()$  nos permite construir distintos tipos de modelos según la distribución que siguen las variables aleatorias
- Un ejemplo son las que siguen distribuciones normales-gaussianas (como las lineales que se estiman por OLS) o las binomiales (como las logísticas que se estiman por MV).

# glm() para regresiones lineales



```
modelo_glm <- glm(y ~ x1 + x2,  
                  family = gaussian(link = "identity"),  
                  data = datos, weights = peso)
```

- Si luego comparamos con modelo normal `lm()` notaremos que llegaremos al mismo resultado

# ¿Y qué tiene que ver todo esto con survey?



¡Ahora verás!

# regresiones lineales con



- El paquete `survey` contiene una función llamada `svyglm` que permite incorporar el diseño de muestreo.
- Los pasos adicionales solo implican:
  1. Crear objeto de diseño de muestra con `as_survey_design`
  2. Crear objeto de modelo incorporando el diseño de la muestra. **¡La función `svyglm()` es muy parecida a las que ya vimos**




# regresiones lineales con



```
# Paso 1:
diseno_muestra <- as_survey_design(datos, ids = 1, weights = p
#Paso 2:
modelo_survey <- svyglm(y ~ x1 + x2,
  family = gaussian(link = "identity"),
  data = datos,
  design = diseno_muestra)
```

¡Notarás que dará el mismo resultado! (*en este caso...*)

**Pero si todos dan el   
mismo resultado... ¿por  
qué no solo ocupar  
lm()?**



# Transformaciones funcionales

# 1. Predictores categóricos



- En R los predictores categóricos pueden estar en clase `character` o `factor`
- Para que la regresión reconozca a estos como predictores categóricos y conserve tanto etiquetas como niveles ocuparemos `forcats::as_factor()` (*es y será un muy buen amigo*)

# 1. Predictores categóricos



- Gracias a este podremos notificar bien *cuál es la **categoría de referencia***. De manera adicional con `relevel()` podremos re-definir esta categoría.
- **Este procedimiento debe ir si o si en su código de procesamiento**

## 2. Modelos log y cuadráticos



- Si queremos transformar una variable a logaritmo, al cuadrado, re-escalarla, hacerla **interactuar** etc. **lo ideal es que creamos esa nueva variable en el procesamiento.**
- Ahora bien, en la realidad, vamos a querer ir testeando/explorando
- Una forma "*sencilla*" es introducir esa transformación al argumento de la fórmula de `lm()`

## 2. Modelos log y cuadráticos



```
# Modelo log-lineal
modelo_log <- lm(log(y) ~ x1 + x2 ,
                 data = datos, weights = fact_cal_esi)

#Modelo log-cuadrático
modelo_log_cuadrático <- lm(log(y) ~ x1 + (x1)^2 + x2 ,
                             data = datos, weights = fact_cal_esi)

#Modelo log interacción
modelo_log_interaccion <- lm(log(y) ~ x1 + x2 +x1*x2,
                              data = datos, weights = fact_cal_esi)
```

# Modelamiento exploratorio



- Sabemos que en análisis de datos existen los análisis **confirmatorios** y *exploratorios*
- En regresiones también existe esta clase de *approch* que se aplica a partir de método **stepwise** (paso a paso) con dirección "*backward*" y "*forward*"
- Los pasos en *R* son sencillos. Primero, construyo una fórmula general y luego defino la dirección del *stepwise*



# Modelamiento exploratorio



```
# Creación exploratoria
modelo_general <- lm(y ~ .,
                     data = datos, weights = datos$pesos,
                     na.action= na.omit)

step(modelo_general,
      direction = "forward"
      keep = nobs)
```



# Obteniendo información sobre mi modelo

# Un breve resumen



Función	Objetivo
<code>summary(modelo)</code>	Resumen general
<code>modelo\$coefficients</code>	Extraer elemento del objeto (coeficiente en este caso)
<code>modelo5\$coefficients[2]</code>	Extraer coeficiente N°2 de objeto
<code>modelo5\$coefficients["x1"]</code>	Extraer coeficiente "x1" del modelo
<code>str(summary(modelo1))</code>	Estructura del resultado del modelo
<code>summary(modelo5)\$fstatistic</code>	Estadístico F (podría ser R cuadrado también)

# Con sjPlot



```
get_model_data(modelo, type = c("pred"))
```

```
get_model_data(modelo, type = "pred",  
                terms = "sexo")
```

```
broom::augment(modelo) #puedo guardar como objeto
```





# Representación gráfica

# Tablas



- Hoy aprenderemos a crear tablas en *tab\_model* (de *sjPlot*). Ahora bien, el paquete *más versátil* para presentación de tablas de modelos es *texreg* (próxima semana)

```
tab_model(modelo0) # con uno
```

```
tab_model(list(modelo0, modelo1, modelo2))
```

```
tab_model(list(modelo0, modelo1, modelo2, modelo3, modelo4, modelo5  
  show.aic = T, # show.[inserte algun estadístico]  
  show.reflvl = T,  
  p.style = "stars")
```



## Forest plot

```
# Forest plot (default)  
plot_model(modelo5)  
plot_model(modelo5_log)  
  
# Forest plot - arreglado  
plot_model(modelo5_log,  
           show.p = T,  
           show.values = T)
```





## Marginal effects

```
plot_model(modelo,  
            type = "pred")
```

```
plot_model(modelo,  
            type = "pred",  
            terms = "grupo1")
```

```
plot_model(modelo5_log,  
            type = "pred",  
            terms = c("grupo1", "grupo2"))
```

# Recursos de la práctica



- Este práctico fue trabajado con datos de **Encuesta Suplementaria de Ingresos**.
- **Manual Metodológico ESI**

# En síntesis



**Construcción modelo regresión lineal con lm**

**Modelamiento con srvyr y glm**

**Predictores categóricos**

**Transformaciones funcionales**

**Creación exploratoria**

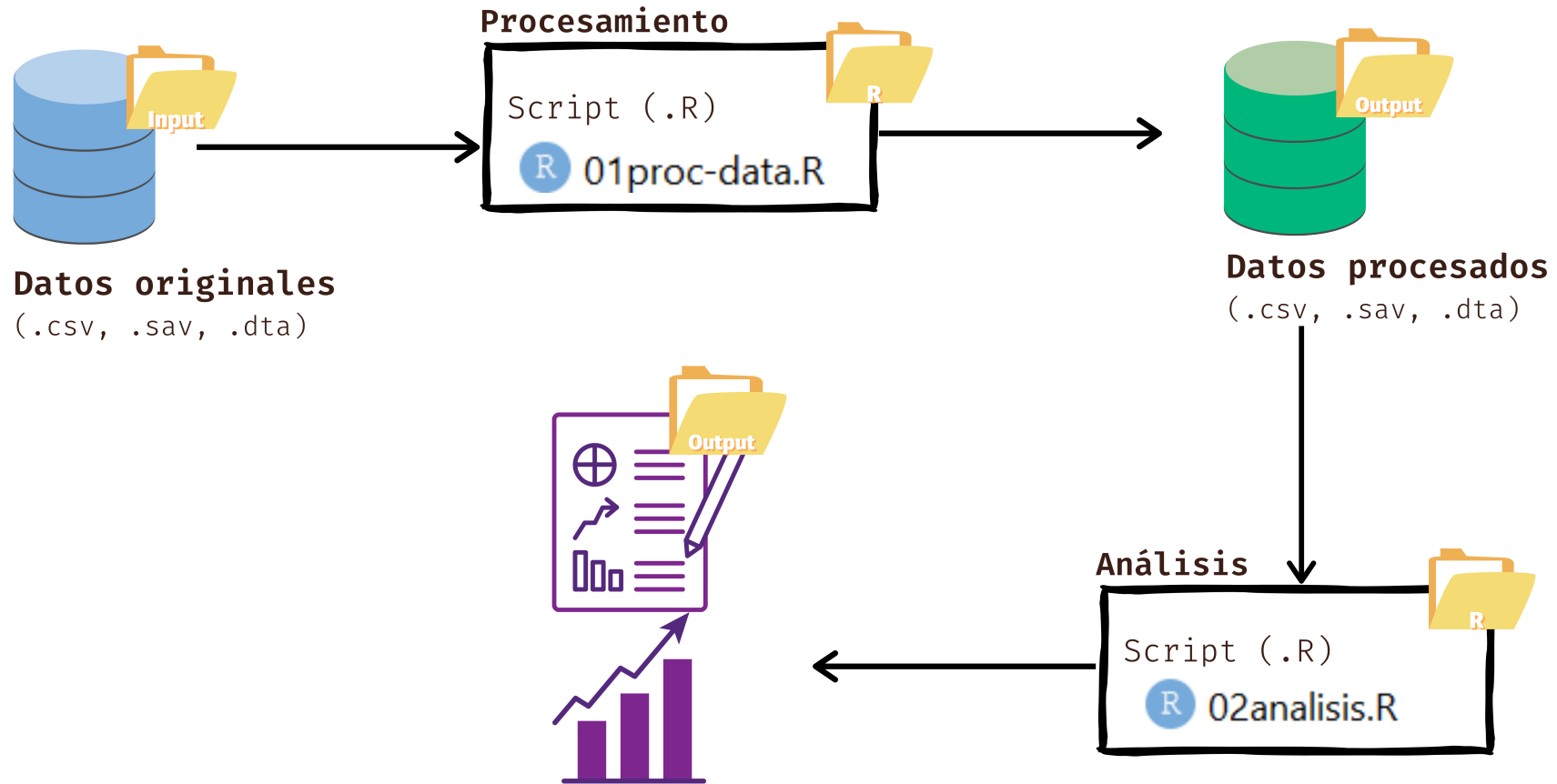
**Extraer información**

**Representación gráfica: Tablas y gráficos**

# ¡Y a no olvidar el flujo para el análisis!



Nos permite hacernos amigas/os más rápido del programa



¿Y eso era?



¡Ahora si que si! Nos vemos el próximo lunes





# Regresión Lineal Múltiple

**Sesión N° 8**

13 octubre 2021

**Análisis de datos estadísticos en R**

**Profesora** Valentina Andrade de la Horra

**Ayudantes** Dafne Jaime y Nicolás Godoy