



Regresión Logística Múltiple

Sesión N° 9

24 octubre 2021

Análisis de datos estadísticos en R

Profesora Valentina Andrade de la Horra
Ayudantes Dafne Jaime y Nicolás Godoy

Contenidos Sesión 9



Repaso de construcción de modelos

Predictores categóricos

Transformaciones funcionales

Modelamiento con `svyr` y `glm`

Modelos logísticos

En el práctico...



Representación gráfica

Tablas

Gráficos



Repaso

Modelo de regresión lineal



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()



Modelo multiple

```
modelo1 <- lm(y ~ x1 + x2,  
             data = datos,  
             weights = ponderador)
```



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 `r_lv_l()` 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_interacción <- lm(log(y) ~ x1 + x2 +x1*x2,
                              data = datos, weights = fact_cal_esi)
```

Modelamiento con glm y svyglm



Regresiones con glm()



- Modelo lineal generalizado (`glm()`)
- `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).

Family	Default link function	Link options
binomial	link = "logit"	"logit", "probit", "cauchit"
gaussian	link = "identity"	"identity", "log", "inverse"
poisson	link = "log"	"log", "identity", "sqrt"



glm() para regresiones lineales



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

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

Al igual que como vimos con `survey::svyglm()`

Se puede aplicar a regresiones logísticas

regresiones con `survey::svyglm()`



- 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 logísticas con



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



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)
<code>modelo5\$fitted.values</code>	Valores predichos



**¿Qué pasa si quiero
hacer cálculos en OR?**

De *log odds* a *OR*



- Debemos recordar que los coeficientes están en *log odds*
- Para obtener los valores en *OR* debo hacer un par de procedimientos.
- Cuando representamos las tablas con `sjPlot` esto es *mucho más fácil*

exp() exponenciar



```
# OR del coeficiente N°2  
exp(modelo$coefficients[2])
```

```
# Crear OR -----  
modelo$or <- exp(modelo$coefficients)
```

- Luego se puede incorporar en tablas hechas en *texreg*

Regresiones en R con `glm()`: Paso a paso**



1. Buen procesamiento de datos
2. Identificar variable *dependiente* e *independientes*

No olvidar que los predictores categóricos estén bien trabajados

1. Seguir la formula $y \sim x_1 + x_2 + \dots x_k$
2. Indicar familia, datos y pesos muestrales
3. Crear objeto
4. Explorar resultado con `summary(modelo1)`
5. Tabla o gráfico

Recursos de la práctica



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

En síntesis



Repaso de construcción de modelos

Predictores categóricos

Transformaciones funcionales

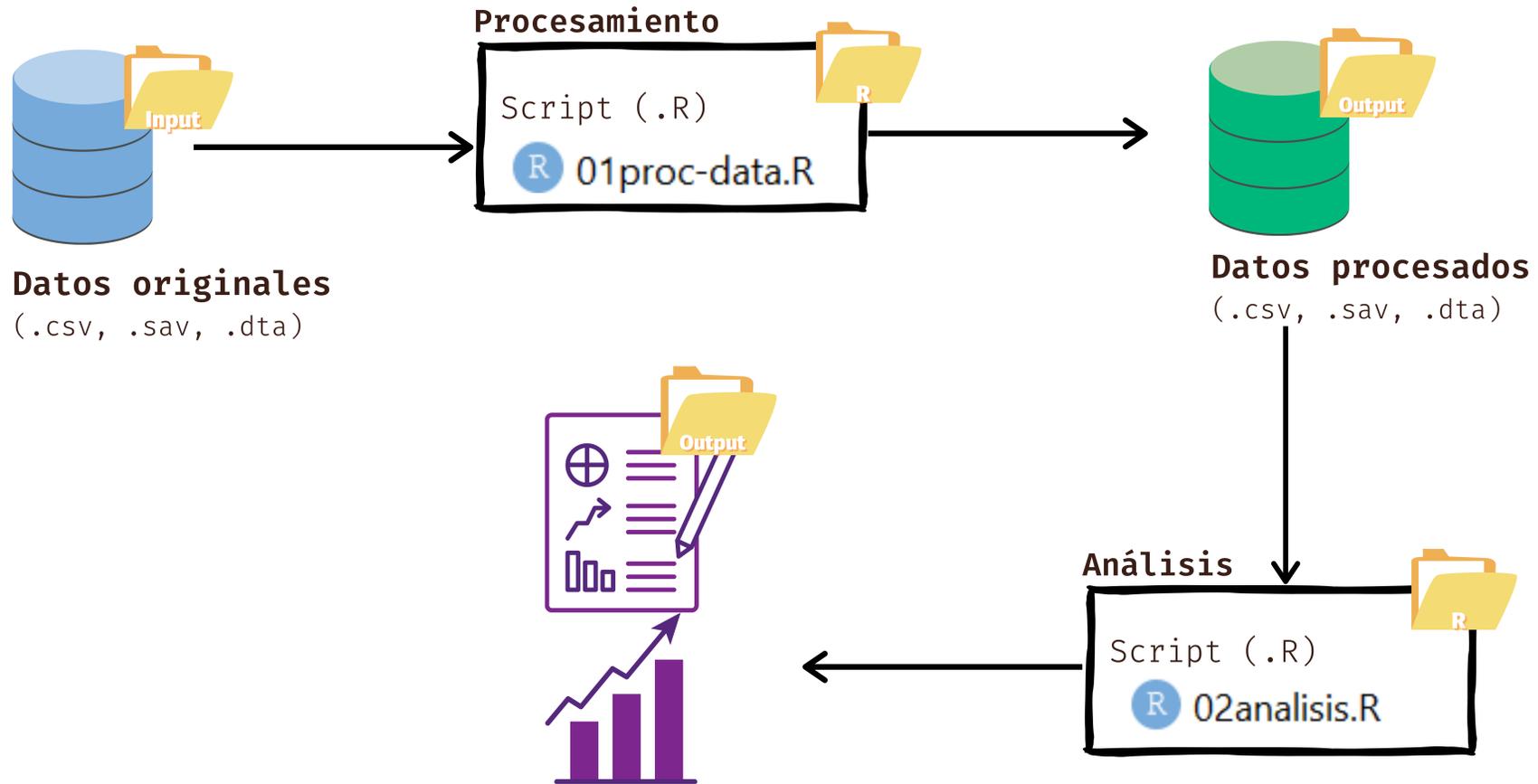
Modelamiento con `svyr` y `glm`

Modelos logísticos

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



Nos permite hacernos amigas/os más rápido del programa



Antes de finalizar...



Desafío(0.5 décimas para la tarea N°4)

Construir una tabla *única* con modelo de regresión lineal y logístico con los datos de la tarea N°4 con el paquete `texreg`. Debe especificar *bien* todos los predictores

Fecha: hasta el día miércoles a las 20 horas por canal #04-tareas.

Debe entregar el código de `texreg` y un pantallazo de la tabla

¡Ahora si que si! Nos vemos el próximo lunes





Regresión Logística Múltiple

Sesión N° 9

24 octubre 2021

Análisis de datos estadísticos en R

Profesora Valentina Andrade de la Horra

Ayudantes Dafne Jaime y Nicolás Godoy