Análisis de encuestas de hogares con R

Módulo 5: Modelos de regresión

CEPAL - Unidad de Estadísticas Sociales

Tabla de contenidos I

Modelos de regresión bajo diseños de muestreo complejos

Introducción

- Un modelo matemático es una relación funcional entre variables.
- El objetivo es encontrar modelos que relacionen variables de entrada con una variable de salida.
- A lo largo de la historia, varios autores han discutido el impacto de los diseños muestrales complejos en las inferencias relacionadas con modelos de regresión.

Introducción

- **Kish y Frankel (1974):** Fueron los primeros en abordar, de manera empírica, cómo los diseños muestrales complejos afectan las inferencias en modelos de regresión.
- **Fuller (1975):** Desarrolló un estimador de varianza que considera ponderaciones desiguales de observaciones, especialmente relevantes en contextos de muestreo complejo de dos etapas.
- **Sha et al. (1977):** Discutieron las violaciones de supuestos en modelos de regresión lineal y presentaron evaluaciones empíricas del desempeño de estimadores de varianza basados en la linealización para modelos de regresión lineal con datos de encuestas.
- **Binder (1983):** Se centró en las distribuciones muestrales de estimadores para parámetros de regresión en poblaciones finitas y estimadores de varianza relacionados.

Introducción

- **Skinner et al. (1989):** Trabajaron en estimadores de varianza para los coeficientes de regresión que permitieron diseños de muestras complejas, y recomendaron el uso de métodos de linealización u otros métodos para la estimación de la varianza.
- **Fuller (2002):** Ofreció un resumen de los métodos de estimación para modelos de regresión que involucran información relacionada con muestras complejas.
- **Pfeffermann (2011):** Discutió enfoques basados en el ajuste de modelos de regresión lineal a datos de encuestas de muestras complejas, respaldando el uso de un método "q-weighted."

Modelos de Regresión Lineal Simple y Múltiple

- Un modelo de regresión lineal simple se define como

$$y = \beta_0 + \beta_1 x + \varepsilon$$

.

- Los modelos de regresión lineal múltiples extienden este concepto para múltiples variables predictoras:

$$y = X\beta + \varepsilon$$

.

- El valor esperado de la variable dependiente condicionado a las variables independientes se representa como $E(y|x)$.

Consideraciones en Modelos de Regresión

- $E(\varepsilon_i|x_i) = 0$: El valor esperado de los residuos condicionado a las covariables es igual a 0.
- $Var(\varepsilon_i|x_i) = \sigma^2_{y,x}$: Homogeneidad de varianza, la varianza de los residuos condicionados es constante.
- $\varepsilon_i|x_i \sim N(0, \sigma^2_{y,x})$: Normalidad en los errores, los residuos condicionados se distribuyen normalmente.
- $cov(\varepsilon_i, \varepsilon_j|x_i, x_j)$: Independencia en los residuos, los residuos en diferentes sujetos no están correlacionados con los valores de sus variables predictoras.

Resultados para el modelo de regresión

Una vez definido el modelo de regresión lineal y sus supuestos, se puede deducir los siguiente:

$$\begin{aligned} \widehat{y} &= E\left(y \mid x\right) \\ &= E\left(x\beta\right) + E\left(\varepsilon\right) \\ &= x\beta + 0 \\ &= \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \end{aligned}$$

y Adicionalmente,

$$\begin{aligned} var\left(y_i \mid x_i\right) &= \sigma^2_{y,x}, \\ cov\left(y_i, y_j \mid x_i, x_j\right) &= 0 \\ &\text{y} \\ y_i &\sim N\left(x_i\beta, \sigma^2_{y,x}\right) \end{aligned}$$

Estimación de los parámetros en un modelo de regresión simple.

La estimación del coeficiente de regresión β_1 en un modelo de regresión simple con muestras complejas involucra el uso de ponderaciones y totales. El estimador $\widehat{\beta}_1$ se calcula como un cociente de totales ponderados.

$$\begin{aligned}\widehat{\beta}_1 &= \frac{\sum_{h}^{H}\sum_{\alpha}^{a_h}\sum_{i=1}^{n_{h\alpha}} \omega_{h\alpha i}\left(y_{h\alpha i}-\bar{y}_{\omega}\right)\left(x_{h\alpha i}-\bar{x}_{\omega}\right)}{\sum_{h}^{H}\sum_{\alpha}^{a_h}\sum_{i=1}^{n_{h\alpha}} \omega_{h\alpha i}\left(x_{h\alpha i}-\bar{x}_{\omega}\right)^2} \\ &= \frac{t_{xy}}{t_{x^2}}\end{aligned}$$

Varianza estimada

La varianza del estimador $\hat{\beta}_1$ se calcula considerando la varianza de los totales ponderados y sus covarianzas. Esta varianza estimada tiene en cuenta el diseño muestral y la estructura de ponderación.

$$var\left(\hat{\beta}_1\right) = \frac{var\left(t_{xy}\right) + \hat{\beta}_1^2 var\left(t_{x^2}\right) - 2\hat{\beta}_1 cov\left(t_{xy}, t_{x^2}\right)}{\left(t_{x^2}\right)^2}$$

Extensión a modelos de regresión múltiple:

Para modelos de regresión múltiple, la estimación de la varianza se generaliza a través de una matriz de varianza-covarianza que involucra los coeficientes de regresión.

$$var\left(\hat{\beta}\right) = \hat{\Sigma}\left(\hat{\beta}\right) = \begin{bmatrix} var\left(\hat{\beta}_0\right) & cov\left(\hat{\beta}_0, \hat{\beta}_1\right) & \cdots & cov\left(\hat{\beta}_0, \hat{\beta}_p\right) \\ cov\left(\hat{\beta}_0, \hat{\beta}_1\right) & var\left(\hat{\beta}_1\right) & \cdots & cov\left(\hat{\beta}_1, \hat{\beta}_p\right) \\ \vdots & \vdots & \ddots & \vdots \\ cov\left(\hat{\beta}_0, \hat{\beta}_p\right) & cov\left(\hat{\beta}_1, \hat{\beta}_p\right) & \cdots & var\left(\hat{\beta}_p\right) \end{bmatrix}$$

Este enfoque de estimación garantiza que se tengan en cuenta las particularidades del diseño muestral en la inferencia sobre los coeficientes de regresión.

Aplicación en encuestas de hogares

El proceso inicia con la lectura de la muestra y definición del objeto `survey.design`

```
options(survey.lonely.psu="adjust")
encuesta <- readRDS("../Data/encuesta.rds")
data("BigCity", package = "TeachingSampling")

library(srvyr)
diseno <- encuesta %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = T
  )
```

Sub-grupos

Dividir la muestra en sub-grupos de la encuesta.

```
sub_Urbano <- diseno %>%  filter(Zone == "Urban")
sub_Rural  <- diseno %>%  filter(Zone == "Rural")
sub_Mujer  <- diseno %>%  filter(Sex == "Female")
sub_Hombre <- diseno %>%  filter(Sex == "Male")
```

Scatterplot con los datos poblacionales

Para observar que existe una correlación entre el ingreso y el gasto es construido un scatterplot.

```
library(ggplot2); library(ggpmisc)
plot_BigCity <-
  ggplot(data = BigCity,
         aes(x = Expenditure, y = Income)) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE,
              formula = y ~ x) +
  theme_cepal()

plot_BigCity <- plot_BigCity +
  stat_poly_eq(formula = y~x,
  aes(label = paste(..eq.label..,
   ..rr.label.., sep = "~~~"),size = 3), parse = TRUE)
```

Scatterplot con los datos poblacionales

El resultado obtenido es:

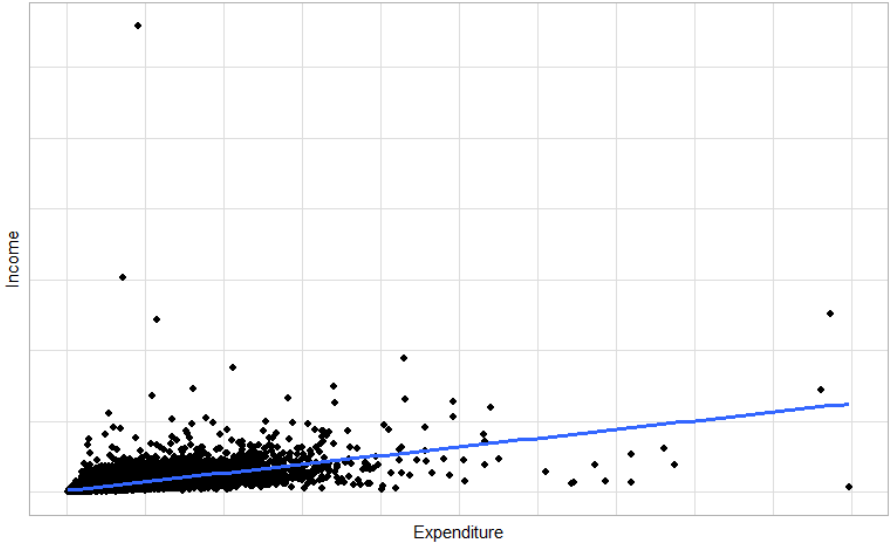


Figura 1: Relación del ingreso y el gasto en la población

Modelo poblacional

El modelo poblacional es estimado con `lm`

```
fit <- lm(Income ~ Expenditure, data = BigCity)
```

Tabla 1: Modelo BigCity

	Income
Expenditure	1.229***
	(0.004)
Constant	123.300***
	(1.988)
N	150266
R-squared	0.359
Adj. R-squared	0.359
Residual Std. Error	461.700 (df = 150264)
F Statistic	84053.000*** (df = 1; 150264)

***p < .01; **p < .05; *p < .1

Scatterplot con los datos encuesta sin ponderar

Una sintaxis similar permite construir el scatterplot en la muestra.

```
plot_sin <-
  ggplot(data = encuesta,
         aes(x = Expenditure, y = Income)) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE,
              formula = y ~ x) +
  theme_cepal()
plot_sin <- plot_sin + stat_poly_eq(formula = y~x,
  aes(label = paste(..eq.label..,
     ..rr.label.., sep = "~~~"), size = 5),
  parse = TRUE)
```

Scatterplot con los datos encuesta sin ponderar

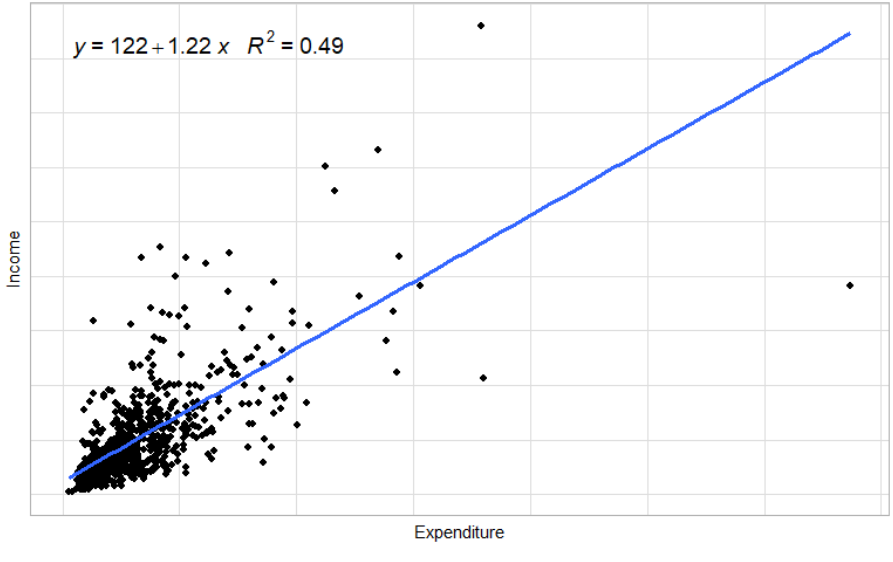


Figura 2: Relación del ingreso y el gasto en la muestra sin ponderar

Modelo sin ponderar

El modelo ignorando los factores de expansión quedas así:

```
fit_sinP <- lm(Income ~ Expenditure, data = encuesta)
stargazer(fit_sinP, header = FALSE,
          title = "Modelo encuesta Sin ponderar",
          style = "ajps")
```

Modelo sin ponderar

Tabla 2: Modelo encuesta Sin ponderar

	Income
Expenditure	1.220***
	(0.025)
Constant	121.500***
	(11.410)
N	2605
R-squared	0.487
Adj. R-squared	0.487
Residual Std. Error	345.000 (df = 2603)
F Statistic	2473.000*** (df = 1; 2603)

***p < .01; **p < .05; *p < .1

Scatterplot con los datos encuesta ponderado

Para que el gráfico tenga en cuenta las ponderaciones debe agregar `mapping = aes(weight = wk)` en la función `geom_smooth`.

```
plot_Ponde <-
  ggplot(data = encuesta,
         aes(x = Expenditure, y = Income)) +
  geom_point(aes(size = wk)) +
  geom_smooth(method = "lm",
              se = FALSE,
              formula = y ~ x,
              mapping = aes(weight = wk)) +
  theme_cepal()
plot_Ponde <- plot_Ponde + stat_poly_eq(formula = y~x,
  aes(weight = wk,
    label = paste(..eq.label..,
      ..rr.label.., sep = "~~~")),
  parse = TRUE,size = 5)
```

Scatterplot con los datos encuesta sin ponderar

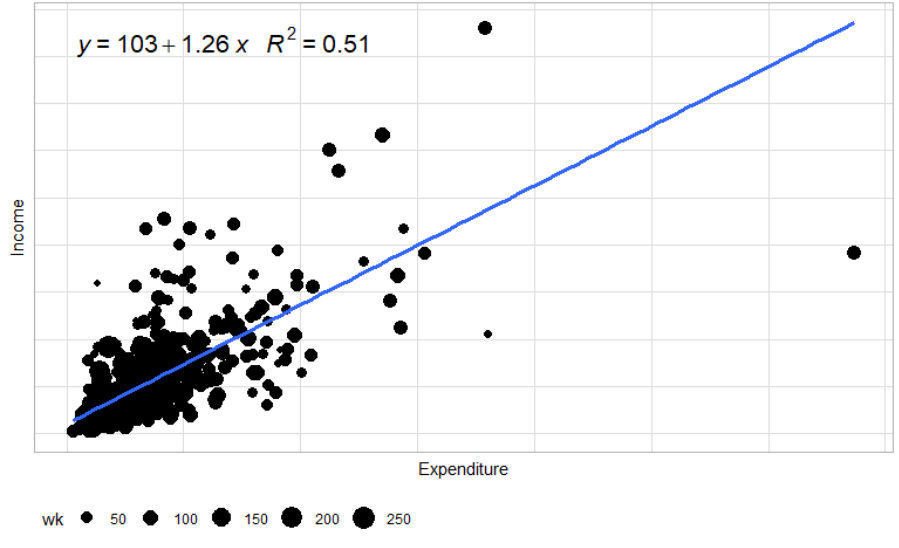


Figura 3: Relación del ingreso y el gasto en la muestra sin ponderar

Modelo ponderado `lm`

La función `lm` permite incluir los `weights` en la estimación de los coeficientes.

```
fit_Ponde <- lm(Income ~ Expenditure,
                data = encuesta, weights = wk)
stargazer(fit_Ponde, header = FALSE,
          title = "Modelo encuesta ponderada",
          style = "ajps")
```

Modelo ponderado lm

Tabla 3: Modelo encuesta ponderada

	Income
Expenditure	1.263***
	(0.024)
Constant	103.100***
	(11.260)
N	2605
R-squared	0.509
Adj. R-squared	0.509
Residual Std. Error	2627.000 (df = 2603)
F Statistic	2703.000*** (df = 1; 2603)

***p < .01; **p < .05; *p < .1

Modelo ponderado svyglm

Ahora, emplee la función `svyglm` de `survey`

```
fit_svy <- svyglm(Income ~ Expenditure,
                  design = diseno)
```

Resumen del Modelo

Tabla 4: Modelo encuesta ponderada, svyglm

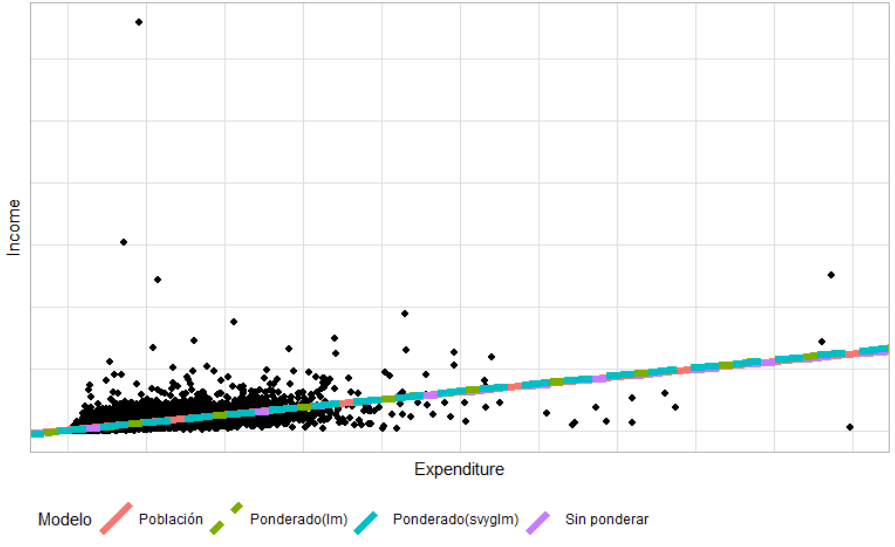
	Income
Expenditure	1.263***
	(0.191)
Constant	103.100
	(64.780)
N	2605
AIC	38281.000

***p $<$ .01; **p $<$ .05; *p $<$ .1

Comparando los resultados

```
df_model <- data.frame(
  intercept = c(coefficients(fit)[1],
                coefficients(fit_sinP)[1],
                coefficients(fit_Ponde)[1],
                coefficients(fit_svy)[1]),
  slope = c(coefficients(fit)[2],
                coefficients(fit_sinP)[2],
                coefficients(fit_Ponde)[2],
                coefficients(fit_svy)[2]),
  Modelo = c("Población", "Sin ponderar",
             "Ponderado(lm)", "Ponderado(svyglm)"))
plot_BigCity +  geom_abline( data = df_model,
    mapping = aes( slope = slope,
      intercept = intercept, linetype = Modelo,
      color = Modelo ), size = 2
  )
```

Comparando los resultados



Comparando los resultados

	Poblacion	Sin Pond	Ponde(lm)	Ponde(svyglm)
(Intercept)	123.337	121.516	103.136	103.136
	p = (<0.001)	p = (<0.001)	p = (<0.001)	p = (0.114)
Expenditure	1.229	1.220	1.263	1.263
	p = (<0.001)	p = (<0.001)	p = (<0.001)	p = (<0.001)
Num.Obs.	150266	2605	2605	2605
R2	0.359	0.487	0.509	0.509
R2 Adj.	0.359	0.487	0.509	-9.826
AIC	2270206.0	37841.7	38281.0	123.5
F	84052.758	2472.919	2702.978	43.885
RMSE	461.74	344.88	345.09	345.09

Diagnostico del modelo

Diagnostico del modelo

Adecuado Ajuste del Modelo: - Verificar que el modelo se ajuste adecuadamente a los datos recopilados en la encuesta. - Evaluar si la relación funcional especificada es apropiada para representar las variables de interés.

Normalidad de Errores: - Examinar si los errores del modelo siguen una distribución normal. - Esto es crucial para realizar pruebas de hipótesis precisas y estimar intervalos de confianza confiables.

Varianza Constante de Errores: - Asegurarse de que la varianza de los errores sea constante en todos los niveles de las variables independientes. - La heterocedasticidad puede impactar en las pruebas y la interpretación de coeficientes.

Diagnostico del modelo

Errores No Correlacionados: - Evaluar si los errores pueden considerarse no correlacionados entre sí. - La autocorrelación de errores puede afectar la eficiencia de las estimaciones.

Datos Influyentes: - Identificar valores atípicos o datos influyentes que tienen un efecto desproporcionadamente grande en el modelo de regresión. - Estos datos deben tratarse con precaución y su impacto debe ser evaluado.

Valores Atípicos (Outliers):

Estimación del R^2 y R^2_{adj}

- En análisis de regresión, el coeficiente de determinación (R^2) mide la variabilidad explicada por el modelo.
- El R^2_ω ajusta R^2 para muestras complejas, considerando ponderaciones de la muestra.
- R^2_ω se basa en la suma de cuadrados totales ponderada (WSST) y la suma de cuadrados del error ponderada (WSSE).
- La fórmula de R^2 es $1 - \frac{SSE}{SST}$, donde SSE es la suma de cuadrados del error y SST es la suma de cuadrados totales.

Estimación del R^2 y R^2_{adj}

- Para R^2_ω, la fórmula es $1 - \frac{WSSE}{WSST}$, considerando las ponderaciones de la muestra.

$$\widehat{WSSE}_\omega = \sum_{h}^{H} \sum_{\alpha}^{a_h} \sum_{i=1}^{n_{h\alpha}} \omega_{h\alpha i} \left(y_{h\alpha i} - x_{h\alpha i} \hat{\beta} \right)^2$$

- Se utiliza el coeficiente de determinación ajustado (R^2_{adj}) para tener en cuenta el tamaño de la muestra y el número de predictores en el modelo.
- R^2_{adj} se calcula como $1 - \frac{(n-1)}{(n-p)} R^2_\omega$, donde n es el tamaño de la muestra y p es el número de predictores.

Estimación del R^2 para el modelo del ingreso.

```
fit_svy <- svyglm(Income ~ Expenditure,
                  design = diseno,family=stats::gaussian())

medY <- diseno %>% summarise(medY = survey_mean(Income))

diseno %<>% mutate(
  ypred = fitted(fit_svy, type = "response"),
  medY = medY,
  sst = (Income - medY$medY)^2,
  sse = (ypred - medY$medY)^2
  )

diseno %>% summarise(WSST = survey_total(sst),
                     WSSE = survey_total(sse)) %>%
  transmute(WSST, WSSE, R2 = WSSE/WSST)
```

Estimación del R^2 para el modelo del ingreso

El resultado para el R^2 es

WSST	WSSE	R2
3.661e+10	1.865e+10	0.5094

De forma alternativa es:

```
modNul <- svyglm(Income ~ 1, design = diseno)
s1 <- summary(fit_svy)
s0 <-summary(modNul)

WSST<- s0$dispersion
WSSE<- s1$dispersion
R2 = 1- WSSE/WSST
R2
```

```
     variance    SE
[1,]    0.509 19005
```

Estimación del R^2_{adj} para el modelo del ingreso

Calculamos el R^2_{adj} utilizando la fórmula adecuada. Asegúrate de definir los valores de `n` y `p` de acuerdo a tu modelo.

```
n = nrow(encuesta)
p = 2
(R2Adj = 1 - ((n-1)/(n-p)) * R2)
```

```
     variance    SE
[1,]     0.49 19005
```

Metodología de los Q_Weighting de pfefferman

Cuando trabajamos con datos de encuestas que siguen un diseño muestral complejo y es posible aplicar la metodología de los q-weights **(Pffeferman, 2011)**.,

1. **Ajuste del Modelo de Regresión a los Q-Weights:** Inicialmente, ajustamos un modelo de regresión lineal a los q-weights en R. Esto se hace utilizando la función `lm()`.

```
fit_wgt <- lm(wk ~ Expenditure, data = encuesta)
```

2. **Obtención de Predicciones de Q-Weights:** A continuación, calculamos las predicciones de los q-weights para cada caso, utilizando las variables predictoras del modelo de regresión.

```
qw <- predict(fit_wgt)
```

Metodología de los Q_Weighting de pfefferman

3. **Creación de Nuevos Q-Weights:** Para obtener los q-weights ajustados, dividimos los weights originales por las predicciones calculadas en el paso anterior.

```
encuesta <- encuesta %>% mutate(wk1 = wk/qw)
```

4. **Definición de un Diseño Muestral con Q-Weights:** Usamos los nuevos q-weights obtenidos para definir un diseño muestral que refleje estos pesos.

```
diseno_qwgt <- encuesta %>%
     as_survey_design(
       strata = Stratum,
       ids = PSU,
       weights = wk1,
       nest = TRUE)
```

Modelos empleando los Q_Weighting

Estimando los coeficientes del modelo con los Q_Weighting de pfefferman

```
library(tidyr)
fit_svy_qwgt <- svyglm(Income ~ Expenditure,
                       design = diseno_qwgt)
s1_qwgt <- summary(fit_svy_qwgt)
tidy(fit_svy_qwgt)
```

term	estimate	std.error	statistic	p.value
(Intercept)	109.123	68.7290	1.588	0.115
Expenditure	1.246	0.2014	6.188	0.000

Calculo del R^2 y R^2_{adj}

Obtenido el R^2

```
WSST<- s0$dispersion
WSSE<- s1_qwgt$dispersion
(R2 = 1- WSSE/WSST)
```

```
     variance    SE
[1,]    0.495 20322
```

Obtenido el R^2_{adj}

```
n = nrow(encuesta)
p = 2
(R2Adj = 1-((1-R2)*(n-1)/(n-1-1)))
```

```
     variance    SE
[1,]    0.495 20322
```

Modelos empleando los Q_Weighting

Tabla 6: Comprando Modelos con Q Weighting

	svyglm(wgt)	svyglm(qwgt)
(Intercept)	103.136	109.123
	p = (0.114)	p = (0.115)
Expenditure	1.263	1.246
	p = (<0.001)	p = (<0.001)
Num.Obs.	2605	2605
R2	0.509	0.509
AIC	123.5	141.5
F	43.885	38.294
RMSE	345.09	344.96

Modelo propuesto

Después de realizar la comparación entre las diferentes formas de estimar los coeficientes del modelo se opta por la metodología consolidadas en `svyglm`

```
diseno_qwgt %<>% mutate(Age2 = Age^2)
mod_svy <- svyglm(
  Income ~ Expenditure + Zone + Sex + Age2 ,
                        design = diseno_qwgt)
s1_final <- summary(mod_svy)

stargazer(mod_svy, header = FALSE,single.row = T,
          title = "Modelo propuesto",
          style = "ajps",  omit.stat=c("bic", "ll"))
```

Resumen del modelo propuesto

Tabla 7: Modelo propuesto

	Income
Expenditure	1.208*** (0.209)
ZoneUrban	67.050 (41.340)
SexMale	21.330 (15.820)
Age2	0.008 (0.006)
Constant	66.980 (66.210)
N	2605
AIC	38332.000

***p $<$ .01; **p $<$ .05; *p $<$ .1

Diagnósticos de los residuales

En el diagnóstico de los modelos es crucial el análisis de los residuales. Estos análisis proporcionan, bajo el supuesto que el modelo ajustado es adecuado, una estimación de los errores.

Residuales Pearson

Los residuales de Pearson como sigue *(Heeringa)*

$$r_{pi} = \left(y_i - \mu_i\left(\hat{\beta}_\omega\right)\right)\sqrt{\frac{\omega_i}{V\left(\hat{\mu}_i\right)}}$$

Donde, μ_i es el valor esperado de y_i, ω_i es la ponderación de la encuesta para el i-ésimo individuo del diseño muestral complejo, Por último, $V(\mu_i)$ es la función de varianza del resultado.

Matriz **Hat**

Otra definición que se debe tener en consideración para el análisis de los residuales es el de la matriz hat, la cual se estima como:

$$H = W^{1/2} X \left(X' W X\right)^{-1} X' W^{1/2}$$

donde,

$$W = diag \left\{ \frac{\omega_1}{V\left(\mu_1\right)\left[g'\left(\mu_1\right)\right]^2}, ..., \frac{\omega_n}{V\left(\mu_n\right)\left[g'\left(\mu_n\right)\right]^2} \right\}$$

W es una matriz diagonal de $n \times n$ y $g()$ es la función de enlace del modelo lineal generalizado.

Distancia de cook

Diagnostica si la i-ésima observación es influyente en la estimación del modelo, por estar lejos del centro de masa de los datos. Se calcula de la siguiente manera:

$$c_i = \frac{w_i^* w_i e_i^2}{p\phi V\left(\hat{\mu}_i\right)\left(1-h_{ii}\right)^2} x_i^t \left[\widehat{Var}\left(U_w\left(\hat{\beta}_w\right)\right)\right]^{-1} x_i$$

Los elementos de la ecuación son:

- $w_i^* =$ Pesos de la encuesta.
- w_i Elementos por fuera de la diagonal de la matriz hat
- $e_i =$ residuales
- $p =$ número de parámetros del Modelo de regresión.
- $\phi =$ parámetro de dispersión en el glm
- $\widehat{Var}\left(U_w\left(\hat{\beta}_w\right)\right) =$ estimación de varianza linealizada de la ecuación de puntuación, que se utiliza para pseudo MLE en Modelos lineales generalizados ajustados a datos de encuestas de muestras complejas

Distancia de cook

Una vez que se ha determinado el valor de la D de Cook para un elemento de muestra individual, se puede calcular la siguiente estadística de prueba para evaluar la importancia de la estadística D:

$$\frac{(df - p + 1) \times c_i}{df} \doteq F_{(p, df-p)}$$

donde $df =$ grados de liberta basados en el diseño. Por otro lado, la literatura como *Tellez (2016)*, *Heeringa* considera a las observaciones influyentes cuando c_i sean mayores a 2 o 3.

$D_f Beta_{(i)}$

Este estadístico mide el cambio en la estimación del vector de coeficientes de regresión cuando la i-ésima observación es eliminada. Se evalúa con la siguiente expresión:

$$D_f Beta_{(i)} = \hat{\beta} - \hat{\beta}_{(i)} = \frac{A^{-1} X^t_{(i)} \hat{e}_i w_i}{1 - h_{ii}}$$

Donde $A = X^t W X$ $\hat{\beta}_{(i)}$ es el vector de parámetros estimados una vez se ha eliminado la i-ésima observación, h_{ii} es el correspondiente elemento de la diagonal de H y $\hat{e}_i$ es el residual de la i-ésima observación.

$D_f Beta_{(i)}$

Otra forma de reescribir este estadístico en términos de la matriz H es:

$$D_f Betas_{(i)} = \frac{c_{ji} e_i / (1 - h_{ii})}{\sqrt{v\left(\hat{\beta}_j\right)}}$$

donde:

- $c_{ji} =$ es el ji-estimo elemento de $A^{-1} w_i^2 X_{(i)} X_{(i)}^t A^{-1}$
- El estimador de $v\left(\hat{B}_j\right)$ basado en el Modelo se obtiene como: $v_m\left(\hat{B}_j\right) = \hat{\sigma} \sum_{i=1}^{n} c_{ji}^2$ con $\hat{\sigma} = \sum_{i \in s} w_i e^2 / \left(\hat{N} - p\right)$ y $\hat{N} = \sum_{i \in s} w_i$
- La i-ésima observación es influyente para B_j si $\mid D_f Betas_{(i)j} \mid \geq \frac{z}{\sqrt{n}}$ con $z = 2$ o 3
- Como alternativa puede usar $t_{0.025, n-p} / \sqrt{(n)}$ donde $t_{0.025, n-p}$ es el percentil 97.5

Estadístico $D_f Fits_{(i)}$

Este estadístico mide el cambio en el ajuste del modelo cuando se elimina el registro i-ésimo. Se calcula de la siguiente manera:

$$D_f Fits_{(i)} = \frac{h_{ii} e_i \big/ (1 - h_{ii})}{\sqrt{v\left(\hat{\beta}_j\right)}}$$

Donde, $\sqrt{v\left(\hat{\beta}_j\right)}$ puede ser aproximada por el diseño o el Modelo. La i-ésima observación se considera influyente en el ajuste del Modelo si $\mid Dffits\,(i) \mid \geq z\sqrt{\frac{p}{n}}$ con $z = 2$ o 3

Practica en R

```
par(mfrow = c(2,2))
plot(mod_svy)
```

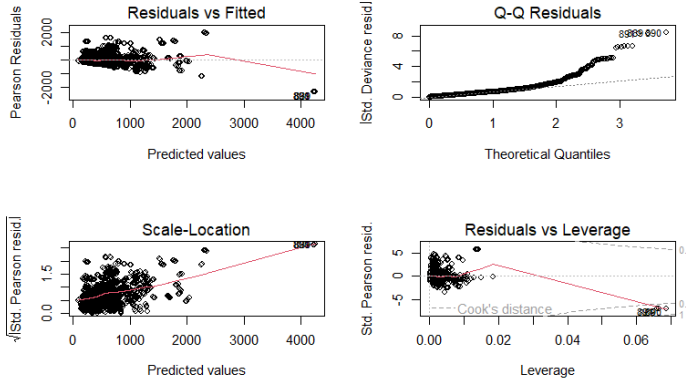


Figura 4: Analisis de residuales

Como se puedo observar en el QQplot, hay evidencia gráfica de que los errores no se distribuyen según una distribución normal.

Pruebas de normalidad

- H_0: Los errores proviene de una distribución normal.
- H_1: Los errores no proviene de una distribución normal.

Ahora, la librería `svydiags` esta pensada ayudar en el diagnostico de modelos de regresión lineal, siendo una extensión más para complementar el paquete `survey`.

Con las librería svydiags se extraen los residuales estandarizados así:

```
library(svydiags)
stdresids = as.numeric(svystdres(mod_svy)$stdresids)
diseno_qwgt$variables %<>%
  mutate(stdresids = stdresids)
```

Histograma de los residuales

El primer análisis de normalidad se hace por medio del histograma.

```
p1_hist <- ggplot(data = diseno_qwgt$variables,
                  aes(x = stdresids)) +
  geom_histogram(
    aes(y = ..density..),
    colour = "black",
    bins = 30,
    fill = "blue",
    alpha = 0.3
  ) + geom_density(linewidth = 2, colour = "blue") +
  geom_function(fun = dnorm, colour = "red",
                linewidth = 2) +
  theme_cepal() + labs(y = "")
```

Histograma de los residuales

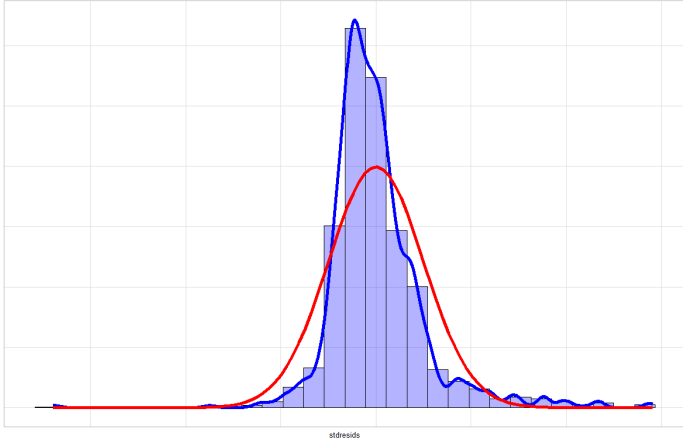


Figura 5: Histogrma de los residuales stdresids

Varianza constante

Agregando las predicciones a la base de datos.

```
library(patchwork)
diseno_qwgt$variables %<>%
  mutate(pred = predict(mod_svy))
g2 <- ggplot(data = diseno_qwgt$variables,
       aes(x = Expenditure, y = stdresids))+
  geom_point() +
  geom_hline(yintercept = 0) + theme_cepal()
g3 <- ggplot(data = diseno_qwgt$variables,
       aes(x = Age2, y = stdresids))+
  geom_point() +
  geom_hline(yintercept = 0) + theme_cepal()
```

Varianza constante

```
g4 <- ggplot(data = diseno_qwgt$variables,
       aes(x = Zone, y = stdresids))+
  geom_point() +
  geom_hline(yintercept = 0) + theme_cepal()
g5 <- ggplot(data = diseno_qwgt$variables,
       aes(x = Sex, y = stdresids))+
  geom_point() +  geom_hline(yintercept = 0) +
  theme_cepal()

(g2|g3)/(g4|g5)
```

Varianza constante

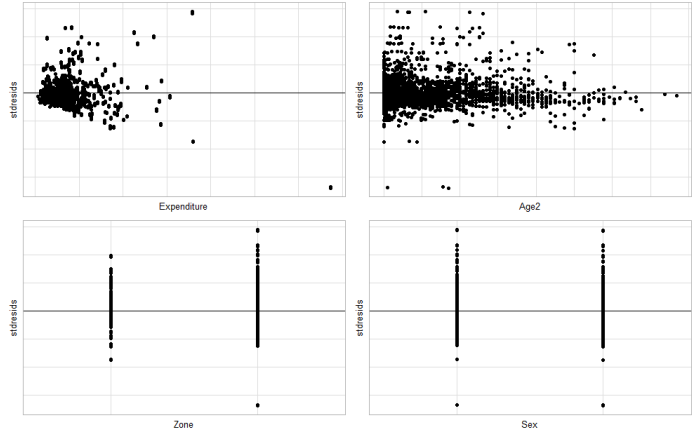


Figura 6: Varianza constante

Detección de observaciones influyentes (Distancia de cook)

La función `svyCooksD` pertenece a la librería `svydiags` permite tener el calculo de la Distancia de cook para el modelo ajustado.

```
 d_cook = data.frame(
   cook = svyCooksD(mod_svy),
     id = 1:length(svyCooksD(mod_svy)))

ggplot(d_cook, aes(y = cook, x = id)) +
  geom_point() +
  theme_bw(20)
```

Detección de observaciones influyentes (Distancia de cook)

Como se puede observar, ninguna de las distancias de Cook's es mayor a 3 por lo que, podemos decir que no existen observaciones influyentes.

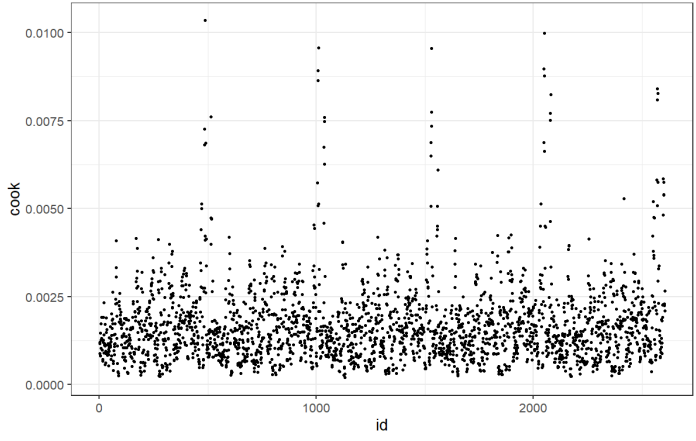


Figura 7: Distancia de cook

Detección de observaciones influyentes ($D_f Betas_{(i)j}$)

se desea observar si hay observaciones influyentes pero utilizando $D_f Betas_{(i)j}$ se realiza con la función `svydfbetas` como se muestra a continuación:

```
d_dfbetas = data.frame(t(svydfbetas(mod_svy)$Dfbetas))
colnames(d_dfbetas) <- paste0("Beta_", 1:5)
d_dfbetas %>% slice(1:5L)
```

Beta_1	Beta_2	Beta_3	Beta_4	Beta_5
5e-04	-2e-04	0.0020	-0.0045	-0.0075
-5e-04	-1e-04	0.0013	0.0026	-0.0030
-8e-04	-1e-04	0.0008	0.0022	0.0008
-4e-04	-1e-04	0.0011	-0.0031	0.0007
-8e-04	0e+00	0.0007	0.0021	0.0014

Detección de observaciones influyentes ($D_f Betas_{(i)j}$)

```
d_dfbetas$id <- 1:nrow(d_dfbetas)
d_dfbetas <- reshape2::melt(d_dfbetas,
                            id.vars = "id")
cutoff <- svydfbetas(mod_svy)$cutoff
d_dfbetas %<>%
  mutate(
    Criterio = ifelse(
      abs(value) > cutoff, "Si", "No"))

tex_label <- d_dfbetas %>%
  filter(Criterio == "Si") %>%
  arrange(desc(abs(value))) %>%
  slice(1:10L)
```

Detección de observaciones influyentes ($D_f Betas_{(i)j}$)

```
ggplot(d_dfbetas, aes(y = abs(value), x = id)) +
  geom_point(aes(col = Criterio)) +
  geom_text(data = tex_label,
            angle = 45,
            vjust = -1,
            aes(label = id)) +
  geom_hline(aes(yintercept = cutoff)) +
  facet_wrap(. ~ variable, nrow = 2) +
  scale_color_manual(
    values = c("Si" = "red", "No" = "black")) +
  theme_cepal()
```

Detección de observaciones influyentes ($D_f Betas_{(i)j}$)

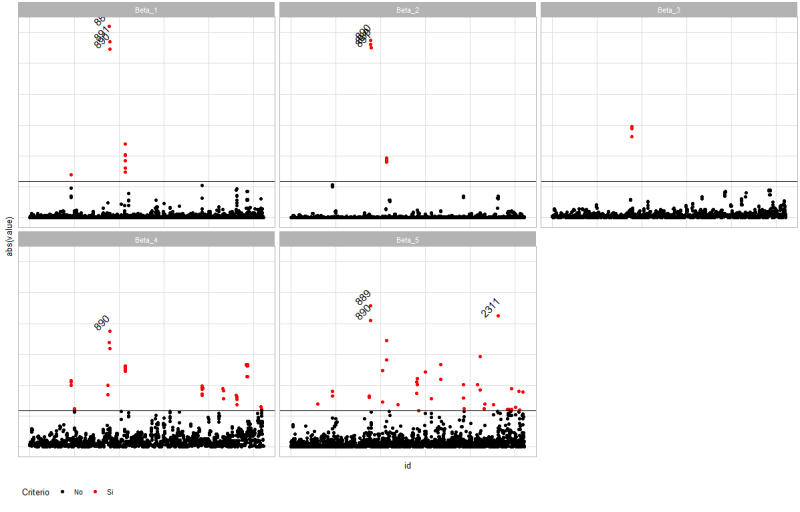


Figura 8: Detección de observaciones influyentes $D_f Betas_{(i)j}$

Detección de observaciones influyentes ($D_f Fits_{(i)}$)

```
d_dffits = data.frame(
  dffits = svydffits(mod_svy)$Dffits,
  id = 1:length(svydffits(mod_svy)$Dffits))

cutoff <- svydffits(mod_svy)$cutoff

d_dffits %<>% mutate(
    C_cutoff = ifelse(abs(dffits) > cutoff, "Si", "No"))
ggplot(d_dffits, aes(y = abs(dffits), x = id)) +
  geom_point(aes(col = C_cutoff)) +
  geom_hline(yintercept = cutoff) +
   scale_color_manual(
    values = c("Si" = "red", "No" = "black"))+
  theme_cepal()
```

Detección de observaciones influyentes ($D_fFits_{(i)}$)

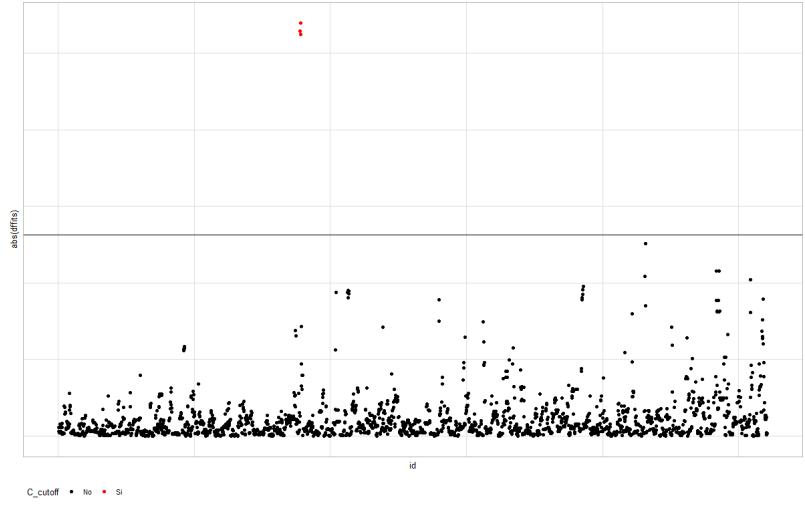


Figura 9: Detección de observaciones influyentes $D_fFits_{(i)}$

Matriz H

- La matriz asociada al Estimador de Pseudo Máxima Verosimilitud (PMLE) de $\hat{B}$ es $H = XA^{-1}X^{-t}W$ cuya diagonal esta dado por $h_{ii} = x_i^t A^{-1} {x_i}^{-t} w_i$.
- Una observación puede ser grande y, como resultado, influir en las predicciones, cuando un x_i es considerablemente diferente del promedio ponderado $\bar{x}_w = \sum_{i \in s} w_i x_i \big/ \sum_{i \in s} w_i$.

Detección de observaciones influyentes (h_{ii})

```
vec_hat <- svyhat(mod_svy, doplot = FALSE)
d_hat = data.frame(hat = vec_hat,
                   id = 1:length(vec_hat))
d_hat %<>% mutate(
  C_cutoff = ifelse(hat > (3 * mean(hat)),
                    "Si", "No"))

ggplot(d_hat, aes(y = hat, x = id)) +
  geom_point(aes(col = C_cutoff)) +
  geom_hline(yintercept = (3 * mean(d_hat$hat))) +
  scale_color_manual(
    values = c("Si" = "red", "No" = "black"))+
  theme_cepal()
```

Detección de observaciones influyentes (h_{ii})

Se puede observar en el gráfico anterior que hay varias observaciones posiblemente influyentes en el conjunto de datos de la muestra de hogares.

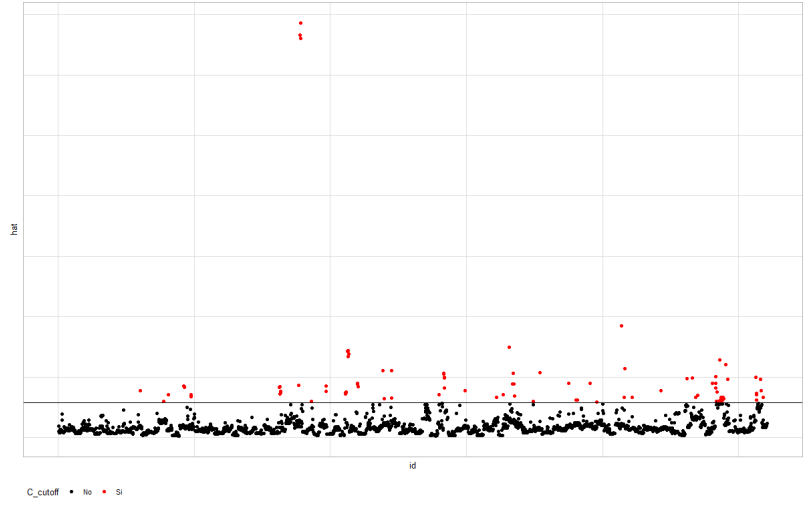


Figura 10: Detección de observaciones influyentes h_{ii}

Inferencia sobre los parámetros del Modelo

Inferencia sobre β_k

- Inferencia sobre los parámetros del modelo es fundamental para evaluar la significancia de los parámetros estimados.
- Utilizamos un estadístico de prueba basado en la distribución "t-student" para evaluar la significancia de los parámetros β_k.

$$t = \frac{\hat{\beta}_k - \beta_k}{se\left(\hat{\beta}_k\right)} \sim t_{n-p}$$

Donde p es el número de parámetros del modelo y n el tamaño de la muestra de la encuesta.

- El estadístico de prueba compara la hipótesis nula $H_0 : \beta_k = 0$ con la alternativa $H_1 : \beta_k \neq 0$.

Intervalo de confianza para β_k

- Podemos construir un intervalo de confianza al $(1-\alpha)\times 100\%$ para β_k usando el estadístico de prueba.

$$\hat{\beta}_k \pm t_{1-\frac{\alpha}{2},\, df}\, se\left(\hat{\beta}_k\right)$$

- Los grados de libertad (df) en una encuesta de hogares (muestras complejas) se calculan como el número de conglomerados finales de la primera etapa menos el número de estratos de la etapa primaria $(df = \sum_h a_h - H)$.
- Las funciones `summary.svyglm` para las pruebas t y `confint.svyglm` para los intervalos de confianza

Inferencia sobre β_k

El uso de `broom::tidy()` permite organizar la salida en la siguiente tabla

```
mod_svy %>% broom::tidy()
```

term	estimate	std.error	statistic	p.value
(Intercept)	66.9825	66.2141	1.012	0.3139
Expenditure	1.2082	0.2093	5.774	0.0000
ZoneUrban	67.0536	41.3446	1.622	0.1076
SexMale	21.3320	15.8214	1.348	0.1802
Age2	0.0085	0.0057	1.491	0.1386

Intervalo de confianza para β_k

Se puede observar que, con una confianza del 95% el único parámetro significativo del modelo es Expenditure y ese mismo resultado lo reflejan los intervalos de confianza.

```
survey:::confint.svyglm(mod_svy)
```

	2.5 %	97.5 %
(Intercept)	-64.1750	198.1399
Expenditure	0.7937	1.6227
ZoneUrban	-14.8421	148.9493
SexMale	-10.0071	52.6711
Age2	-0.0028	0.0197

Estimación de una observación

- Los modelos de regresión lineal se utilizan para dos propósitos principales: explicar la variable respuesta en función de las covariables y predecir valores de la variable de interés.
- Para realizar predicciones, utilizamos la fórmula

$$\hat{E}(y_i|x_{obs,i}) = x_{obs,i}\hat{\beta}$$

,

donde $x_{obs,i}$ representa las covariables de una observación no incluida en la muestra.

- La varianza de la estimación se calcula como

$$var(\hat{E}(y_i|x_{obs,i})) = x'_{obs,i}cov(\hat{\beta})x_{obs,i}$$

.

Estimación de una observación

(Intercept)	Expenditure	ZoneUrban	SexMale	Age2
1	346.3	0	1	4624
1	346.3	0	0	3136
1	346.3	0	0	576
1	346.3	0	1	676
1	346.3	0	0	9
1	392.2	0	0	3721
1	392.2	0	0	529

$$
\begin{aligned}
\hat{E}(y_i \mid x_{obs,i}) &= 67.0 + 1.21x_{1i} + 67.1x_{2i} + 21.3x_{3i} + 0.00846x_{4i} \\
&= 67.0 + 1.21(346.3) + 67.1(0) + 21.3(1) + 0.00846(68^2) \\
&= 546.4
\end{aligned}
$$

Estimando el IC de predicción

Para calcular la varianza de la estimación, primero se deben obtener las varianzas de la estimación de los parámetros:

```
vcov(mod_svy)
```

	(Intercept)	Expenditure	ZoneUrban	SexMale	Age2
(Intercept)	4384.3129	-12.6820	462.7656	306.4189	-0.2019
Expenditure	-12.6820	0.0438	-4.1970	-0.4944	0.0006
ZoneUrban	462.7656	-4.1970	1709.3771	-131.0058	-0.0629
SexMale	306.4189	-0.4944	-131.0058	250.3161	-0.0036
Age2	-0.2019	0.0006	-0.0629	-0.0036	0.0000

Estimando el IC de predicción

Ahora bien, se procede a realizar los cálculos como lo indica la expresión mostrada anteriormente:

```
xobs <- model.matrix(mod_svy) %>%
        data.frame() %>% slice(1) %>% as.matrix()

cov_beta <- vcov(mod_svy) %>% as.matrix()

as.numeric(xobs %*% cov_beta %*% t(xobs))
```

```
[1] 1977
```

Intervalo de confianza para la predicción

Si el objetivo ahora es calcular el intervalo de confianza para la predicción se utiliza la siguiente ecuación:

$$x_{obs,i}\widehat{\beta} \pm t_{(1-\frac{\alpha}{2},n-p)}\sqrt{var\left(\widehat{E}\left(y_i \mid x_{obs,i}\right)\right)}$$

Para realizar los cálculos en R, se utiliza la función `confint` y `predict` como sigue:

```
pred <- data.frame(predict(mod_svy, type = "response"))
pred_IC <- data.frame(confint(predict(mod_svy, type = "response")))
colnames(pred_IC) <- c("Lim_Inf", "Lim_Sup")
pred <- bind_cols(pred, pred_IC)
```

Intervalo de confianza para la predicción

```
pred %>% slice(1:10)
```

response	SE	Lim_Inf	Lim_Sup
545.9	44.46	458.8	633.0
512.0	33.67	446.0	578.0
490.3	29.26	433.0	547.7
512.5	37.08	439.8	585.2
485.5	29.19	428.3	542.7
572.4	42.08	489.9	654.8
545.4	34.68	477.4	613.3
562.4	40.60	482.9	642.0
540.9	34.24	473.8	608.0
207.3	52.80	103.8	310.8

Intervalo de confianza para la predicción

Ahora, de manera gráfica las predicciones e intervalos se vería de la siguiente manera:

```
pred$Expenditure <- encuesta$Expenditure
pd <- position_dodge(width = 0.2)
ggplot(pred %>% slice(1:100L),
       aes(x = Expenditure , y = response)) +
  geom_errorbar(aes(ymin = Lim_Inf,
                    ymax = Lim_Sup),
                width = .1,
                linetype = 1) +
  geom_point(size = 2, position = pd) +
  theme_bw()
```

Intervalo de confianza para la predicción

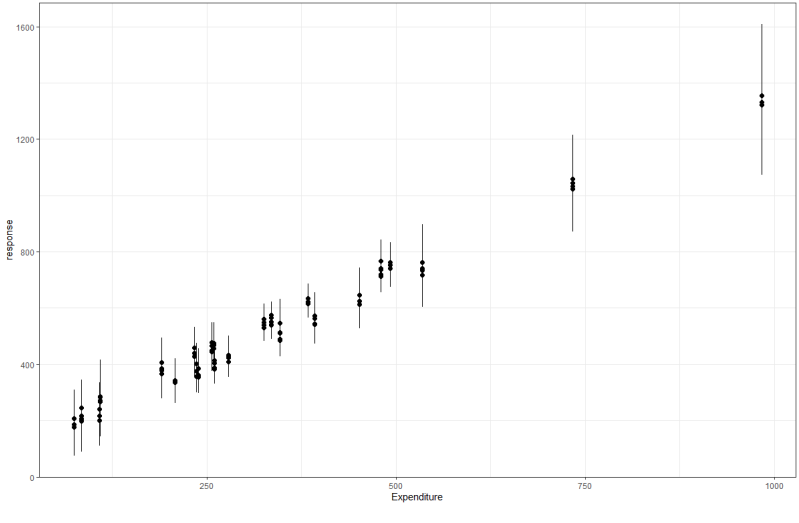


Figura 11: Intervalo de confianza para la predicción

Predicción fuera fuera del rango de valores.

Por último, si el interés es hacer una predicción fuera del rango de valores que fue capturado en la muestra. Para esto, supongamos que se desea predecir:

```
datos_nuevos <- data.frame(Expenditure = 1600,
                           Age2 = 40^2, Sex = "Male",
                           Zone = "Urban")
```

La varianza para la predicción se hace siguiendo la siguiente ecuación:

$$var\left(\hat{E}\left(y_i \mid x_{obs,i}\right)\right) = x_{obs,i}^t cov\left(\beta\right) x_{obs,i} + \hat{\sigma}_{yx}^2$$

Predicción fuera fuera del rango de valores.

Se construye la matriz de observaciones y se calcula la varianza como sigue:

```
x_noObs = matrix(c(1,1600,1,1,40^2),nrow = 1)
as.numeric(sqrt(x_noObs%*%cov_beta%*%t(x_noObs)))
```

```
[1] 257.4
```

Por último, el intervalo de confianza sigue la siguiente ecuación:

$$x_{obs,i}\hat{\beta} \pm t_{(1-\frac{\alpha}{2},n-p)}\sqrt{var\left(\hat{E}\left(y_i \mid x_{obs,i}\right)\right) + \hat{\sigma}^2_{yx}}$$

Predicción fuera fuera del rango de valores.

En R se hace la predicción de la siguiente manera:

```
predict(mod_svy, newdata = datos_nuevos, type =  "link") %>%
  data.frame()
```

link	SE
2102	257.4

y el intervalo:

```
confint(predict(mod_svy, newdata = datos_nuevos))
```

2.5 %	97.5 %
1598	2607

Procesando múltiples bases.

Lectura de múltiples bases

Para realizar la lectura de múltiples bases debemos conocer las rutas donde estas estas guardadas para ello empleamos la función `file.list` del paquete base, que nos permite tener un listado completo de los archivos.

```
library(stringr)
(data_path <- list.files("V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/",
                         full.names = TRUE, pattern = "2020") %>%
  tibble(path = .) %>%
   mutate(
     Pais = basename(path) %>% str_extract("^[A-Z]+")))
```

Note que utiliza la función gsub para separar el nombre del país de la ruta.

Lectura de múltiples bases

path	Pais
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/ARG_2020N.dta	ARG
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/BOL_2020N.dta	BOL
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/BRA_2020N1.dta	BRA
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/CHL_2020N.dta	CHL
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/COL_2020N1.dta	COL
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/CRI_2020N1.dta	CRI
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/DOM_2020N1.dta	DOM
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/ECU_2020N.dta	ECU
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/MEX_2020N1.dta	MEX
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/PER_2020N.dta	PER
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/PRY_2020N.dta	PRY
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/SLV_2020N.dta	SLV
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/URY_2020N.dta	URY

Lectura de encuestas

Para la lectura de los archivos, se procede de la siguiente forma.

```
require(purrr)
require(haven)
data_path %<>%
  mutate(encuesta = path %>% map(~read_dta(.x) %>%
           transmute(upm = `_upm`,
           estrato =`_estrato`,
           sexo, area_ee,lp,li,ingcorte,
           fep =`_fep`)))
```

La función `map` es utilizada para trabajar con los elementos de una lista. Las variables seleccionadas son sexo, área geográfica (area_ee), Linea de pobreza (lp), Linea de indigencia (li), Íngreso persona (ingcorte) y factor de expansión por persona (fep).

Lectura de encuestas (resultado)

```
> data_path
# A tibble: 13 × 3
   path                                                  Pais  encuesta
   <chr>                                                 <chr> <list>
 1 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/ARG_2020N.dta ARG   <tibble [85,452 × 8]>
 2 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/BOL_2020N.dta BOL   <tibble [37,092 × 8]>
 3 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/BRA_2020N1.dta BRA   <tibble [355,436 × 8]>
 4 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/CHL_2020N.dta CHL   <tibble [185,437 × 8]>
 5 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/COL_2020N1.dta COL   <tibble [747,822 × 8]>
 6 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/CRI_2020N1.dta CRI   <tibble [25,530 × 8]>
 7 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/DOM_2020N1.dta DOM   <tibble [71,378 × 8]>
 8 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/ECU_2020N.dta ECU   <tibble [30,646 × 8]>
 9 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/MEX_2020N1.dta MEX   <tibble [315,743 × 8]>
10 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/PER_2020N.dta PER   <tibble [120,346 × 8]>
11 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/PRY_2020N.dta PRY   <tibble [17,582 × 8]>
12 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/SLV_2020N.dta SLV   <tibble [37,030 × 8]>
13 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/URY_2020N.dta URY   <tibble [145,166 × 8]>
```

Figura 12: Lectura de encuestas

El resultado es objeto tipo `tibble` el cual permite observar de forma compacta el contenido de una lista e indica el tipo y tamaño de cada objeto en la contenido en la lista.

Definir el diseño

Ahora se debe definir un diseño para cada encuesta, para nuestro ejemplo se define el diseño muestral.

```
options(survey.lonely.psu="adjust")
data_path <- readRDS("Imagenes/06_regresion/15_data_path.rds")
data_path %<>% mutate(
  diseno = encuesta %>%
           map(~as_survey_design(.data = .x,
               ids = upm,
               strata = estrato,
               weights = fep,
               nest = T
               )))
```

Definir el diseño (resultado)

```
> data_path
# A tibble: 13 × 4
   path                                                  Pais  encuesta               diseno
   <chr>                                                 <chr> <list>                 <list>
 1 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/ARG_2020N.dta ARG   <tibble [85,452 × 8]>  <tbl_svy[,8]>
 2 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/BOL_2020N.dta BOL   <tibble [37,092 × 8]>  <tbl_svy[,8]>
 3 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/BRA_2020N1.dta BRA  <tibble [355,436 × 8]> <tbl_svy[,8]>
 4 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/CHL_2020N.dta CHL   <tibble [185,437 × 8]> <tbl_svy[,8]>
 5 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/COL_2020N1.dta COL  <tibble [747,822 × 8]> <tbl_svy[,8]>
 6 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/CRI_2020N1.dta CRI  <tibble [25,530 × 8]>  <tbl_svy[,8]>
 7 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/DOM_2020N1.dta DOM  <tibble [71,378 × 8]>  <tbl_svy[,8]>
 8 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/ECU_2020N.dta ECU   <tibble [30,646 × 8]>  <tbl_svy[,8]>
 9 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/MEX_2020N1.dta MEX  <tibble [315,743 × 8]> <tbl_svy[,8]>
10 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/PER_2020N.dta PER   <tibble [120,346 × 8]> <tbl_svy[,8]>
11 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/PRY_2020N.dta PRY   <tibble [17,582 × 8]>  <tbl_svy[,8]>
12 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/SLV_2020N.dta SLV   <tibble [37,030 × 8]>  <tbl_svy[,8]>
13 V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/URY_2020N.dta URY   <tibble [145,166 × 8]> <tbl_svy[,8]>
```

Figura 13: Diseño muestral para multiples bases

Estimación de los coeficiente del modelo en multiples encuestas

```
library(tidyr)
data_path %>% mutate(
  model = map(diseno,
              ~svyglm(ingcorte~sexo+area_ee,.x) %>%
               coef() %>%
               data.frame(estimado = .) %>%
               tibble::rownames_to_column(var = "Coef"))) %>%
  dplyr::select(pais,model) %>% unnest(model)
```

Estimación de los coeficiente del modelo en multiples encuestas (Resultado)

pais	Coef	estimado
CRI	(Intercept)	401758.26
CRI	sexo	-435.35
CRI	area_ee	-111216.11
DOM	(Intercept)	15857.78
DOM	sexo	-687.20
DOM	area_ee	-2683.59
ECU	(Intercept)	369.66
ECU	sexo	-10.08
ECU	area_ee	-107.37
PER	(Intercept)	1202.07
PER	sexo	-22.46
PER	area_ee	-398.57

Alternativa para el procesamiento de múltiples archivos.

En ocasiones solo se desea obtener un resultado rápido para realizar un reporte o una comparación rápida de información, en estas ocasiones no es necesario guardar en la memoria de R toda la encuesta, por esta razón se ilustra una alternativa de procesamiento de múltiples archivos.

- **Paso 1** Leer archivo y organizar encuestas.
- **Paso 2** Definir diseño muestral.
- **Paso 3** Procesar información.
- **Paso 4** Organizar y presentar resultados.

Creando función para el procesamiento de múltiples archivos.

```
options(survey.lonely.psu="adjust")
library(tidyr)
model_aux <- function(input_file){
  ## Paso 1
  encuesta <- read_dta(input_file) %>%
    transmute(upm = `_upm`,   estrato =`_estrato`,
              sexo, area_ee,lp,li,ingcorte, fep =`_fep`)
  ## Paso 2
  diseno <- as_survey_design(.data = encuesta,
                             ids = upm,
                             strata = estrato,
                             weights = fep,
                             nest = T)
  ## Paso 3
  s <- svyglm(ingcorte~sexo+area_ee,diseno)
  tidy(s)
}
```

Procesando encuestas múltiples

Para el *Paso 4* realizamos la siguiente sintaxis.

```
setNames(data_path$path, data_path$pais) %>%
  map_df(~model_aux(.x), .id = "País" )  %>%
  saveRDS("Imagenes/06_Regresion/16_modelo_multiples.rds")
```

Procesando encuestas múltiples

Los resultados se muestran en el orden de lectura de los archivos

País	term	estimate	std.error	statistic	p.value
CRI	(Intercept)	401758.26	12721.261	31.5816	0.0000
CRI	sexo	-435.35	3181.853	-0.1368	0.8912
CRI	area_ee	-111216.11	7214.760	-15.4151	0.0000
DOM	(Intercept)	15857.78	576.642	27.5002	0.0000
DOM	sexo	-687.20	136.548	-5.0327	0.0000
DOM	area_ee	-2683.59	323.351	-8.2993	0.0000
ECU	(Intercept)	369.66	25.387	14.5610	0.0000
ECU	sexo	-10.08	3.462	-2.9112	0.0037
ECU	area_ee	-107.37	13.499	-7.9541	0.0000
PER	(Intercept)	1202.07	24.660	48.7463	0.0000
PER	sexo	-22.46	5.500	-4.0827	0.0000
PER	area_ee	-398.57	12.911	-30.8717	0.0000

¡Gracias!

Email: andres.gutierrez@cepal.org