Análisis de encuestas de hogares con R

Módulo 4: 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 definiendo algunas variables de interés.

- CANTIDAD_PERSONAS: Cantidad de miembros pertenecientes al hogar.
- YDISPONIBLE_PER: Corresponde al ingreso disponible del hogar, dividido por la cantidad de personas en el hogar.
- GASTO_CORRIENTE_HOGAR: Gasto corriente del Hogar
- CONSUMO_FINAL_HOGAR: Gasto de consumo final del Hogar

Aplicación en encuestas de hogares

CANTIDAD_PERSONAS	GASTO_CORRIENTE_HOGAR	CONSUMO_FINAL_HOGAR	YDISPONIBLE_PER	ingreso_per	ingreso_hog
4	18193	18011	-2811.51	0.00	0.00
1	28131	18184	-1772.00	0.00	0.00
1	21560	15961	-1702.58	0.00	0.00
2	13479	12968	-255.54	0.00	0.00
1	46836	46579	-189.08	0.00	0.00
2	7079	6789	-53.54	0.00	0.00
1	2652	2652	0.00	0.00	0.00
4	5384	5384	18.75	18.75	75.00
3	5494	5494	20.83	20.83	62.50
2	8967	8967	57.71	57.71	115.42
1	12585	11418	68.06	68.06	68.06
4	2828	2828	111.94	111.94	447.75
3	32166	15307	160.00	160.00	479.99
3	5928	5928	166.67	166.67	500.00
3	4625	4625	180.56	180.56	541.67
2	8213	8213	187.50	187.50	375.00
3	4063	4063	196.06	196.06	588.17
6	14287	14287	226.67	226.67	1360.00
4	6406	6406	232.08	232.08	928.33
3	8294	8294	247.00	247.00	741.00

Definición del objeto `survey.design`

```
diseno <-  encuesta %>% as_survey_design(
    strata = estrato,  # Id de los estratos.
    ids = F1_A0_UPM,          # Id para las observaciones.
    weights = Factor,       # Factores de expansión.
    nest = TRUE            # Valida el anidado dentro del estrato
  )
```

Sub-grupos

Dividir la muestra en sub-grupos de la encuesta.

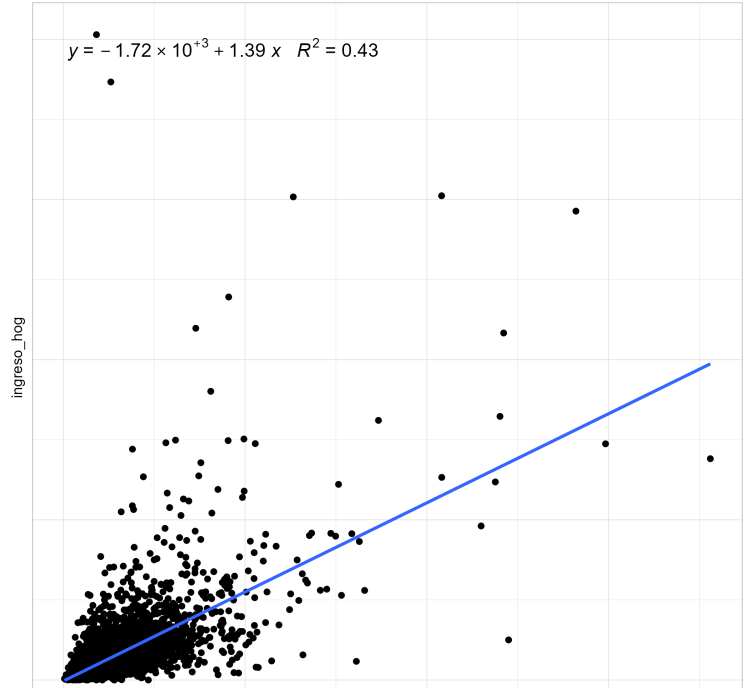
```
sub_Urbano <- diseno %>%  filter(Area == "1. Urbana") #
sub_Rural  <- diseno %>%  filter(Area == "2. Rural") #
```

Scatterplot con los datos encuesta sin ponderar

Una sintaxis similar permite construir el scatterplot en la muestra.

```
plot_sin <-
  ggplot(data = encuesta,
         aes(x = GASTO_CORRIENTE_HOGAR, y = ingreso_hog)) +
  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



Modelo sin ponderar

El modelo ignorando los factores de expansión quedas así:

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

Modelo sin ponderar

Tabla 2: Modelo encuesta Sin ponderar

	ingreso_hog
GASTO_CORRIENTE_HOGAR	1.391***
	(0.017)
Constant	−1723.000***
	(479.200)
N	8746
R-squared	0.434
Adj. R-squared	0.434
Residual Std. Error	28819.000 (df = 8744)
F Statistic	6713.000*** (df = 1; 8744)

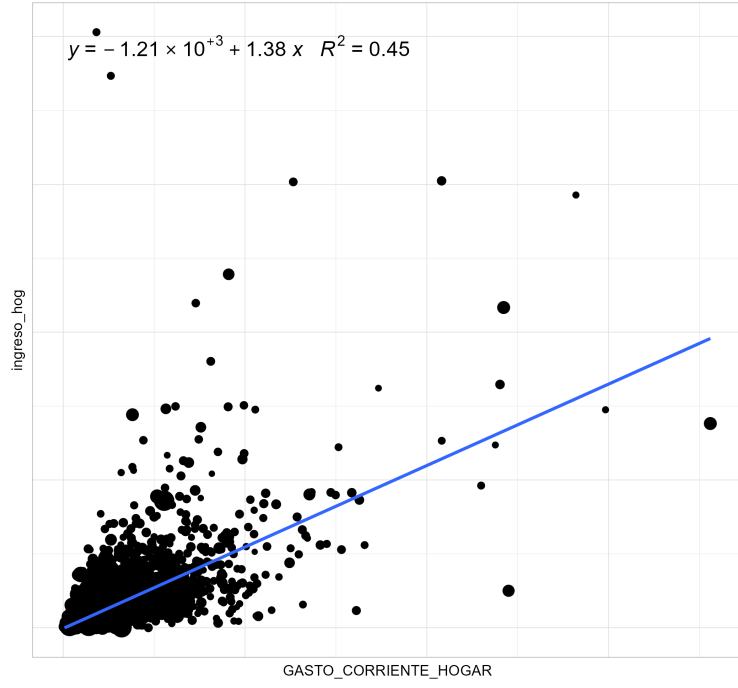
***p < .01; **p < .05; *p < .1

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 = GASTO_CORRIENTE_HOGAR, y = ingreso_hog)) +
  geom_point(aes(size = Factor)) +
  geom_smooth(method = "lm",
              se = FALSE,
              formula = y ~ x,
              mapping = aes(weight = Factor)) +
  theme_cepal()
plot_Ponde <- plot_Ponde + stat_poly_eq(formula = y~x,
  aes(weight = Factor,
    label = paste(..eq.label..,
      ..rr.label.., sep = "~~~")),
  parse = TRUE,size = 5)
```

Scatterplot con los datos encuesta sin ponderar



Modelo ponderado `lm`

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

```
fit_Ponde <- lm(ingreso_hog ~ GASTO_CORRIENTE_HOGAR,
                data = encuesta, weights = Factor)
stargazer(fit_Ponde, header = FALSE,
          title = "Modelo encuesta ponderada",
          style = "ajps")
```

Modelo ponderado lm

Tabla 3: Modelo encuesta ponderada

	ingreso_hog
GASTO_CORRIENTE_HOGAR	1.378***
	(0.016)
Constant	−1209.000***
	(415.200)
N	8746
R-squared	0.452
Adj. R-squared	0.452
Residual Std. Error	448570.000 (df = 8744)
F Statistic	7213.000*** (df = 1; 8744)

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

Modelo ponderado svyglm

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

```
fit_svy <- svyglm(ingreso_hog ~ GASTO_CORRIENTE_HOGAR,
                  design = diseno)
```

Resumen del Modelo

Tabla 4: Modelo encuesta ponderada, svyglm

	ingreso_hog
GASTO_CORRIENTE_HOGAR	1.378***
	(0.081)
Constant	−1209.000
	(1361.000)
N	8746
AIC	205813.000

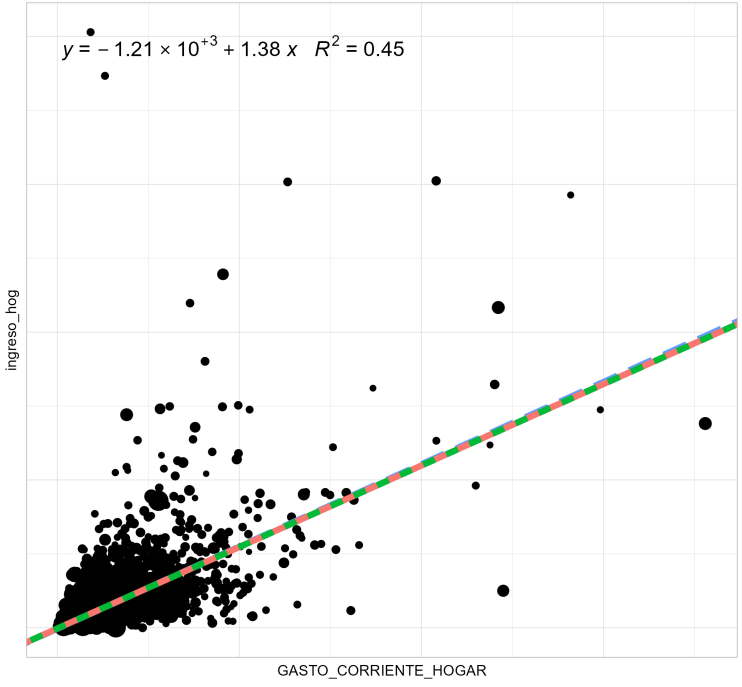
***p < .01; **p < .05; *p < .1

Comparando los resultados

```
df_model <- data.frame(
  intercept = c(coefficients(fit_sinP)[1],
                coefficients(fit_Ponde)[1],
                coefficients(fit_svy)[1]),
  slope = c(  coefficients(fit_sinP)[2],
                coefficients(fit_Ponde)[2],
                coefficients(fit_svy)[2]),
  Modelo = c("Sin ponderar",
             "Ponderado(lm)", "Ponderado(svyglm)"))

plot_Ponde2  <- plot_Ponde +  geom_abline( data = df_model,
    mapping = aes( slope = slope,
      intercept = intercept, linetype = Modelo,
      color = Modelo ), size = 2
  )
```

Comparando los resultados



Comparando los resultados

Variable	Sin Pond	Ponde(lm)	Ponde(svyglm)
(Intercept)	-1722.608	-1209.015	-1209.015
p-value	(<0.001)	(0.004)	(0.375)
GASTO_CORRIENTE_HOGAR	1.391	1.378	1.378
p-value	(<0.001)	(<0.001)	(<0.001)
Num.Obs.	8746	8746	8746
R2	0.434	0.452	0.452
R2 Adj.	0.434	0.452	-5.619
AIC	204445.8	205813.2	202638.9
F	6712.633	7213.379	288.839
RMSE	28815.76	28817.66	28817.66

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(ingreso_hog ~ GASTO_CORRIENTE_HOGAR ,
                  design =  diseno,family=stats::gaussian())

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

diseno %<>% mutate(
  ypred = fitted(fit_svy, type = "response"),
  medY = medY,
  sst = (ingreso_hog - 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
3.211e+15	1.451e+15	0.452

De forma alternativa es:

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

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

```
     variance       SE
[1,]    0.452 1.07e+08
```

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,]    0.548 1.07e+08
```

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(1/Factor ~ GASTO_CORRIENTE_HOGAR, 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)
summary(qw)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0058	0.0061	0.0062	0.0064	0.0065	0.0151

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 = Factor/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 = estrato,  # Id de los estratos.
    ids = F1_A0_UPM,          # Id para las observaciones.
    weights = wk1,       # Factores de expansión.
    nest = TRUE            # Valida el anidado dentro del estrato
  )
```

Modelos empleando los Q_Weighting

Estimando los coeficientes del modelo con los Q_Weighting de pfefferman

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

term	estimate	std.error	statistic	p.value
(Intercept)	-1306.548	1109.1754	-1.178	0.2392
GASTO_CORRIENTE_HOGAR	1.383	0.0696	19.872	0.0000

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,]    0.531 92109176
```

Obtenido el R^2_{adj}

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

```
     variance       SE
[1,]    0.531 92109176
```

Modelos empleando los Q_Weighting

Tabla 9: Comprando Modelos con Q Weighting

Variable	svyglm(wgt)	svyglm(qwgt)
(Intercept)	-1209.015	-1306.548
p-value	(0.375)	(0.239)
GASTO_CORRIENTE_HOGAR	1.378	1.383
p-value	(<0.001)	(<0.001)
Num.Obs.	8746	8746
R2	0.452	0.423
AIC	202638.9	201262.1
F	288.839	394.884
RMSE	28817.66	28817.13

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 <-
  diseno_qwgt %>% mutate(
    TIPOVIVIENDA = as_factor(F1_A1_P1_TIPOVIVIENDA),
    TIENEVEHICULOS = as_factor(F2_A2_P1_TIENEVEHICULOS))

mod_svy <- svyglm(
  log(ingreso_hog + 500) ~ log(GASTO_CORRIENTE_HOGAR +500) +
  TIENEVEHICULOS + Area  ,
  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 10: Modelo propuesto

	log(ingreso_hog + 500)
log(GASTO_CORRIENTE_HOGAR + 500)	0.853^{***} (0.021)
TIENEVEHICULOS2. No	-0.242^{***} (0.026)
Area2. Rural	-0.095^{***} (0.035)
Constant	1.684^{***} (0.208)
N	8746
AIC	18627.000

$^{***}p < .01$; $^{**}p < .05$; $^{*}p < .1$

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,

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

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

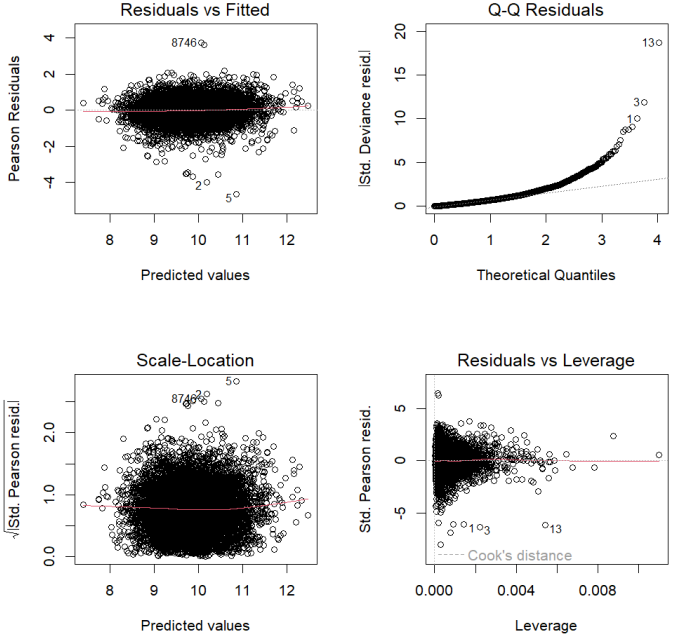
Estadístico $D_f Fits_{(i)}$

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



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

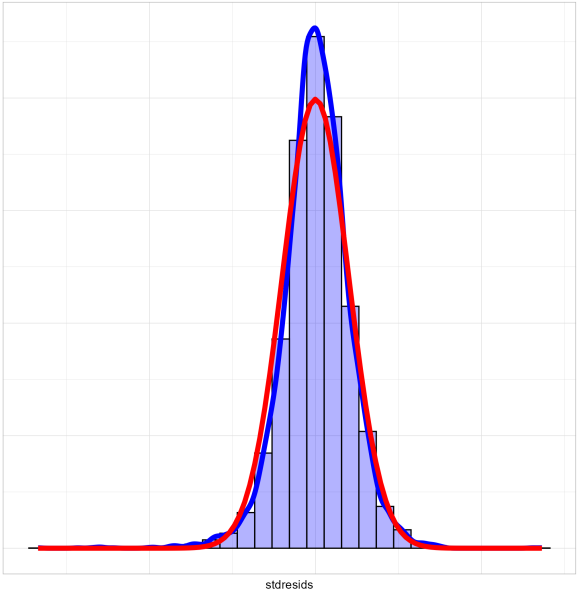


Figura 4: Histogrma de los residuales stdresids

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 = log(GASTO_CORRIENTE_HOGAR +
    500), y = stdresids))+
  geom_point() +
  geom_hline(yintercept = 0) + theme_cepal()
g3 <- ggplot(data = diseno_qwgt$variables,
       aes(x = TIENEVEHICULOS , y = stdresids))+
  geom_point() +
  geom_hline(yintercept = 0) + theme_cepal()
```

Varianza constante

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

g6 <- (g4|g3)/(g2)
```

Varianza constante

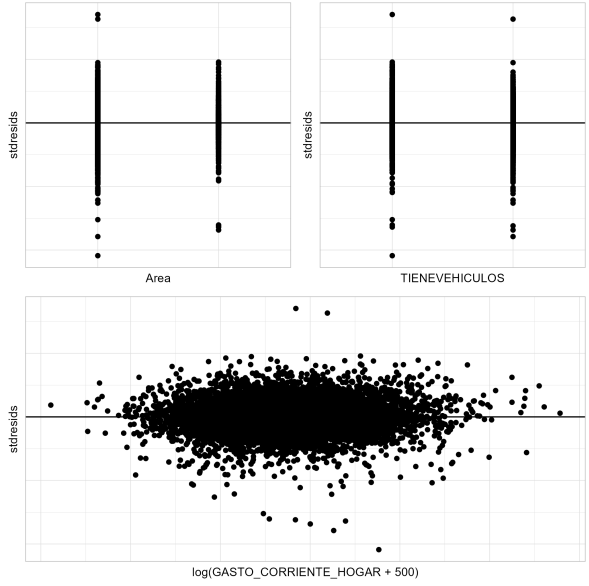


Figura 5: Varianza constante

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)))

g_dcook<- 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.

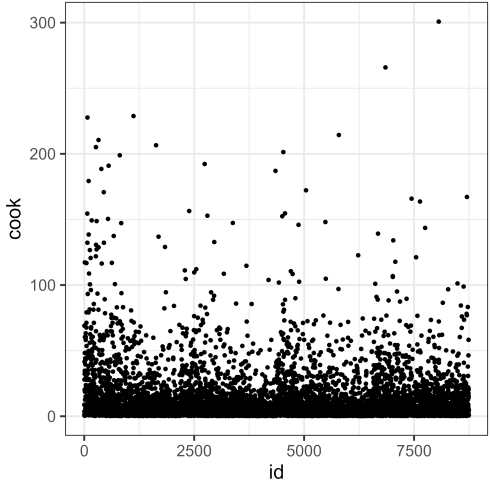


Figura 6: Distancia de cook

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_", 1:4)
d_dfbetas %>% slice(1:5L)
```

Beta_1	Beta_2	Beta_3	Beta_4
0.0784	-0.0729	-0.1043	-0.2168
0.0769	-0.0761	-0.0893	0.0080
0.1539	-0.1458	-0.1623	-0.3058
-0.0707	0.0642	0.1114	-0.1556
0.0363	-0.0391	0.0080	0.0006

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

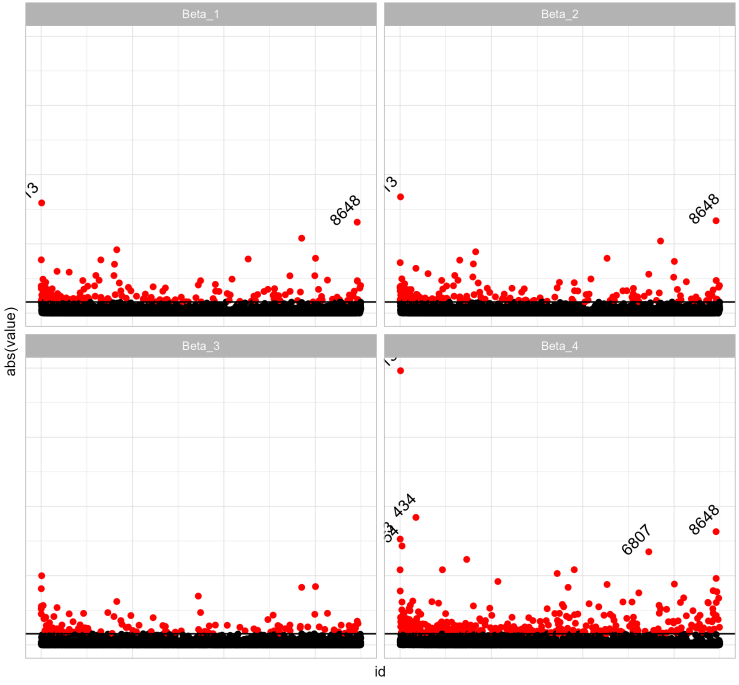
```
d_dfbetas$id <- 1:nrow(d_dfbetas)
d_dfbetas <- reshape2::melt(d_dfbetas,
                            id.vars = "id")
cutoff <- svydfbetas(mod_svy)$cutoff
d_dfbetas %<>%
  mutate(
    Criterio = ifelse(
      abs(value) > cutoff, "Si", "No"))

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

Detección de observaciones influyentes ($D_fBetas_{(i)j}$)

```
Fig_DFbetas <- 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 ($D_fBetas_{(i)i}$)



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

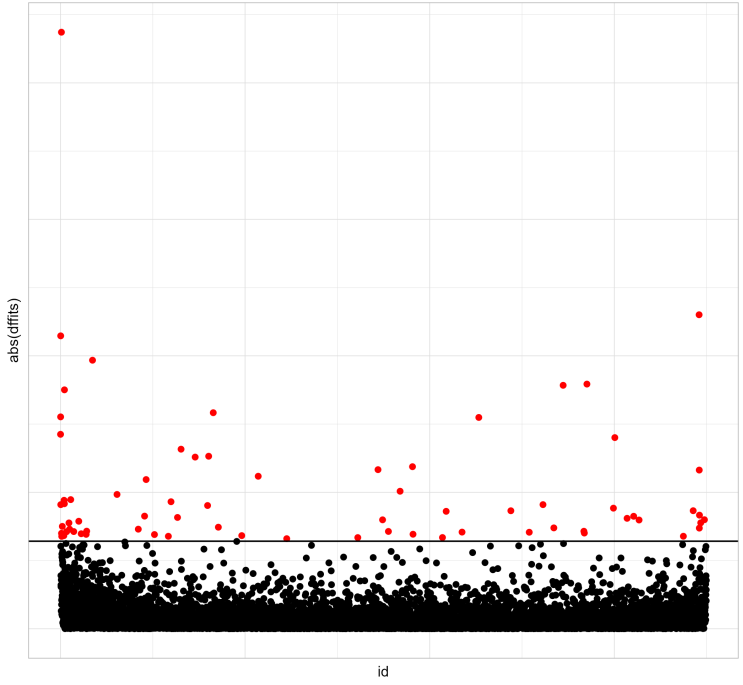
```
d_dffits = data.frame(
  dffits = svydffits(mod_svy)$Dffits,
  id = 1:length(svydffits(mod_svy)$Dffits))

cutoff <- svydffits(mod_svy)$cutoff

d_dffits %<>% mutate(
    C_cutoff = ifelse(abs(dffits) > cutoff, "Si", "No"))

g_dffits <- 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 ($D_f Fits_{(i)}$)



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 $\bar{x}_w = \sum_{i \in s} w_i x_i \big/ \sum_{i \in s} w_i$.

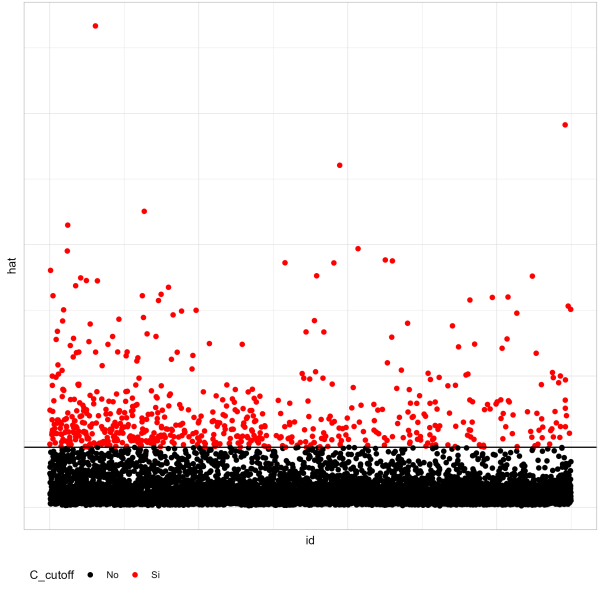
Detección de observaciones influyentes (h_{ii})

```
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"))

g_hat <- 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.



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 $H_1 : \beta_k \neq 0$.

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)	1.6840	0.2084	8.081	0.0000
log(GASTO_CORRIENTE_HOGAR + 500)	0.8527	0.0210	40.689	0.0000
TIENEVEHICULOS2. No	-0.2417	0.0259	-9.348	0.0000
Area2. Rural	-0.0951	0.0350	-2.714	0.0068

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)	1.2749	2.0932
log(GASTO_CORRIENTE_HOGAR + 500)	0.8115	0.8938
TIENEVEHICULOS2. No	-0.2925	-0.1909
Area2. Rural	-0.1639	-0.0263

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)	log(GASTO_CORRIENTE_HOGAR + 500)	TIENEVEHICULOS2. No	Area2. Rural
1	9.836	1	1
1	10.262	1	0
1	10.002	1	1
1	9.545	0	1
1	10.765	0	0
1	8.933	1	0
1	8.056	1	0

$$
\begin{aligned}
\hat{E}(y_i \mid x_{obs,i}) &= 1.6840 + 0.8527x_{1i} - 0.2417x_{2i} - 0.0951x_{3i} \\
&= 1.6840 + 0.8527(9.836) + -0.2417(1) - 0.0951(1) \\
&= 9.734
\end{aligned}
$$

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)	log(GASTO_CORRIENTE_HOGAR + 500)	TIENEVEHICULOS2. No	Area2. Rural
(Intercept)	0.0434	-0.0044	-0.0014	-0.0039
log(GASTO_CORRIENTE_HOGAR + 500)	-0.0044	0.0004	0.0001	0.0004
TIENEVEHICULOS2. No	-0.0014	0.0001	0.0007	-0.0001
Area2. Rural	-0.0039	0.0004	-0.0001	0.0012

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] 0.001271
```

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)
```

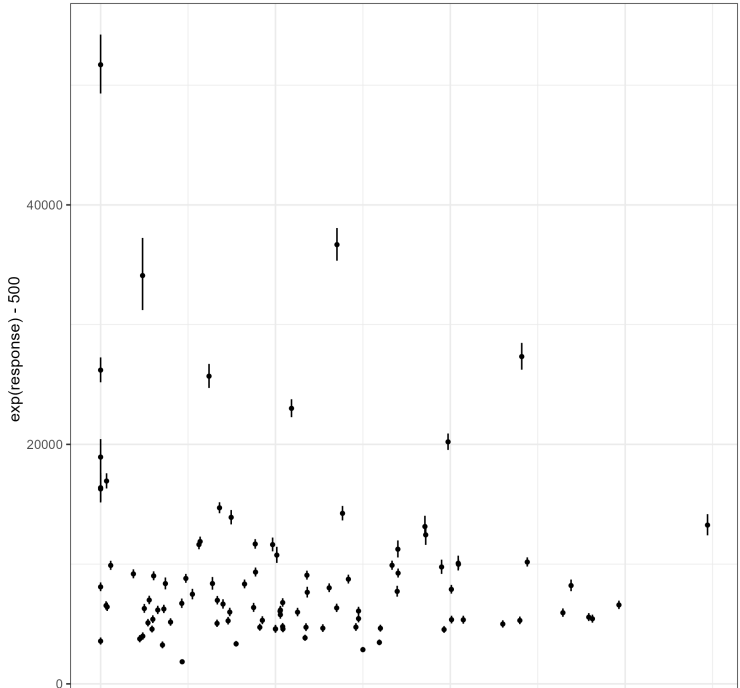
response	SE	Lim_Inf	Lim_Sup
9.734	0.0357	9.664	9.804
10.193	0.0198	10.154	10.231
9.875	0.0379	9.801	9.950
9.728	0.0355	9.658	9.797
10.863	0.0240	10.816	10.910
9.059	0.0217	9.017	9.102
8.311	0.0370	8.239	8.384
8.843	0.0257	8.793	8.894
8.859	0.0254	8.809	8.909
9.249	0.0187	9.212	9.286

Intervalo de confianza para la predicción

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

```
pred$log_ingreso_hog <- log(encuesta$ingreso_hog + 500)
pd <- position_dodge(width = 0.2)
pred_mod <-  ggplot(pred %>% slice(1:100L),
       aes(x = exp(log_ingreso_hog) - 500 , y = exp(response) - 500 )) +
  geom_errorbar(aes(ymin = exp(Lim_Inf) - 500,
                    ymax = exp(Lim_Sup) -500 ),
                width = .1,
                linetype = 1) +
  geom_point(size = 1, position = pd) +
  theme_bw()
```

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(GASTO_CORRIENTE_HOGAR  = 500,
                           TIENEVEHICULOS = "2. No",
                           Area = "2. Rural" )
```

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,log(500+500) ,1,1),nrow = 1)
as.numeric(sqrt(x_noObs%*%cov_beta%*%t(x_noObs)))
```

```
[1] 0.0478
```

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
7.237	0.0478

y el intervalo:

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

2.5 %	97.5 %
7.144	7.331

¡Gracias!

Email: andres.gutierrez@cepal.org