

Modelos de regresión

CEPAL

17/2/2022

Lectura de la base

```
encuesta <- readRDS("../Data/encuesta.rds")
data("BigCity", package = "TeachingSampling")
```

Definir diseño de la muestra con srvyr

```
library(srvyr)

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

Sub-grupos

Extraer 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")
```

Modelo de regresión

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

$$E(y | x) = B_0 + B_1 x$$

donde \$ B = [B_0, B1]\$ y el estimador de \$ B\$ esta dado por:

$$\hat{B} = (\mathbf{x}^T \mathbf{W} \mathbf{x})^{-1} \mathbf{x}^T \mathbf{W} \mathbf{y}$$

$$F(B) = \sum_{i=1}^N (y_i - \mathbf{x}_i \mathbf{B})^2$$

$$\widehat{WSSE}_{pop} = \sum_h^H \sum_{\alpha}^{a_h} \sum_{i=1}^{n_{h\alpha}} w_{h\alpha i} (y_{hai} - \mathbf{x}_{h\alpha i} \mathbf{B})^2$$

Modelo nulo (Q_W)

```
modNul <- svyglm(Income ~ 1, design = diseno)
fit_Nul <- lm(wk ~ 1, data = encuesta)
qw <- predict(fit_Nul)

encuesta %<gt;% mutate(wk1 = wk/qw)

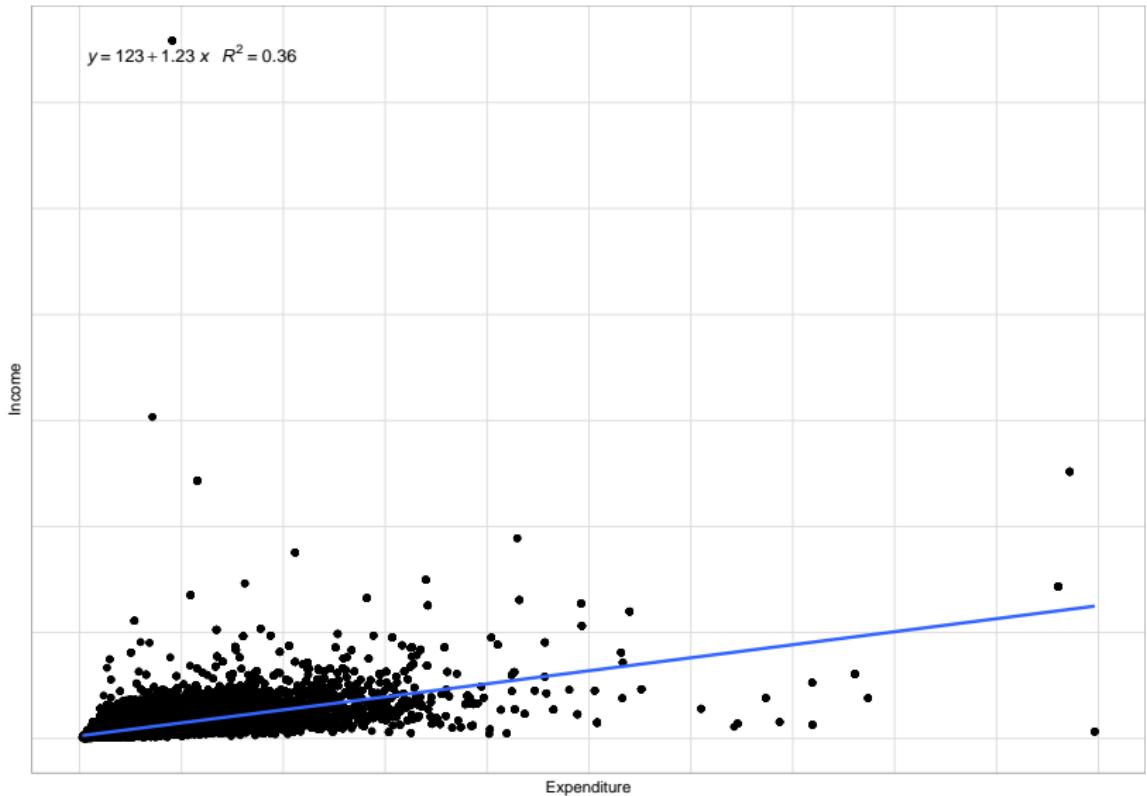
diseno_qwgt <- encuesta %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk1,
    nest = T
  )
modNul_qw <- svyglm(Income ~ 1, design = diseno_qwgt)
```

Scaterplot con los datos poblacionales

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

Scaterplot con los datos poblacionales



Modelo poblacional

```
library(modelsummary)
fit <- lm(Income ~ Expenditure, data = BigCity)
modelsummary(list(Pob = fit), statistic = NULL,
            title = "Modelo BigCity",
            output = "markdown",
            gof_omit = 'BIC|Log|AIC|F' )
```

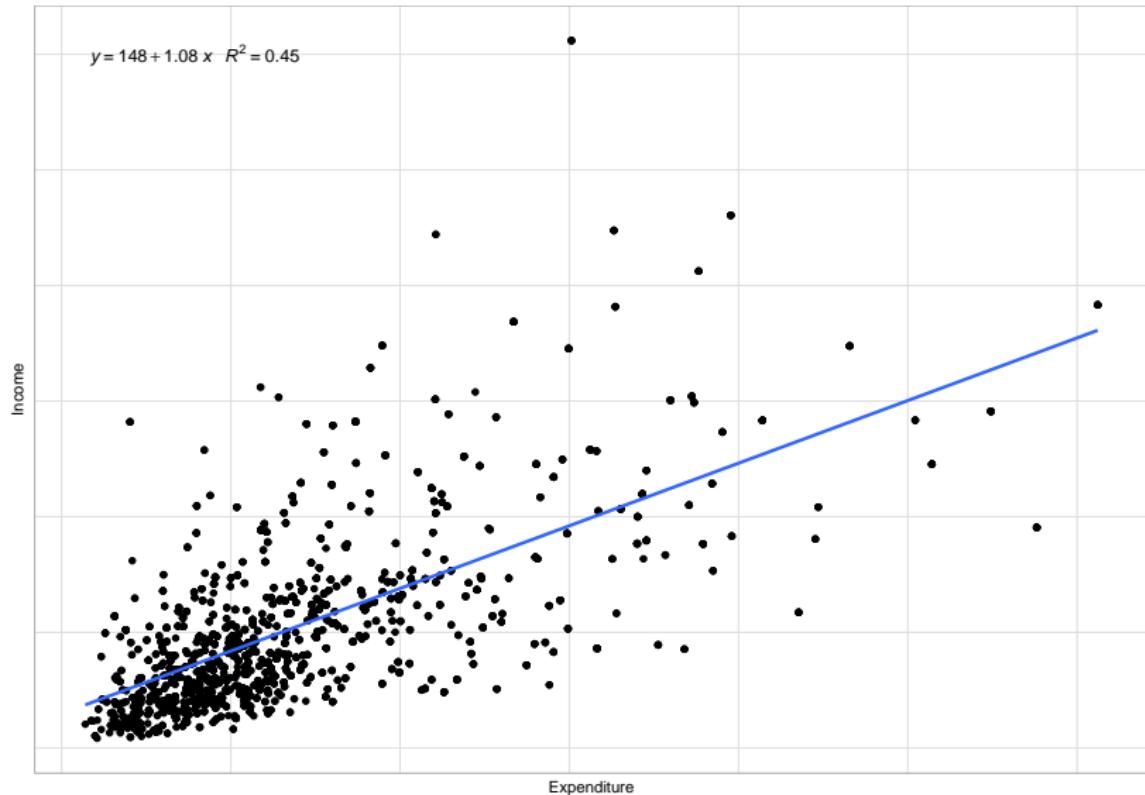
Table 1: Modelo BigCity

	Pob
(Intercept)	123.337
Expenditure	1.229
Num.Obs.	150266
R2	0.359
R2 Adj.	0.359
RMSE	461.74

Scaterplot con los datos encuesta sin ponderar

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

Scaterplot con los datos encuesta sin ponderar



Modelo sin ponderar

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

Modelo sin ponderar

Table 2: Modelo encuesta Sin ponderar

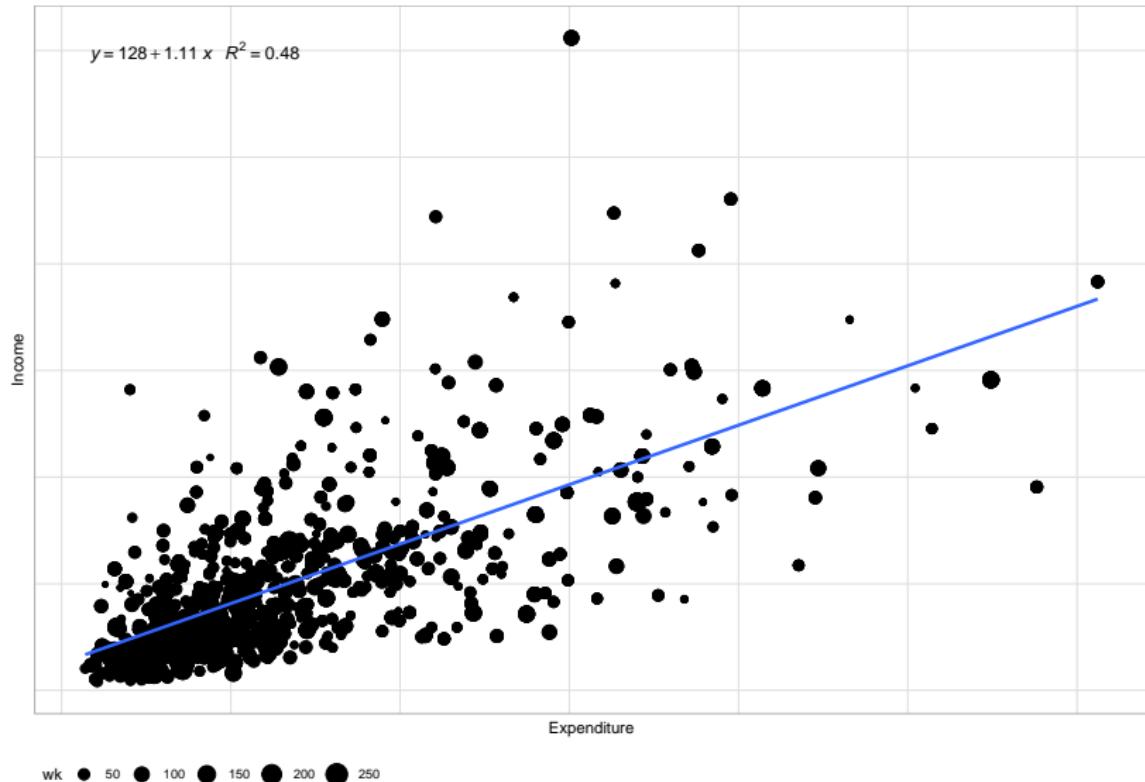
	Income
Expenditure	1.083*** (0.024)
Constant	148.200*** (10.130)
N	2597
R-squared	0.448
Adj. R-squared	0.447
Residual Std. Error	283.900 (df = 2595)
F Statistic	2102.000*** (df = 1; 2595)

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

Scaterplot con los datos encuesta ponderado

```
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 + stat_poly_eq(formula = y~x,
                           aes(weight = wk,
                               label = paste(..eq.label..,
                               ..rr.label.., sep = "~~~")),
                           parse = TRUE)
```

Scaterplot con los datos encuesta sin ponderar



Modelo ponderado lm

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

Modelo ponderado Im

Table 3: Modelo encuesta ponderada

	Income
Expenditure	1.115*** (0.023)
Constant	128.400*** (9.910)
N	2597
R-squared	0.476
Adj. R-squared	0.476
Residual Std. Error	2129.000 (df = 2595)
F Statistic	2358.000*** (df = 1; 2595)

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

Modelo ponderado svyglm

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

Calculo del R^2

$$R^2 = 1 - \frac{SSE}{SST}$$

donde

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = (y_i - \mathbf{x}_i \mathbf{B})^2$$

$$R^2_{weighted} = 1 - \frac{WSSE}{WSST}$$

Calculo del R^2

```
s1$dispersion
```

```
##      variance      SE
## [1,]    78320 10590
```

```
s0$dispersion
```

```
##      variance      SE
## [1,]    149477 17083
```

```
(R2 = 1-78320/149477)
```

```
## [1] 0.476
```

```
n = sum(diseno_qwgt$variables$wk)
```

```
(R2Adj = 1-((1-R2)*(n-1)/(n-1-1)))
```

```
## [1] 0.476
```

Resumen del Modelo

Table 4: Modelo encuesta ponderada, svyglm

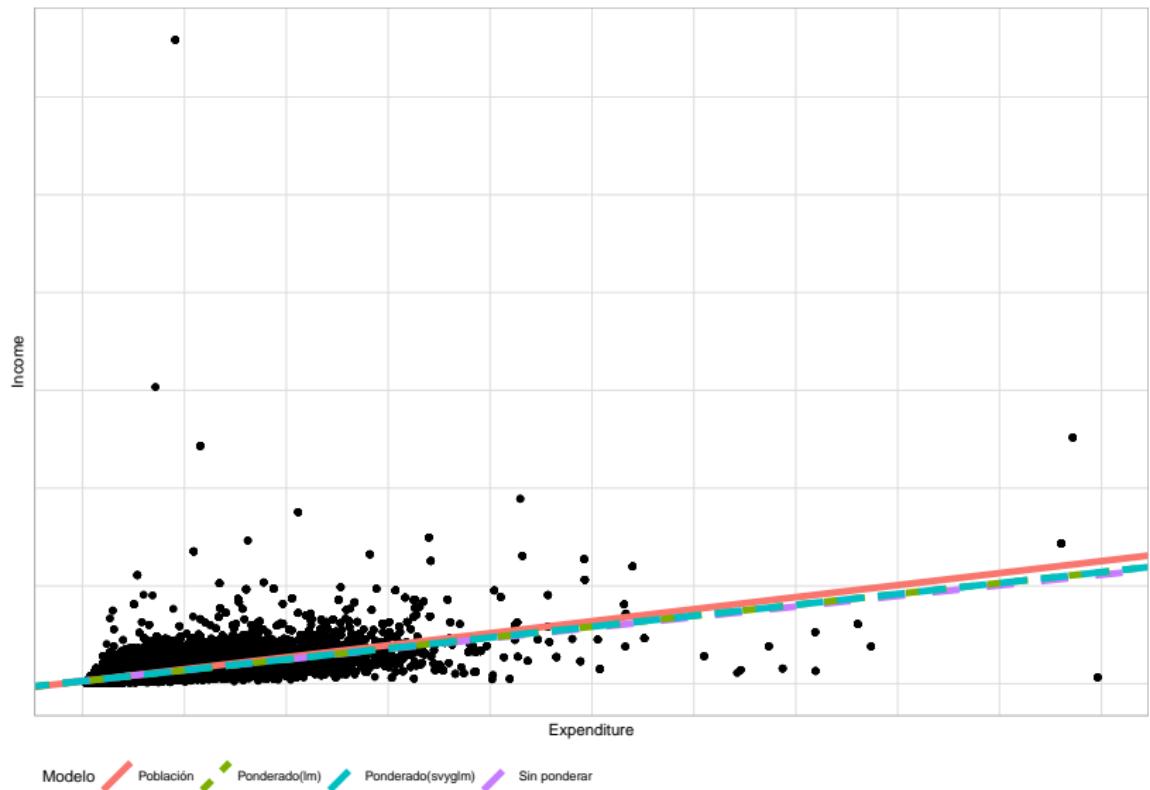
Income	
Expenditure	1.115*** (0.073)
Constant	128.400*** (23.420)
N	2597
AIC	37073.000

***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

	Pob
(Intercept)	123.337
Expenditure	1.229
Num.Obs.	150266
R2	0.359
R2 Adj.	0.359
AIC	2270206.0
F	84052.758
RMSE	461.74

Comparando los resultados

	Sin Pond	Ponde(lm)	Ponde(svyglm)
(Intercept)	148.227 p = (0.000)	128.439 p = (0.000)	128.439 p = (0.000)
Expenditure	1.083 p = (0.000)	1.115 p = (0.000)	1.115 p = (0.000)
Num.Obs.	2597	2597	
R2	0.448	0.476	
R2 Adj.	0.447	0.476	
AIC	36712.3	37073.2	21.0
F	2102.108	2357.662	230.264
RMSE	283.87	2129.19	279.91

Metodología de los Q_Weighting de pfefferman

```
fit_wgt <- lm(wk ~ Expenditure, data = encuesta)
wgt_hat <- predict(fit_wgt)
encuesta %<-% mutate(wk2 = wk/wgt_hat)

diseno_qwgt <- encuesta %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk2,
    nest = T
  )
```

Modelos empleando los Q_Weighting

```
fit_svy_qwgt <- svyglm(Income ~ Expenditure,  
                         design = diseno_qwgt)  
modNul <- svyglm(Income ~ 1, design = diseno_qwgt)  
s0 <- summary(modNul)  
s1 <- summary(fit_svy_qwgt)  
tidy(fit_svy_qwgt)
```

term	estimate	std.error	statistic	p.value
(Intercept)	128.018	23.3550	5.481	0
Expenditure	1.116	0.0734	15.204	0

Calculo del R^2

```
s1$dispersion
```

```
##      variance      SE
## [1,]    78053 10522
```

```
s0$dispersion
```

```
##      variance      SE
## [1,]   148800 16969
```

```
(R2 = 1-78053/148800)
```

```
## [1] 0.4755
```

```
n = sum(diseno_qwgt$variables$wk2)
```

```
(R2Adj = 1-((1-R2)*(n-1)/(n-1-1)))
```

```
## [1] 0.4752
```

Modelos empleando los Q_Weighting

Table 7: Comprando Modelos con Q Weighting

	svyglm(wgt)	svyglm(qwgt)
(Intercept)	128.439	128.018
	p = (0.000)	p = (0.000)
Expenditure	1.115	1.116
	p = (0.000)	p = (0.000)
AIC	21.0	20.8
F	230.264	231.166
RMSE	279.91	279.43

Modelo escogido

```
diseno_qwgt %>% mutate(Age2 = Age^2)
mod_svy <- svyglm(
  Income ~ Expenditure + Zone + Sex + Age2 ,
  design = diseno_qwgt)
s1 <- summary(mod_svy)
s0 <- summary(modNul)
stargazer(mod_svy, header = FALSE, single.row = T,
           title = "Modelo",
           style = "ajps", omit.stat=c("bic", "ll"))
```

Resumen del Modelo escogido

Table 8: Modelo

Income	
Expenditure	1.089*** (0.075)
ZoneUrban	48.770** (23.680)
SexMale	8.093 (7.313)
Age2	0.012** (0.005)
Constant	91.630*** (25.560)
N	2597
AIC	37037.000

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

Calculo del R^2

```
s1$dispersion
```

```
##      variance      SE
## [1,]    77063 10268
```

```
s0$dispersion
```

```
##      variance      SE
## [1,]   148800 16969
```

```
(R2 = 1-76821/148800)
```

```
## [1] 0.4837
```

```
n = sum(diseno_qwgt$variables$wk2)
```

```
(R2Adj = 1-((1-R2)*(n-1)/(n-1-1)))
```

```
## [1] 0.4835
```

Residuales estandarizados

$$r_{pi} = \left(y_i - \mu_i (\hat{\boldsymbol{B}}_w) \right) \sqrt{\frac{w_i}{V(\hat{\mu}_i)}}$$

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

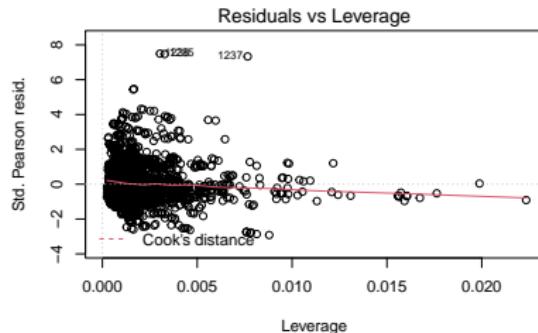
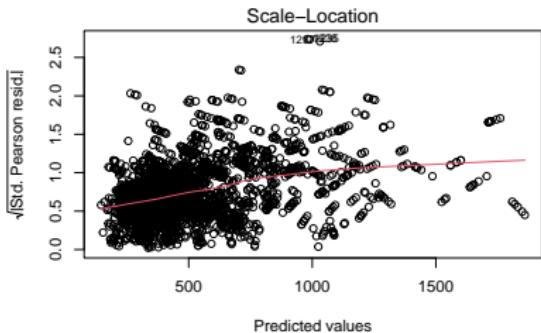
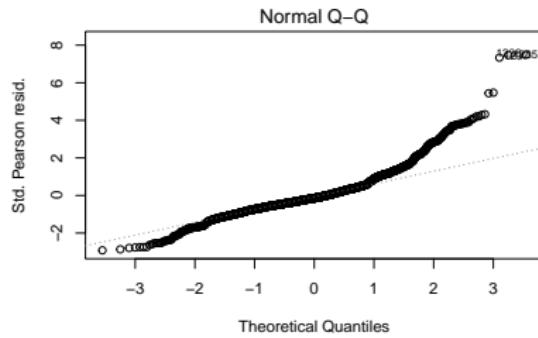
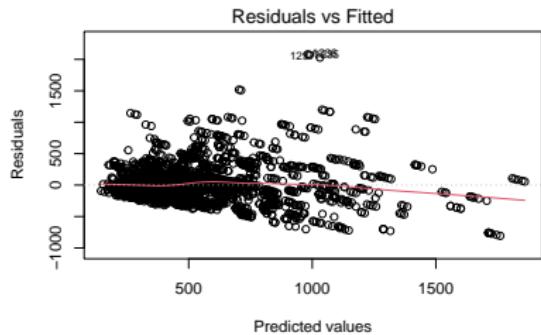
donde

$$W = diag \left\{ \frac{w_1}{V(\hat{\mu}_1) [g'(\mu_1)]^2}, \dots, \frac{w_n}{V(\hat{\mu}_1) [g'(\mu_n)]^2} \right\}$$

con g es una función de enlace que es especificada mediante un Modelo lineal generalizado.

Diagnostico del Modelo

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



Pruebas de normalidad

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

Algunas librerías que podemos emplear son:

```
library(normtest) #REALIZA 5 PRUEBAS  
library(nortest)  #REALIZA 10 PRUEBAS  
library(moments) #REALIZA 1 PRUEBA
```

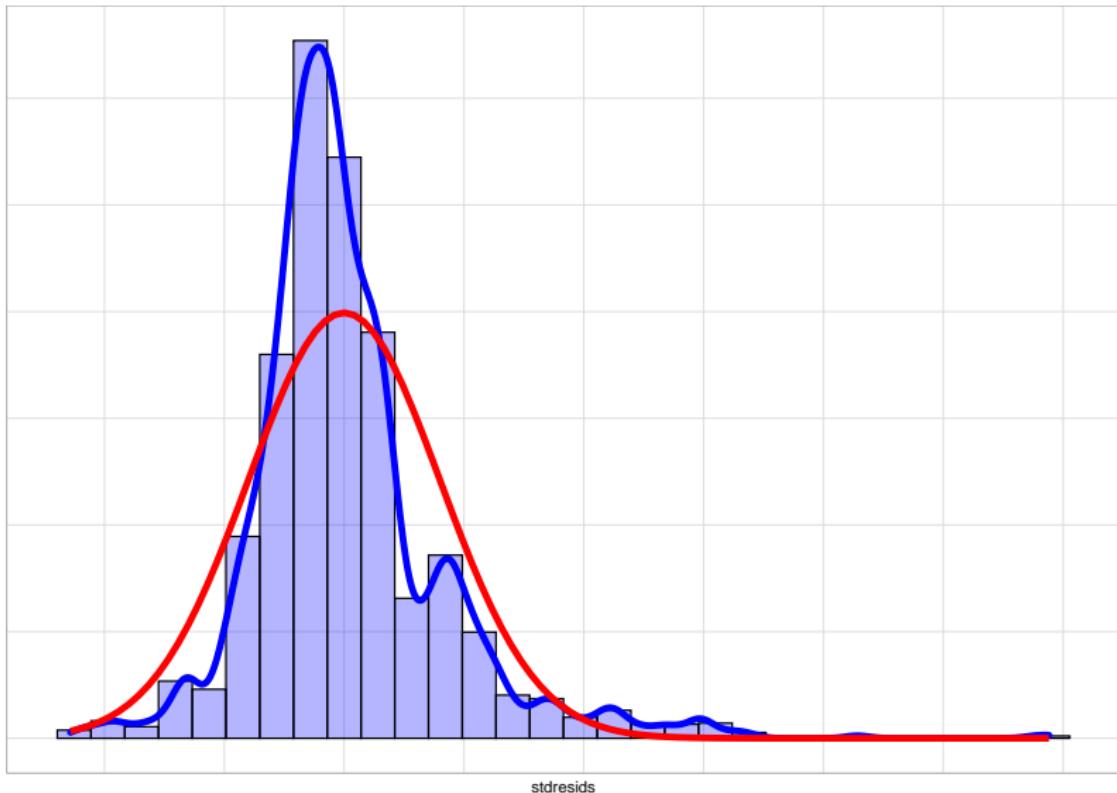
Extrayendo residuales standarizados

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

Histograma de los residuales

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

Histograma de los residuales



Pruebas de normalidad Kolmogorov-Smirnov

```
nortest::lillie.test(diseno_qwgt$variables$stdresids)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: diseno_qwgt$variables$stdresids
## D = 0.11, p-value <2e-16
```

Varianza constante

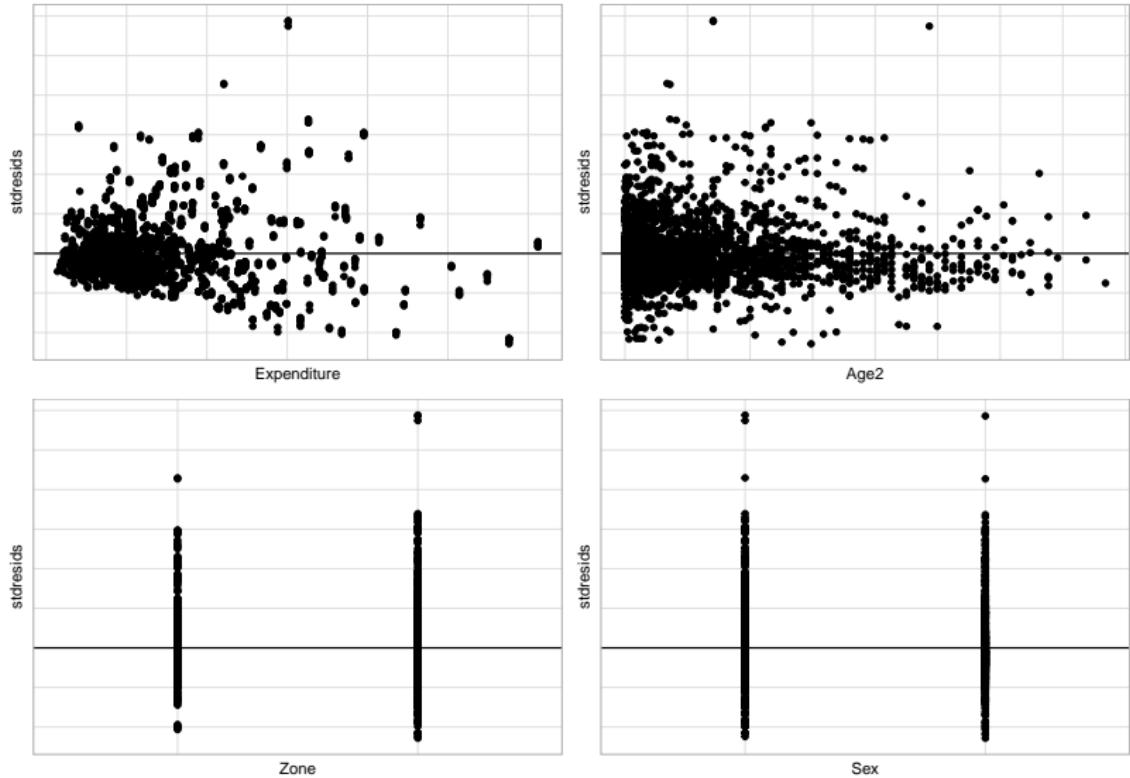
```
library(patchwork)
diseno_qwgt$variables %<>%
  mutate(pred = predict(mod_svy))
g2 <- ggplot(data = diseno_qwgt$variables,
              aes(x = Expenditure, y = stdresids))+
  geom_point() +
  geom_hline(yintercept = 0) + theme_cepal()
g3 <- ggplot(data = diseno_qwgt$variables,
              aes(x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



Distancia de cook

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

donde,

- ▶ 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.
- ▶ ϕ = parámetro de dispersión en el glm
- ▶ $\widehat{\text{Var}} \left(U_w (\hat{\mathbf{B}}_w) \right)$ = 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 :

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

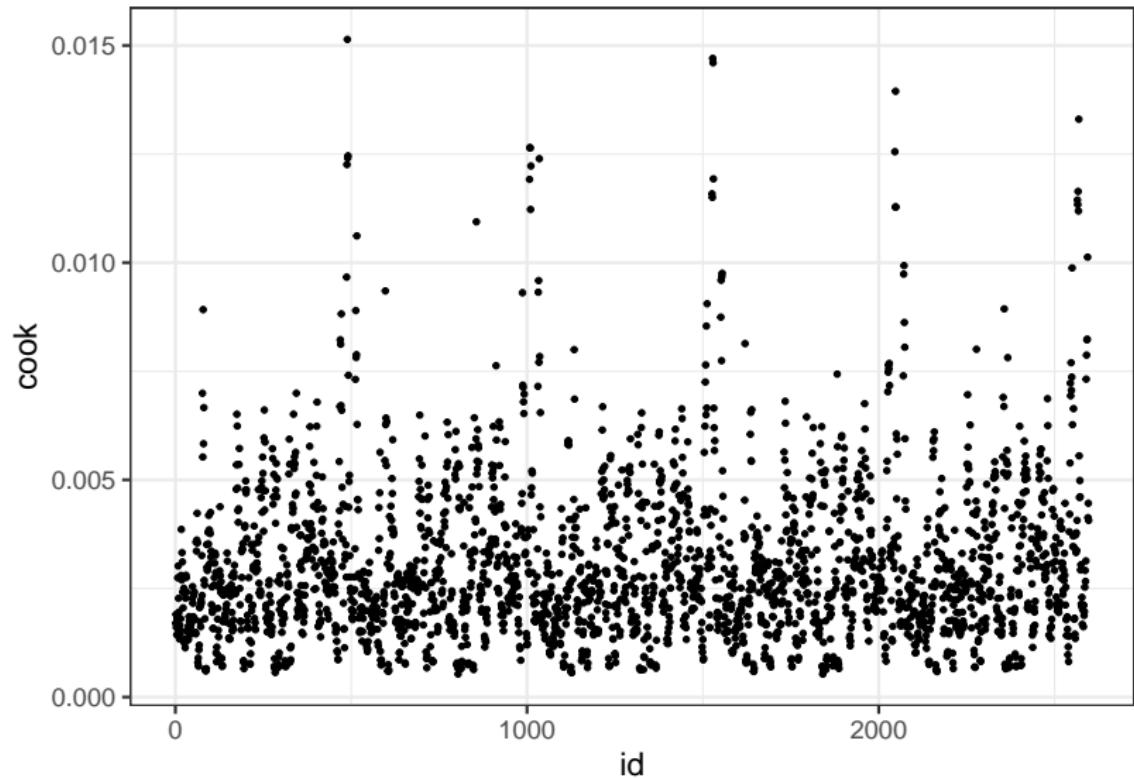
donde df = grados de libertad basados en el diseño.

Por otro lado, la literatura considera a las observaciones influentes cuando sean mayores a 2 o 3

Detección de observaciones influyentes (Distancia de cook)

```
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)



$D_f Beta_{(i)}$

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

Donde $\mathbf{A} = \mathbf{X}^t \mathbf{W} \mathbf{X}$ $\hat{B}_{(i)}$ 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.

$D_f Beta_{(i)}$

$$D_f Beta_{(i)} = \frac{c_{ji} e_i / (1 - h_{ii})}{\sqrt{\nu(\hat{B}_j)}}$$

donde:

- ▶ c_{ji} = es el ji-estimo elemento de $\mathbf{A}^{-1} w_i^2 \mathbf{X}_{(i)} \mathbf{X}_{(i)}^t \mathbf{A}^{-1}$
- ▶ El estimador de $\nu(\hat{B}_j)$ basado en el Modelo se obtiene como:
 $\nu_m(\hat{B}_j) = \hat{\sigma} \sum_{i=1}^n c_{ji}^2$ con $\hat{\sigma} = \sum_{i \in s} w_i e^2 / (\hat{N} - p)$ y
 $\hat{N} = \sum_{i \in s} w_i$
- ▶ La i-ésima observación es influyente para B_j si
 $|D_f Beta_{(i)}| \geq \frac{z}{\sqrt{n}}$ 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

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

```
d_dfbetas = data.frame(t(svydfbetas(mod_svy)$Dfbetas))
colnames(d_dfbetas) <- paste0("Beta_", 1:5)
d_dfbetas %>% slice(1:10L)
```

Beta_1	Beta_2	Beta_3	Beta_4	Beta_5
0.0002	-1e-04	-0.0013	0.0045	0.0019
0.0024	-1e-04	-0.0019	-0.0056	0.0010
0.0018	-3e-04	-0.0018	0.0061	-0.0019
-0.0014	6e-04	0.0013	-0.0051	0.0005
-0.0028	5e-04	0.0013	0.0048	0.0009
-0.0027	5e-04	0.0011	0.0044	0.0015
-0.0027	5e-04	0.0011	0.0043	0.0016
-0.0020	1e-03	-0.0034	0.0098	0.0094
0.0043	1e-03	-0.0047	-0.0124	0.0014
0.0020	8e-04	-0.0042	0.0118	-0.0024

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

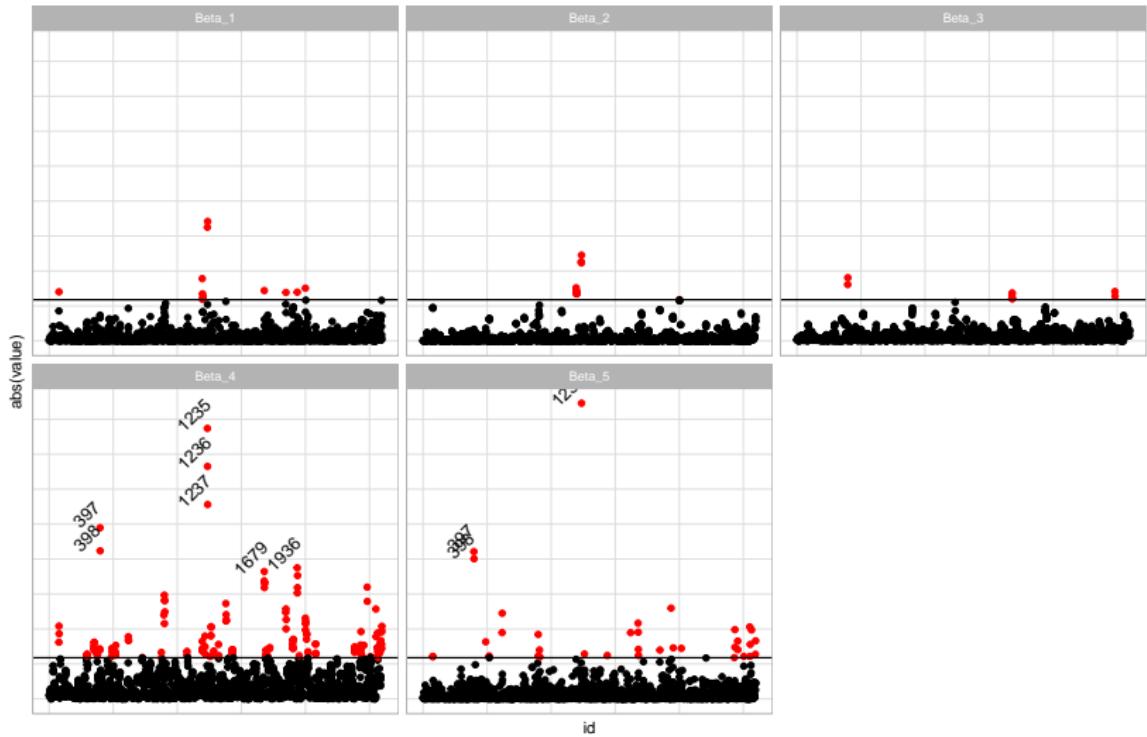
```
d_dfbetas$id <- 1:nrow(d_dfbetas)
d_dfbetas <- reshape2::melt(d_dfbetas, id.vars = "id")
cutoff <- svydfbetas(mod_svy)$cutoff
d_dfbetas %>%
  mutate(
    Criterio = ifelse(abs(value) > cutoff, "Si", "No"))

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

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

```
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 ($D_f Beta_{(i)j}$)



Criterio • Si • No

Matriz H asociada al PMLE

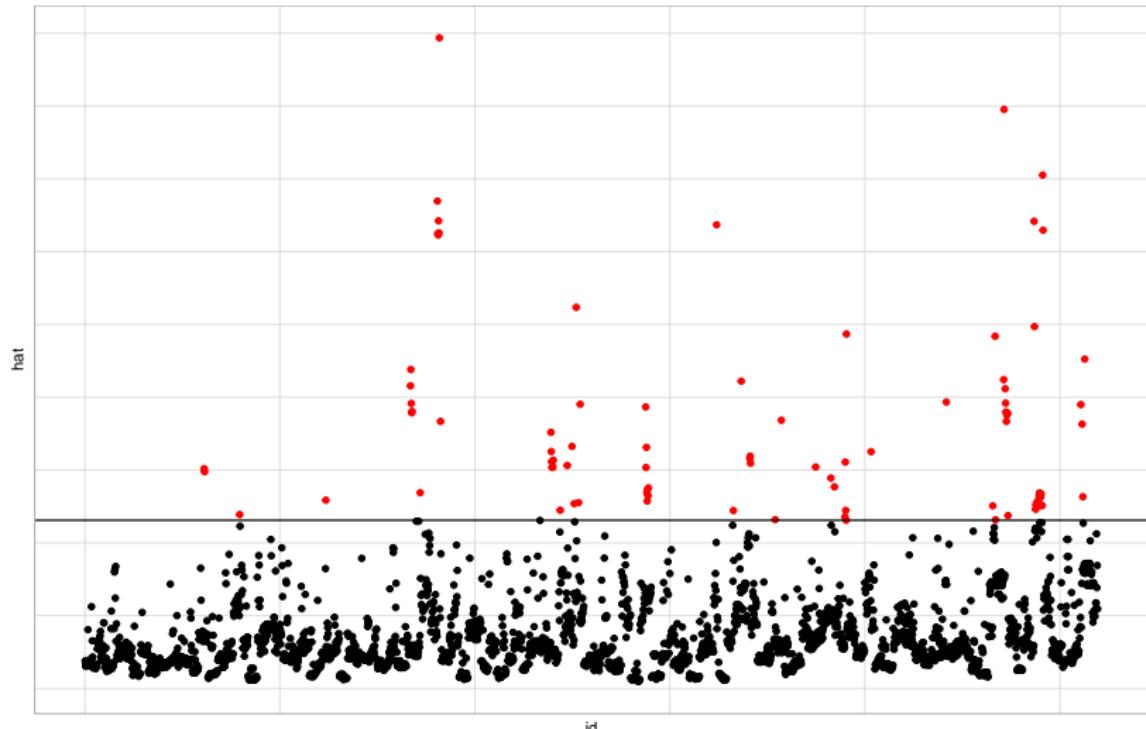
- ▶ La matriz asociada al Estimador de Pseudo Máxima Verosimilitud (PMLE) de $\hat{\boldsymbol{B}}$ es $\boldsymbol{H} = \boldsymbol{X}\boldsymbol{A}^{-1}\boldsymbol{X}^{-t}\boldsymbol{W}$ cuya diagonal esta dado por $h_{ii} = \boldsymbol{x}_i^t\boldsymbol{A}^{-1}\boldsymbol{x}_i^{-t}w_i$.
- ▶ Una observación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

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})



C_cutoff ● Si ● No

Estadístico $D_f Fits_{(i)}$

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

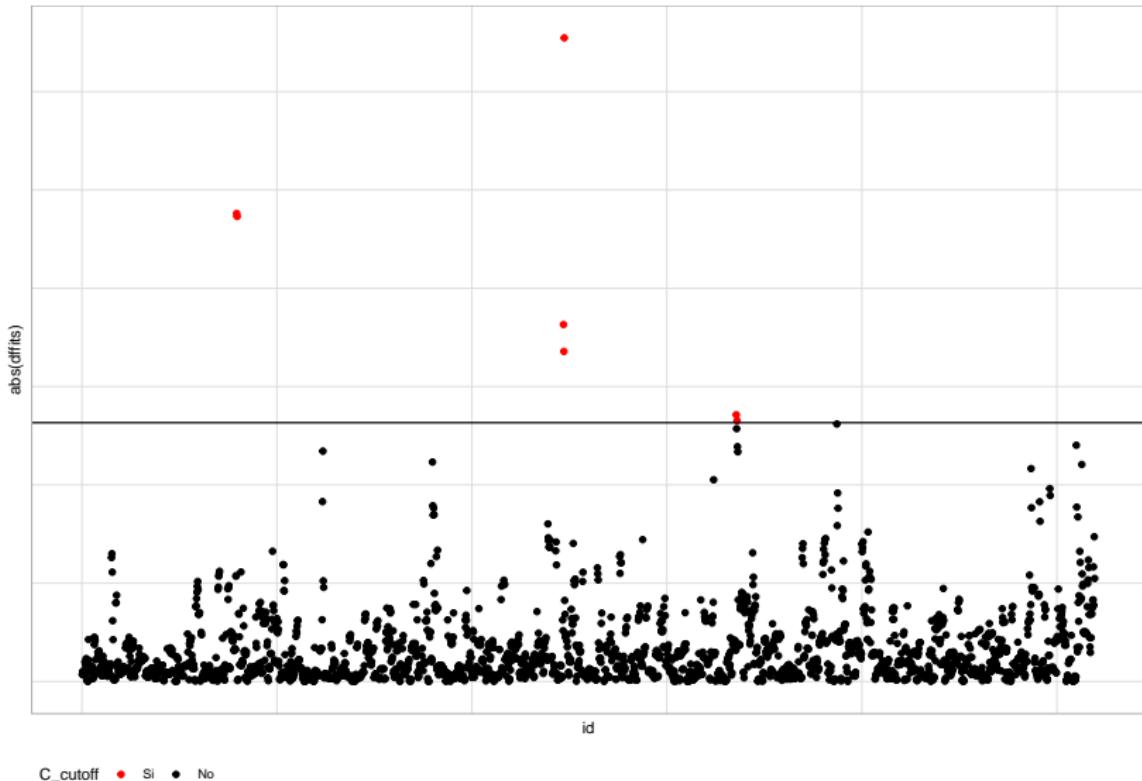
Donde, $\sqrt{v(\hat{\beta}_j)}$ puede ser aproximada por el diseño o el Modelo.

La i-ésima observación se considera influyente en el ajuste del Modelo si $| DfFits(i) | \geq z \sqrt{\frac{p}{n}}$ con $z = 2$ o 3

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 ($D_f \text{Fits}_{(i)}$)



Inferencia sobre los parámetros del Modelo

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

$$\hat{B} \pm t_{(1-\frac{\alpha}{2}, df)} \times se(\hat{B})$$

Estimación del dato

$$\hat{E}(y_i \mid \mathbf{x}_{obs,i}) = \mathbf{x}_{obs,i} \hat{\beta}$$

$$\hat{E}(y_i \mid \mathbf{x}_{obs,i}) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i}$$

term	estimate	std.error	statistic	p.value
(Intercept)	91.6319	25.5628	3.585	0.0005
Expenditure	1.0893	0.0750	14.523	0.0000
ZoneUrban	48.7667	23.6791	2.059	0.0417
SexMale	8.0933	7.3125	1.107	0.2707
Age2	0.0115	0.0048	2.376	0.0192

$$\hat{E}(y_i \mid \mathbf{x}_{obs,i}) = 91.6319 + 1.0893x_{1i} + 48.7667x_{2i} + 8.0933x_{3i} + 0.0115x_{4i}$$

Estimación del dato

(Intercept)	Expenditure	ZoneUrban	SexMale	Age2
1	247.9	0	1	3025
1	247.9	0	0	2116
1	247.9	0	1	169
1	164.6	0	1	961
1	164.6	0	0	784
1	164.6	0	0	81
1	164.6	0	0	25

$$\hat{y}_i = 91.6319 + 1.0893(247.9) + 48.7667(0) + 8.0933(1) + 0.0115(55^2) = 404$$

Estimando el IC de predicción

$$\text{var} \left(\hat{E}(y_i | \mathbf{x}_{obs,i}) \right) = \mathbf{x}_{obs,i}^t \text{cov}(\beta) \mathbf{x}_{obs,i}$$

```
vcov(mod_svy)
```

	(Intercept)	Expenditure	ZoneUrban	SexMale	Age2
(Intercept)	653.4555	-1.5626	-106.1848	7.1785	-0.0506
Expenditure	-1.5626	0.0056	-0.3421	-0.0620	0.0001
ZoneUrban	-106.1848	-0.3421	560.6998	2.0210	-0.0169
SexMale	7.1785	-0.0620	2.0210	53.4730	0.0012
Age2	-0.0506	0.0001	-0.0169	0.0012	0.0000

```
xobs <- model.matrix(mod_svy) %>%
  data.frame() %>% slice(1) %>% as.matrix()
cov_beta <- vcov(mod_svy) %>% as.matrix()
as.numeric(sqrt((xobs) %*% cov_beta %*% t(xobs)))
```

```
## [1] 18.09
```

Intervalo de confianza para la predicción

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

Utilizando la función predict

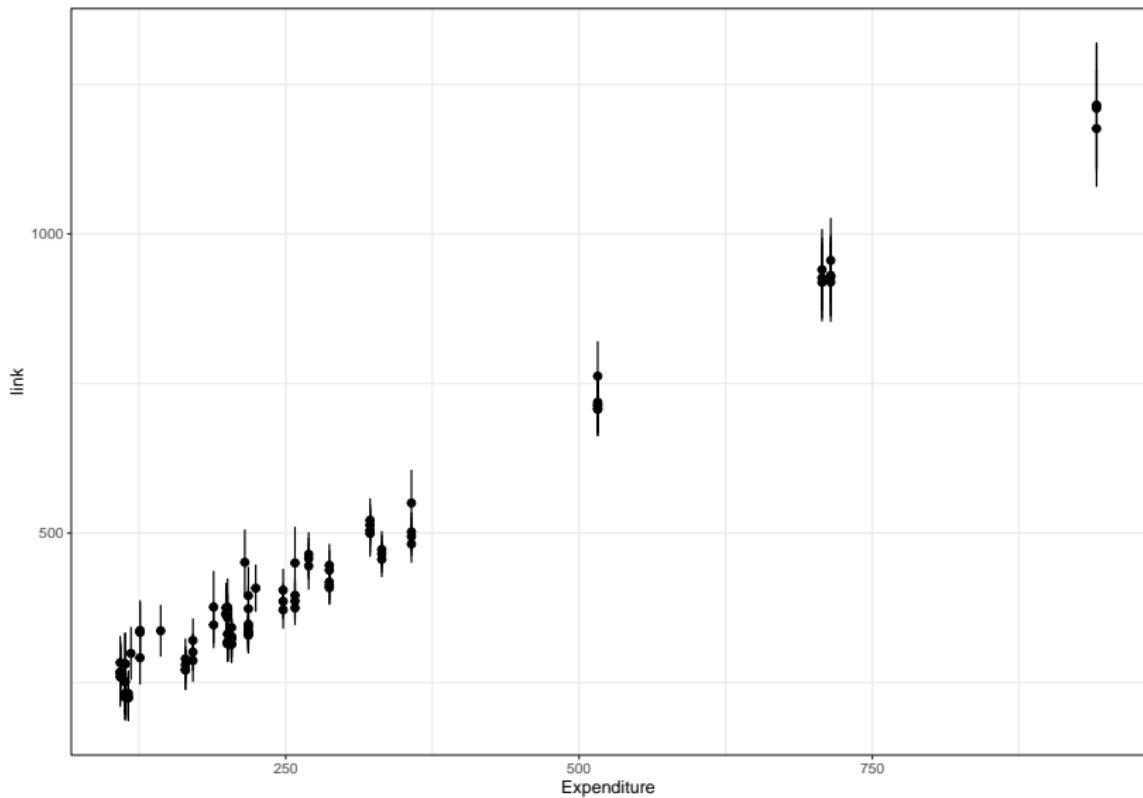
```
pred <- data.frame(predict(mod_svy, type = "link"))
pred_IC <- data.frame(
  confint(predict(mod_svy, type = "link")))
colnames(pred_IC) <- c("Lim_Inf", "Lim_Sup")
pred <- bind_cols(pred, pred_IC)
pred$Expenditure <- encuesta$Expenditure
pred %>% slice(1:6L)
```

link	SE	Lim_Inf	Lim_Sup	Expenditure
404.6	18.09	369.1	440.0	247.9
386.0	14.84	356.9	415.1	247.9
371.7	15.93	340.4	402.9	247.9
290.0	17.24	256.3	323.8	164.6
279.9	15.89	248.8	311.0	164.6
271.8	16.92	238.7	305.0	164.6

Scaterplot de la predicción

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

Scaterplot de la predicción



Predictión fuera de las observaciones.

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

$$\hat{y}_i = 91.6319 + 1.0893(1600) + 48.7667(0) + 8.0933(1) + 0.0115(40^2) = 1910$$

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

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

```
## [1] 97.68
```

Intervalo de confianza para la predicción

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

Predictión fuera de las observaciones.

```
predict(mod_svy, newdata = datos_nuevos, type = "link")  
  
##      link      SE  
## 1 1910 97.7  
  
confint(predict(mod_svy,newdata = datos_nuevos))
```

2.5 %	97.5 %
1718	2101