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

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

$$E(y \mid x) = B_0 + B_1 x$$

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

$$\hat{B} = \left(\boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{W} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{W} \boldsymbol{y}$$

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

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

Modelo nulo (Q_W)

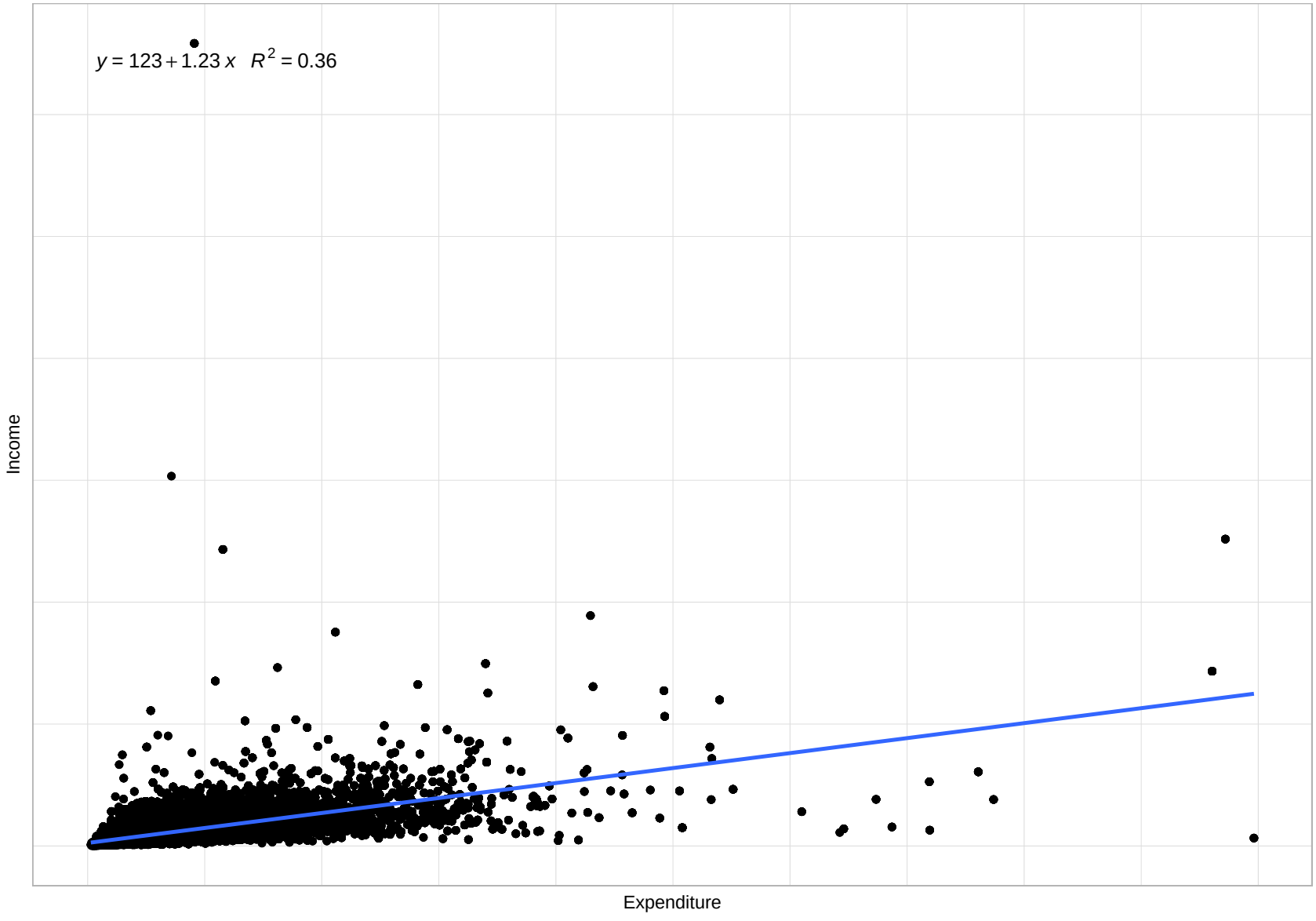
```
modNul <- svyglm(Income ~ 1, design = diseno)
fit_Nul <- lm(wk ~ 1, data = encuesta)
qw <- predict(fit_Nul)
```

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

```
fit <- lm(Income ~ Expenditure, data = BigCity)
stargazer(fit, header = FALSE, title = "Modelo BigCity",
          style = "ajps")
```

modelo poblacional

Table 1: Modelo BigCity

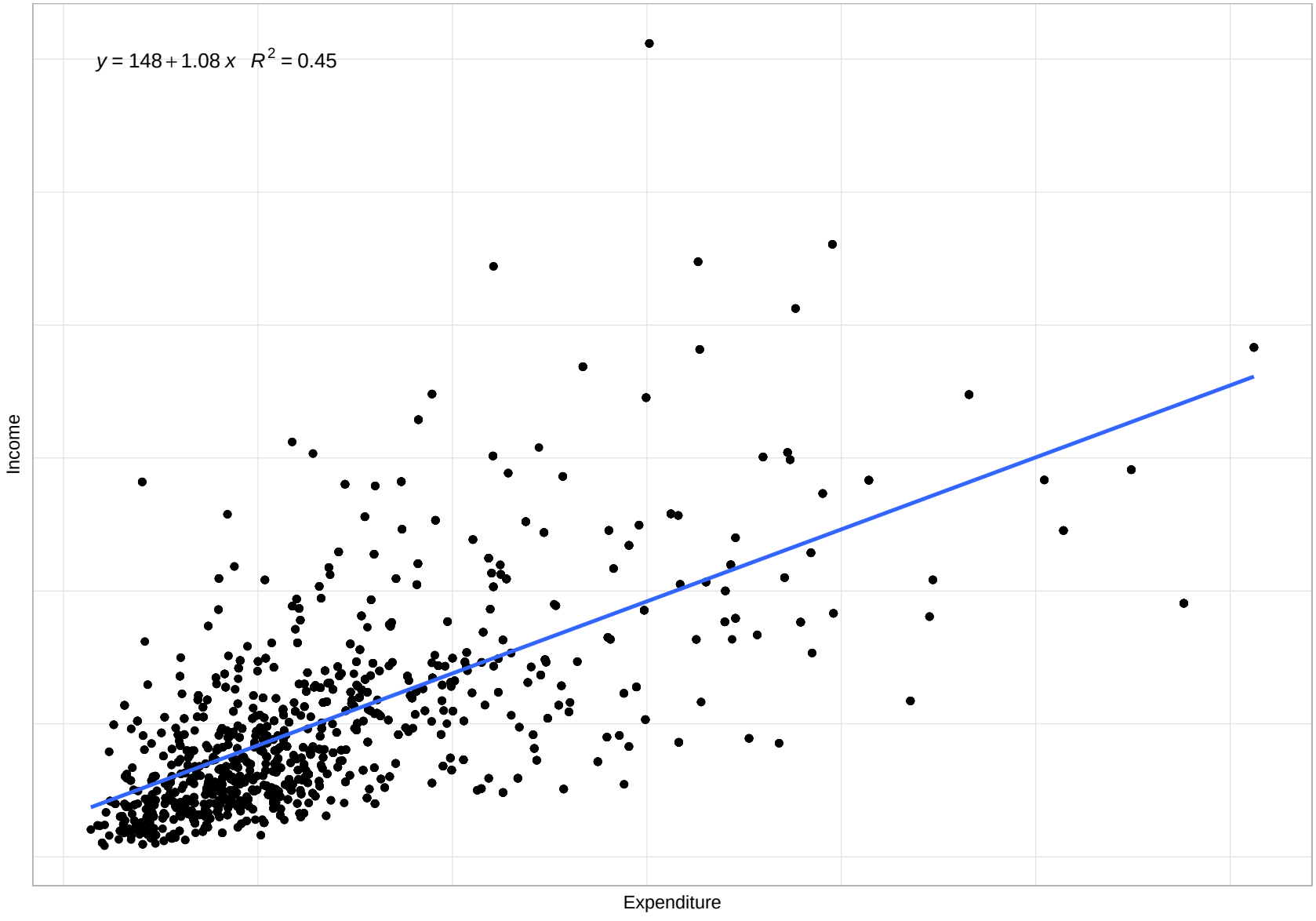
	Income
Expenditure	1.229***
	(0.004)
Constant	123.300***
	(1.988)
N	150266
R-squared	0.359
Adj. R-squared	0.359
Residual Std. Error	461.700 (df = 150264)
F Statistic	84053.000*** (df = 1; 150264)

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

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

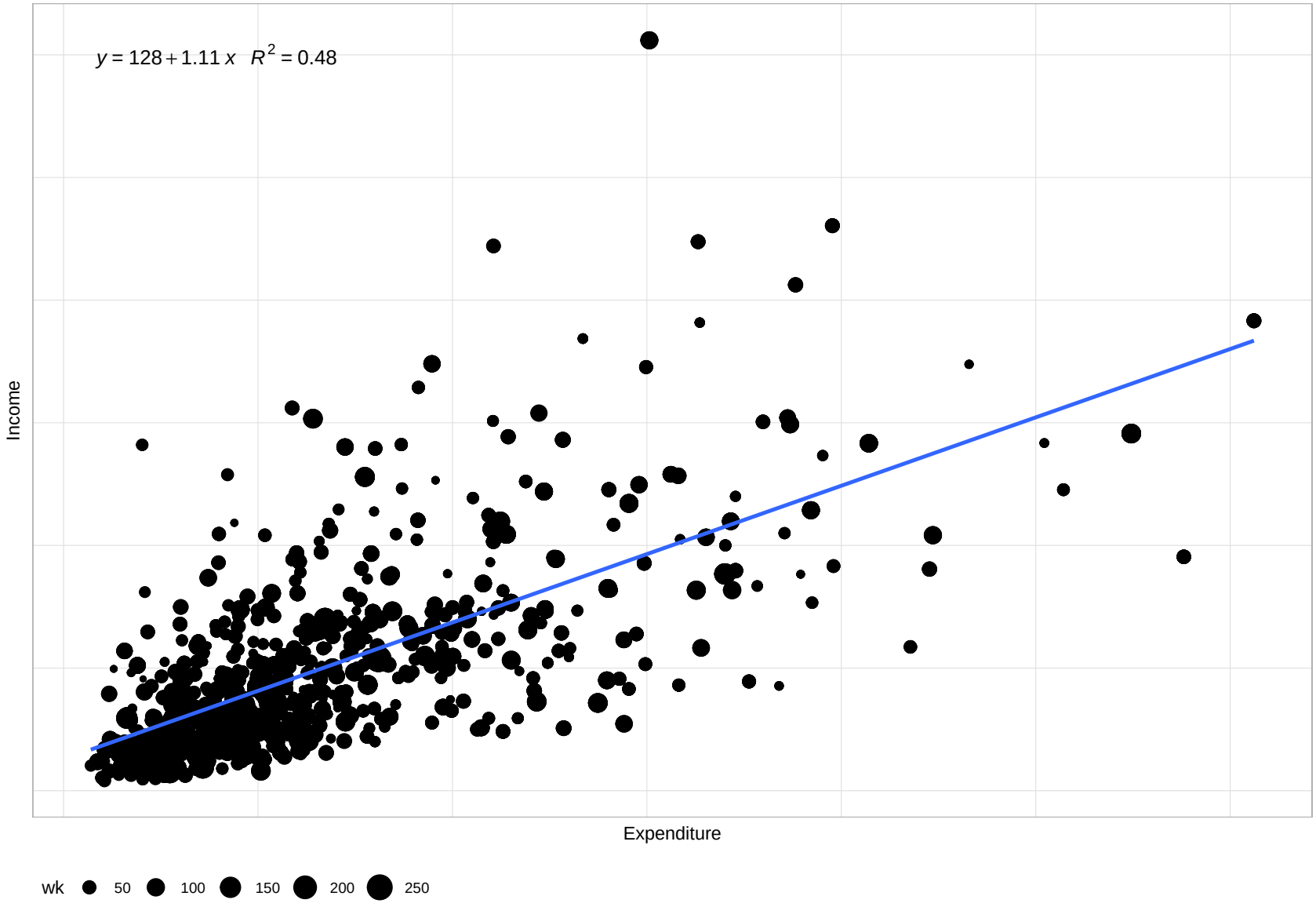
	Income
Expenditure	1.083^{***}
	(0.024)
Constant	148.200^{***}
	(10.130)
N	2597
R-squared	0.448
Adj. R-squared	0.447
Residual Std. Error	283.900 (df = 2595)
F Statistic	2102.000^{***} (df = 1; 2595)

***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 lm

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

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

Calculo del R^2

```
s1$dispersion
```

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

```
s0$dispersion
```

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

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

Resumen del modelo

Table 4: Modelo encuesta ponderada, svyglm

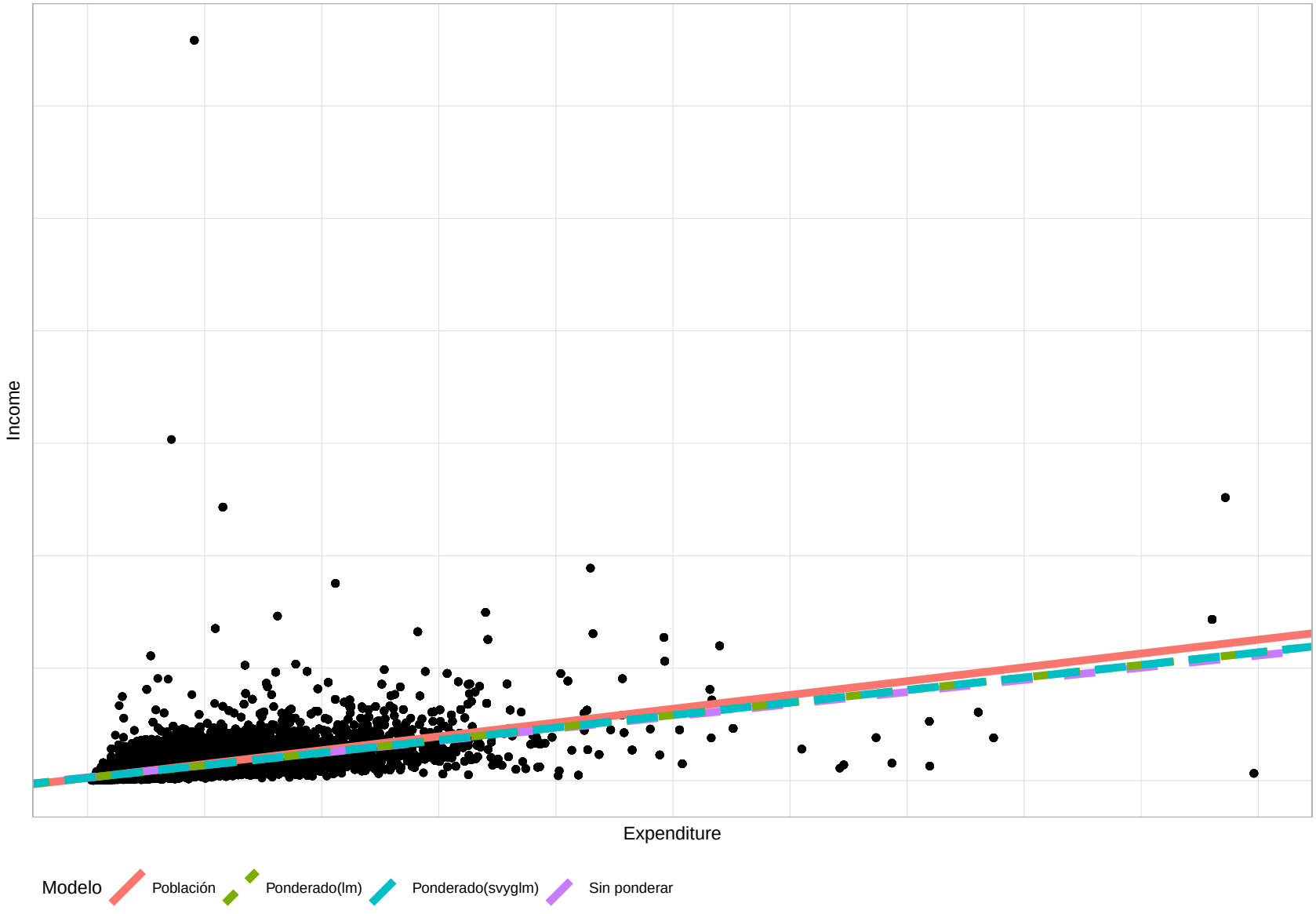
	Income
Expenditure	1.115***
	(0.073)
Constant	128.400***
	(23.420)
N	2597
Log Likelihood	−18535.000
AIC	37073.000

***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	Sin Pond	Ponde(lm)	Ponde(svy
(Intercept)	123.337	148.227	128.439	128.43
	p.v = (0.000)	p.v = (0.000)	p.v = (0.000)	p.v = (0.
Expenditure	1.229	1.083	1.115	1.115
	p.v = (0.000)	p.v = (0.000)	p.v = (0.000)	p.v = (0.
Num.Obs.	150266	2597	2597	
R2	0.359	0.448	0.476	
R2 Adj.	0.359	0.447	0.476	
AIC	2270206.0	36712.3	37073.2	21.0
F	84052.758	2102.108	2357.662	230.26
RMSE	461.74	283.87	2129.19	279.9

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_Ponde_qwgt <- lm(Income ~ Expenditure,
                     data = encuesta, weights = wk2)
fit_svy_qwgt <- svyglm(Income ~ Expenditure,
                       design = diseno_qwgt)
modNul <- svyglm(Income ~ 1, design = diseno_qwgt)
```

Calculo del R^2

```
s1 <- summary(fit_svy_qwgt)
s0 <- summary(modNul)
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 6: Comprando modelos con Q Weighting

	lm(wgt)	svyglm(wgt)	lm(qwgt)	svyglm(qw
(Intercept)	128.439	128.439	128.018	128.01
	p.v = (0.000)	p.v = (0.000)	p.v = (0.000)	p.v = (0.(
Expenditure	1.115	1.115	1.116	1.116
	p.v = (0.000)	p.v = (0.000)	p.v = (0.000)	p.v = (0.(
Num.Obs.	2597		2597	
R2	0.476		0.475	
R2 Adj.	0.476		0.475	
AIC	37073.2	21.0	37064.3	20.8
F	2357.662	230.264	2352.088	231.16
RMSE	2129.19	279.91	279.43	279.43

Modelo escogido

```
mod_svy <- svyglm(
  Income ~ Expenditure + Zone + Sex + Age^2 ,
                          design = diseno_qwgt)
stargazer(mod_svy, header = FALSE,
           title = "Modelo completo",
           style = "ajps",  omit.stat=c("bic", "ll"))
```

Resumen del modelo escogido

Table 7: Modelo completo

	Income
Expenditure	1.090***
	(0.075)
ZoneUrban	47.650**
	(23.660)
SexMale	9.171
	(7.326)
Age	1.169***
	(0.411)
Constant	71.520**
	(28.190)
N	2597
AIC	37029.000

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

Calculo del R^2

```
s1 <- summary(mod_svy)
s0 <- summary(modNul)
s1$dispersion
```

```
##       variance    SE
## [1,]    76821 10189
```

```
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 estándarizados

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

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

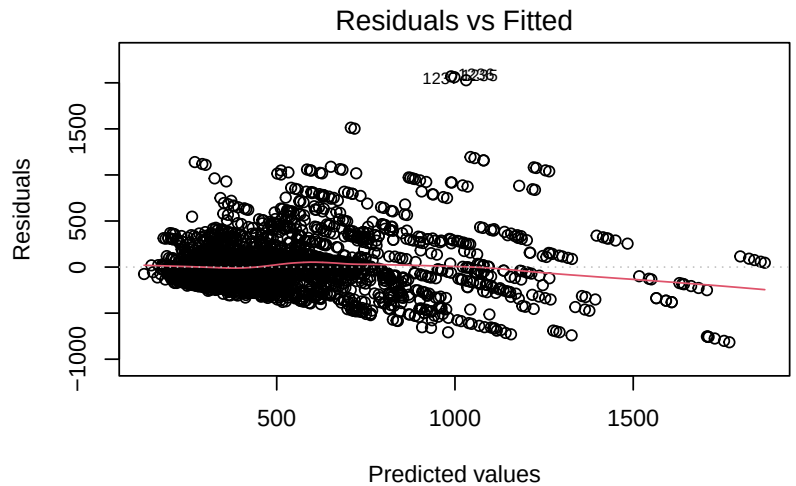
donde

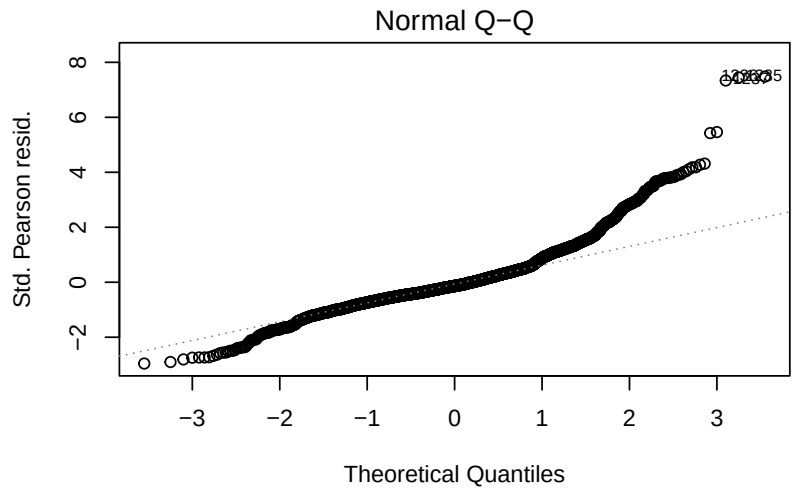
$$W = diag\left\{\frac{w_1}{V\left(\hat{\mu}_1\right)\left[g'\left(\mu_1\right)\right]^2}, \ldots, \frac{w_n}{V\left(\hat{\mu}_1\right)\left[g'\left(\mu_n\right)\right]^2}\right\}$$

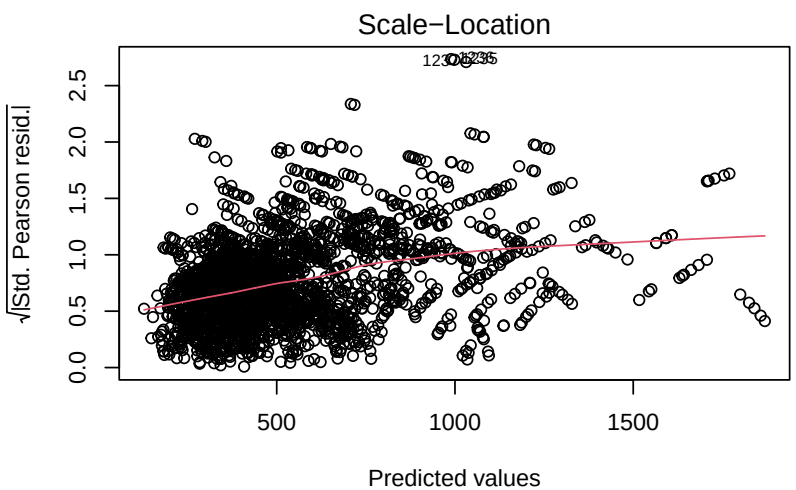
con g es una función de enlace que es especificada mediante un modelo lineal generalizado.

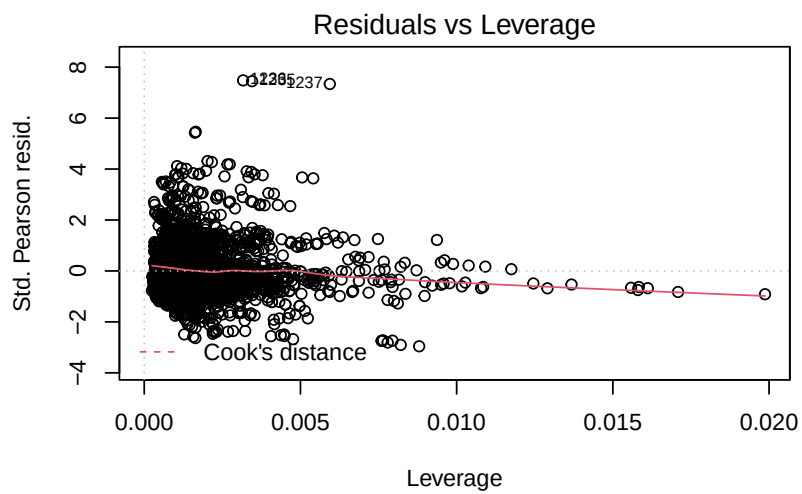
Diagnostico del modelo

```
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.

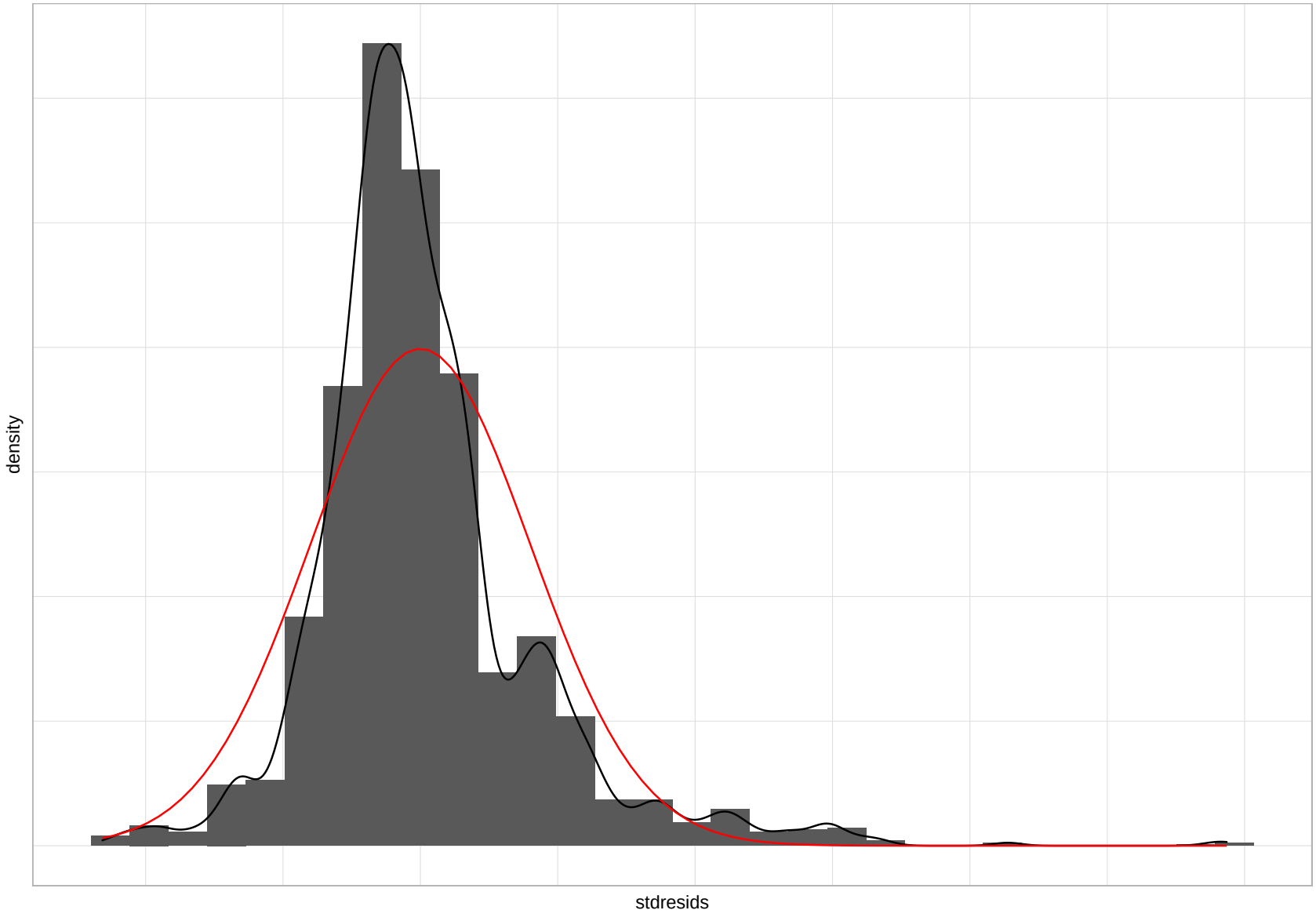
Algunas librerías que podemos emplear son:

```
library(normtest) ###REALIZA 5 PRUEBAS DE NORMALIDAD###
library(nortest) ###REALIZA 10 PRUEBAS DE NORMALIDAD###
library(moments) ###REALIZA 1 PRUEBA DE NORMALIDAD###
library(svydiags)
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..)) +
  geom_density() +
  geom_function(fun = dnorm, colour = "red") +
  theme_cepal()
```

Histograma de los residuales



Pruebas de normalidad Kolmogorov-Smirnov

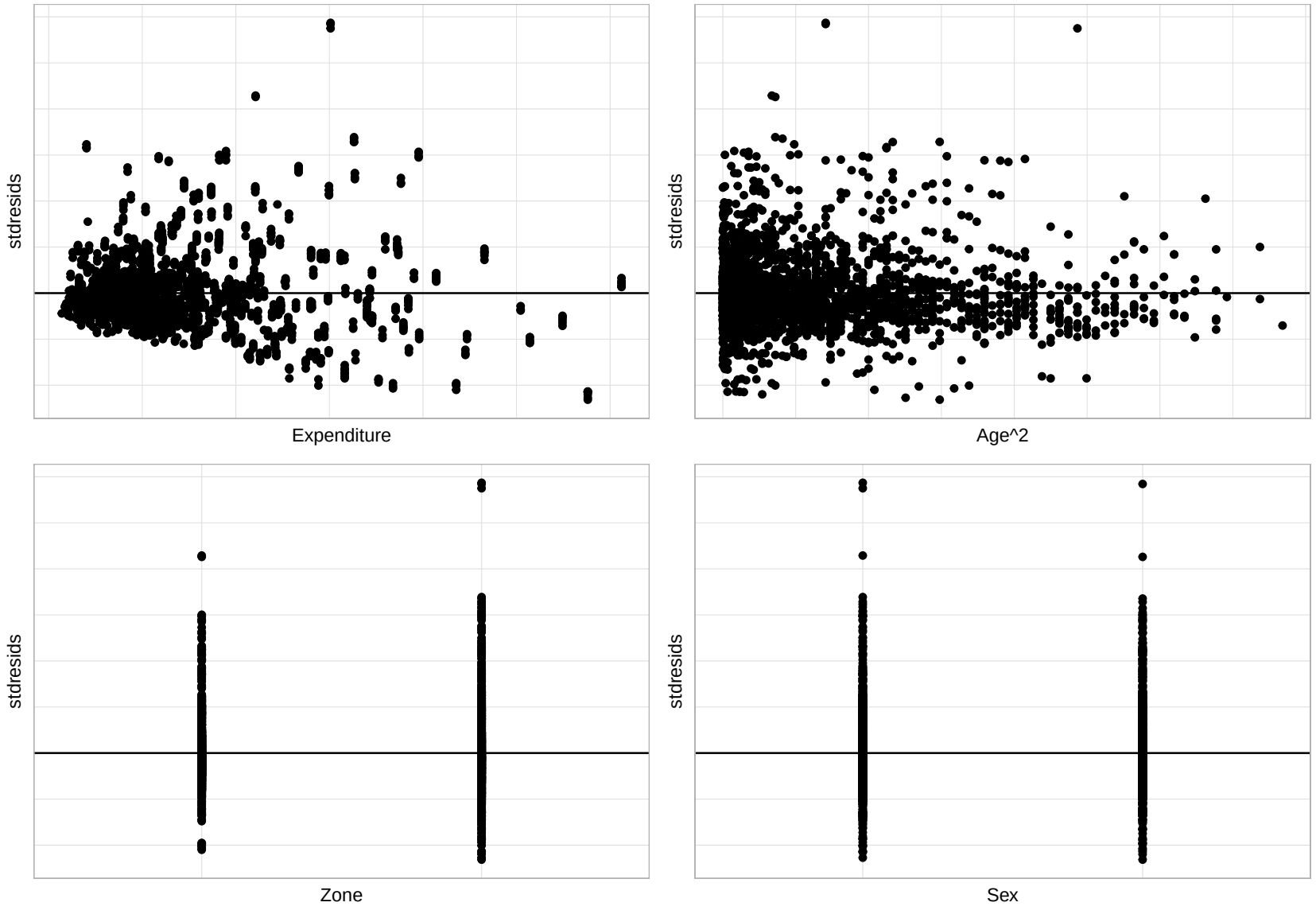
```
nortest::lillie.test(diseno_qwgt$variables$stdresids)
```

```
##
##  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$variables,
       aes(x = Age^2, y = stdresids))+
  geom_point() +
  geom_hline(yintercept = 0) + theme_cepal()
g4 <- ggplot(data = diseno_qwgt$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) +
```

Varianza constante



Distancia de cook

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

donde,

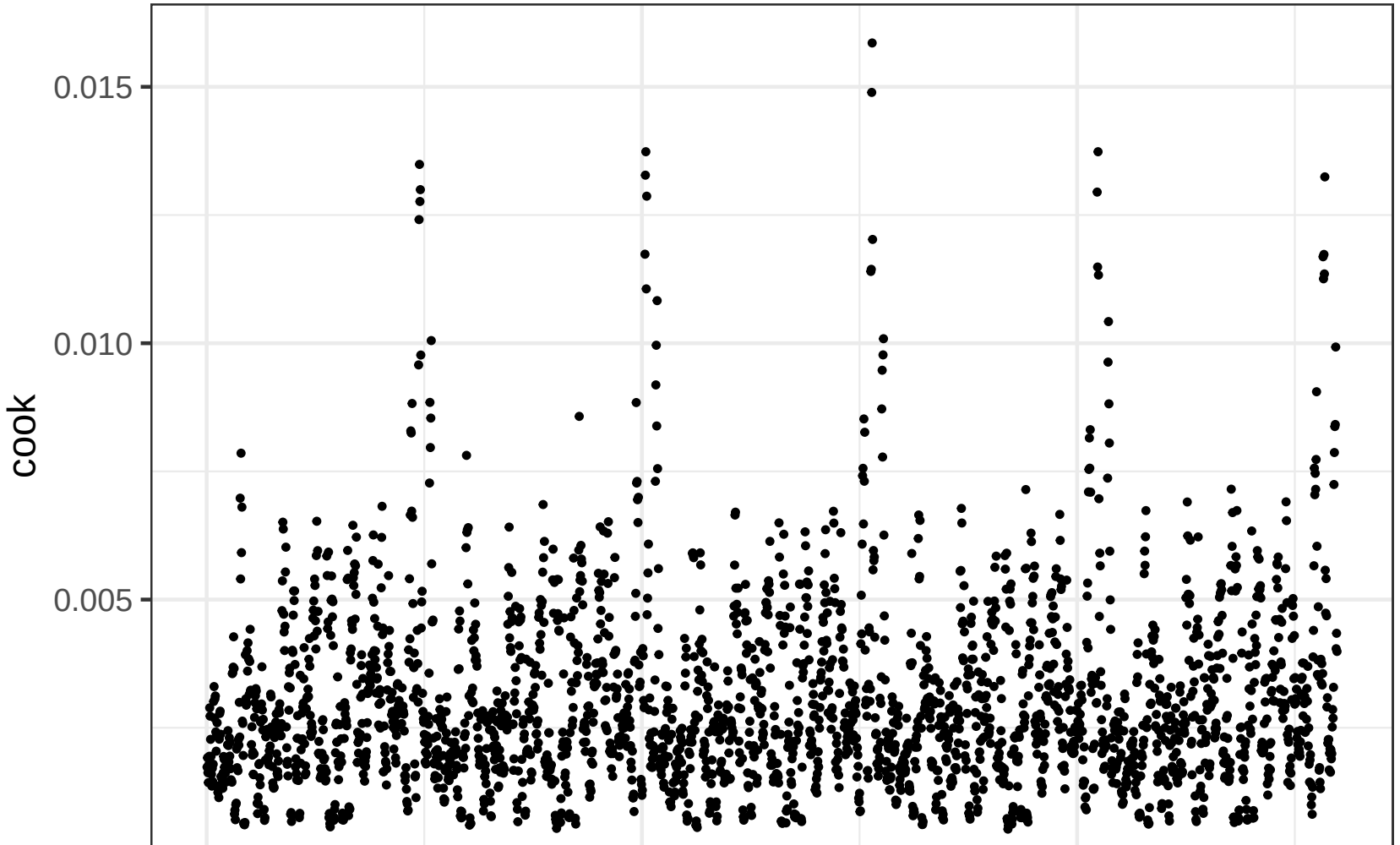
- w_i^* = Pesos de la encuesta.
- w_i Elementos por fuera de la diagonal de la matriz hat
- e_i = residuales
- p = número de parámetros del modelo de regresión.
- ϕ = parámetro de dispersión en el glm
- $\widehat{Var}\left(U_w\left(\hat{\boldsymbol{B}}_w\right)\right)$ = estimación de varianza linealizada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

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_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)
```



$D_f Betas_{(i)}$

$$D_f Beta_{(i)} = \hat{B} - \hat{B}_{(i)} = \frac{\left(\boldsymbol{X}^t \boldsymbol{X}\right)^{-1} \boldsymbol{X}_{(i)}^t \hat{e}_i}{1 - h_{ii}}$$

Donde $\hat{B}_{(i)}$ es el vector de parámetros estimados una vez se ha eliminado la i-ésima observación, h_{ii} es el correspondiende elemento de la matriz H y $\hat{e}_i$ es el residual de la i-ésima observación. La i-ésima observación es influyente para B_j si $\mid D_f Beta_{(i)j} \mid \geq \frac{2}{\sqrt{n}}$

Detección de observaciones influyentes (dfbetas)

```
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
-0.0003	-1e-04	-0.0011	0.0039	0.0019
0.0015	-1e-04	-0.0017	-0.0049	0.0014
0.0021	-3e-04	-0.0019	0.0063	-0.0021
-0.0012	6e-04	0.0014	-0.0056	-0.0002
-0.0026	5e-04	0.0014	0.0050	0.0004
-0.0024	4e-04	0.0009	0.0037	0.0016
-0.0023	4e-04	0.0008	0.0034	0.0017
-0.0029	8e-04	-0.0032	0.0095	0.0075
0.0027	1e-03	-0.0045	-0.0115	0.0027
0.0018	8e-04	-0.0042	0.0118	-0.0015

Detección de observaciones influyentes (dfbetas)

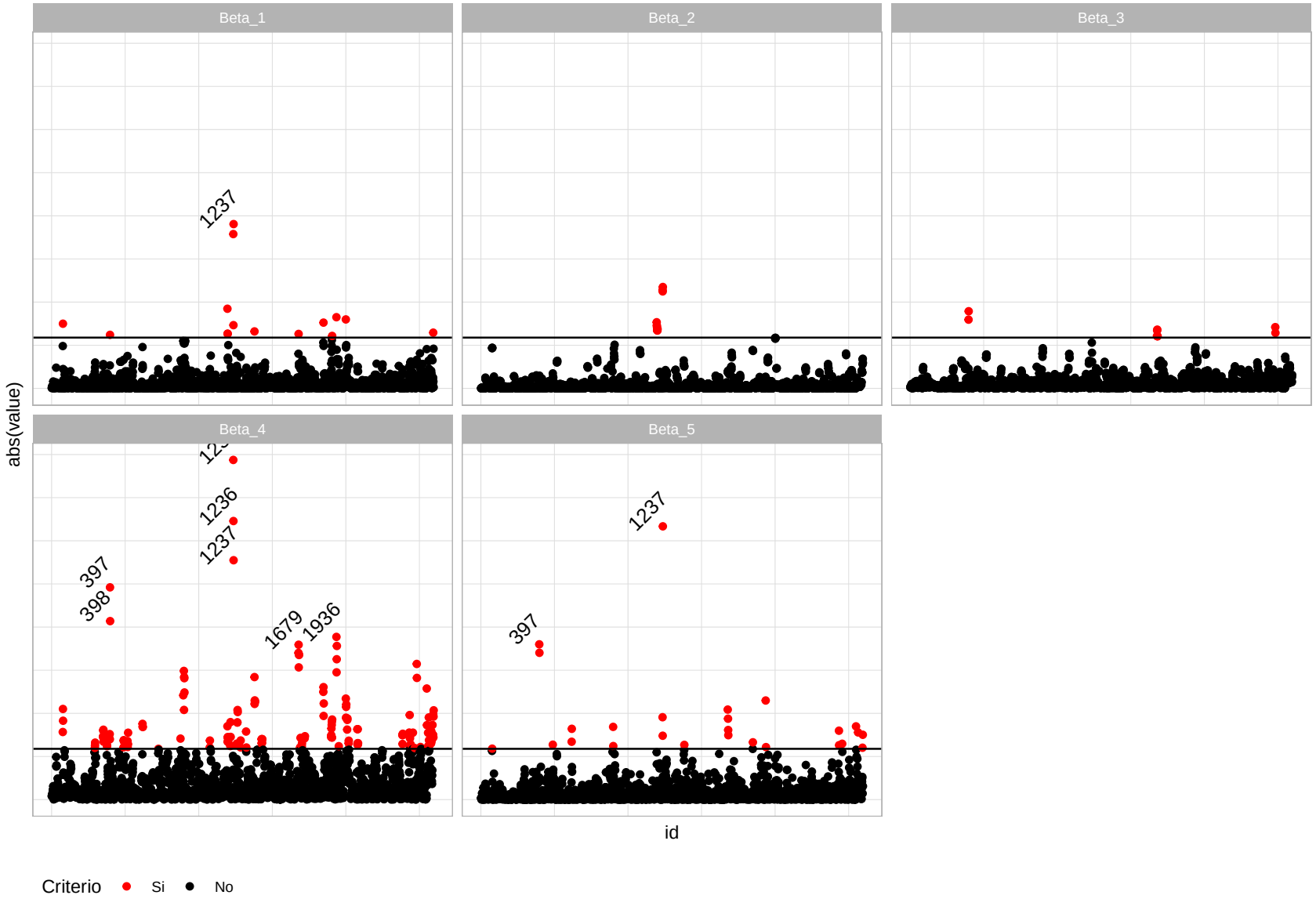
```
d_dfbetas$id <- 1:nrow(d_dfbetas)
d_dfbetas <- reshape2::melt(d_dfbetas,  id.vars = "id")
cutoff <- svydfbetas(mod_svy)$cutoff
d_dfbetas %<>%
  mutate(
    Criterio = ifelse(abs(value) > cutoff, "Si", "No"))

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

Detección de observaciones influyentes (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 (dfbetas)

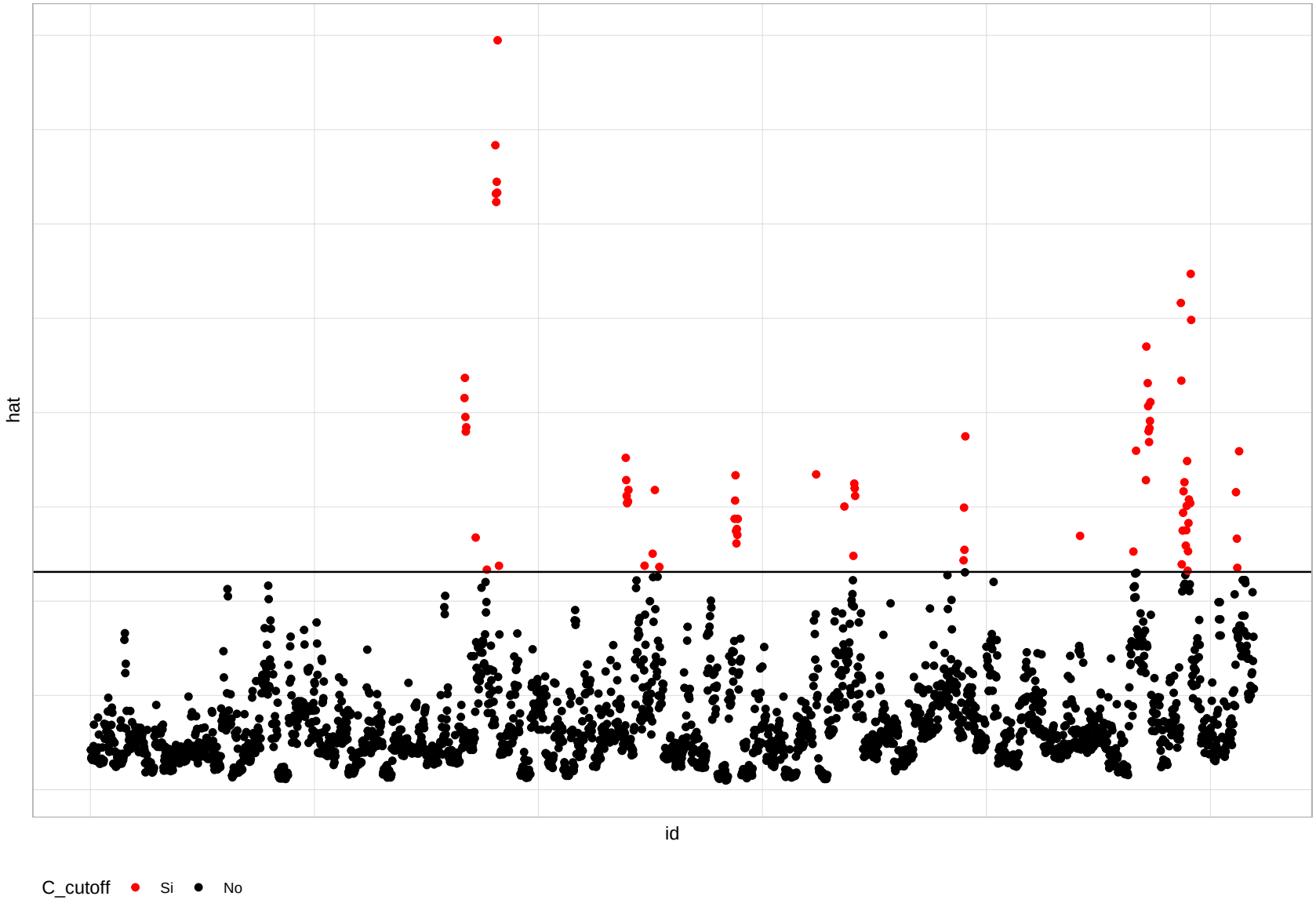


Detección de observaciones influyentes (valores hat)

```
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 (valores hat)



Estadístico DfFits

$$DfFits\,(i) = \frac{\left[\left(\hat{B} - \hat{B}_{(i)}\right)^{t} \left(\boldsymbol{X}^{t}\boldsymbol{X}\right) \left(\hat{B} - \hat{B}_{(i)}\right)\right]^{1/2}}{\hat{\sigma}_{(i)}}$$

La i-ésima observación se considera influyente en el ajuste del modelo si $\mid DfFits\,(i) \mid \geq 2\sqrt{\frac{p}{n}}$

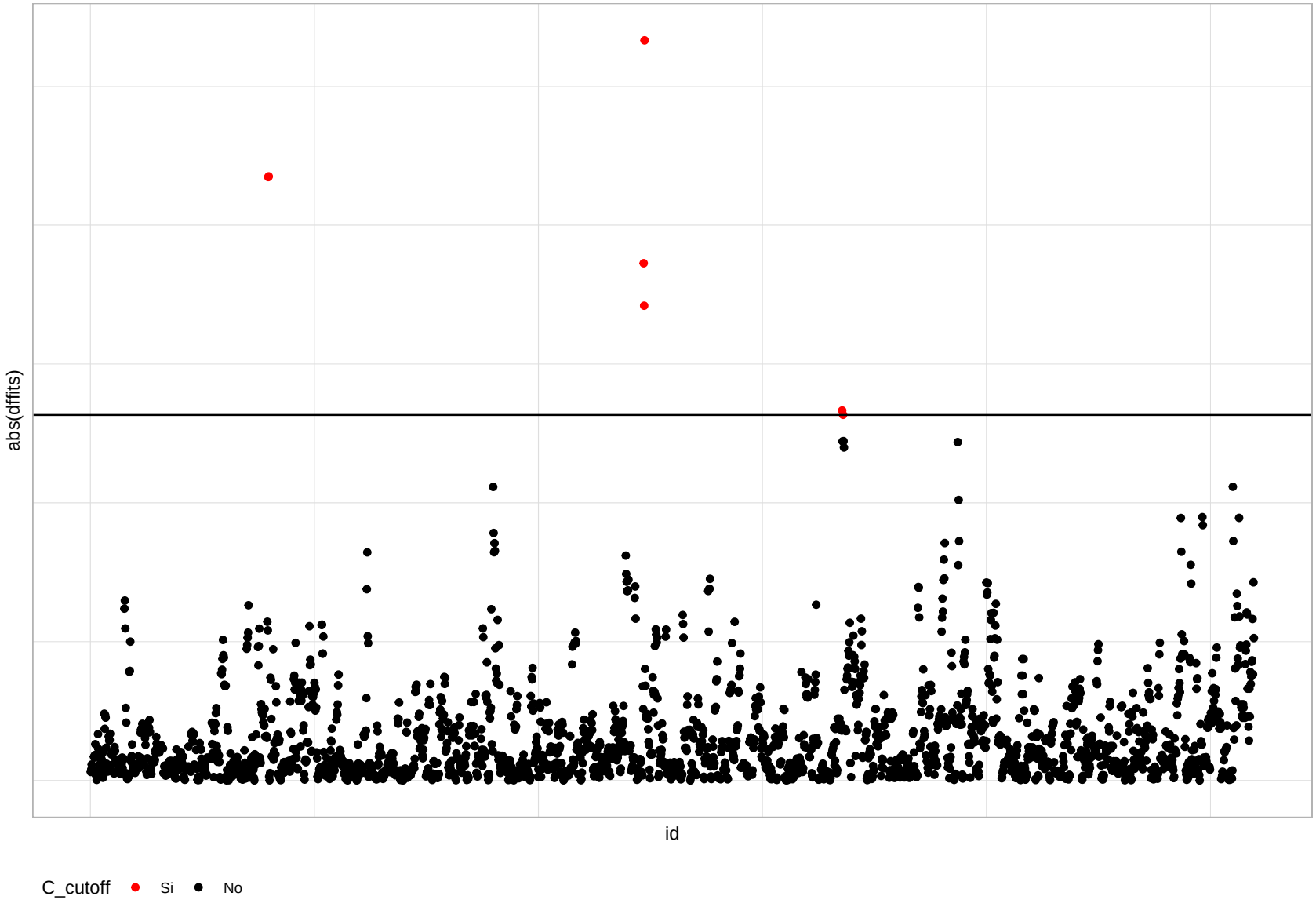
Detección de observaciones influyentes (dffit)

```
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 (dffit)



Inferencia sobre los parámetros del modelo

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

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

Estimación del dato

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

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

	Estimado
(Intercept)	71.518
Expenditure	1.090
ZoneUrban	47.645
SexMale	9.171

Estimación del dato

(Intercept)	Expenditure	ZoneUrban	SexMale	Age
1	247.9	0	1	55
1	247.9	0	0	46
1	247.9	0	1	13
1	164.6	0	1	31
1	164.6	0	0	28
1	164.6	0	0	9
1	164.6	0	0	5

$$\hat{y}_i = 71.518 + 1.090(247.9) + 47.645(0) + 9.171(1) + 1.169(55) = 415.2$$

Estimando el IC de predicción

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

```
vcov(mod_svy)
```

	(Intercept)	Expenditure	ZoneUrban	SexMale	A
(Intercept)	794.547	-1.6529	-90.2239	0.3300	-6.49
Expenditure	-1.653	0.0056	-0.3448	-0.0588	0.00
ZoneUrban	-90.224	-0.3448	559.9048	1.0237	-1.29
SexMale	0.330	-0.0588	1.0237	53.6730	0.27
Age	-6.498	0.0078	-1.2995	0.2795	0.16

```
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] 19.57
```

Intervalo de confianza para la predicción

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

Utilizando la función predict

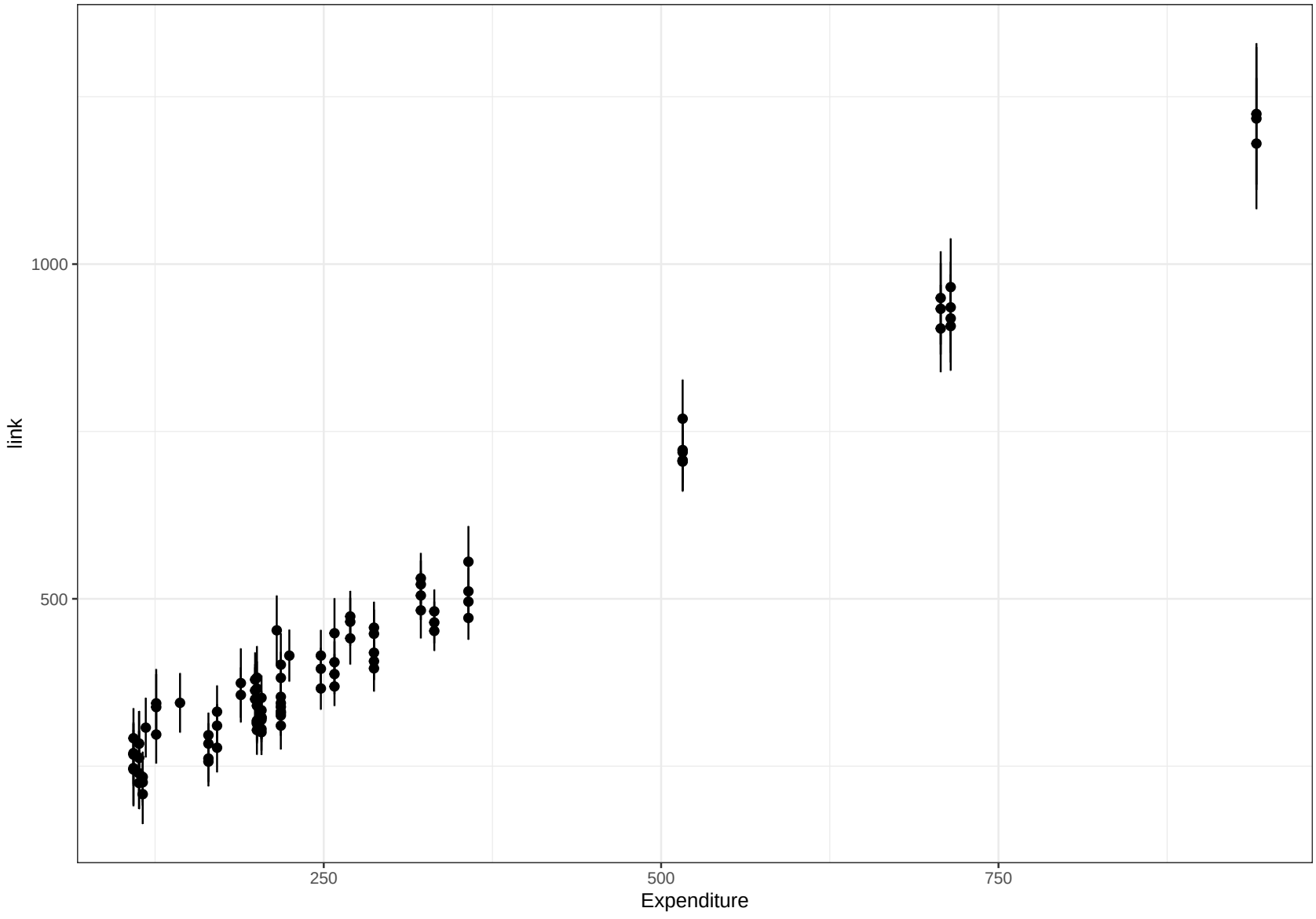
```
pred <- data.frame(predict(mod_svy, type="link"))
pred_IC <- data.frame(
  confint(predict(mod_svy, type="link")))
colnames(pred_IC) <- c("Lim_Inf", "Lim_Sup")
pred <- bind_cols(pred,pred_IC)
pred$Expenditure <- encuesta$Expenditure
pred %>% slice(1:6L)
```

link	SE	Lim_Inf	Lim_Sup	Expenditure
415.2	19.57	376.8	453.5	247.9
395.5	16.00	364.1	426.8	247.9
366.1	16.17	334.4	397.7	247.9
296.3	17.12	262.7	329.9	164.6
283.6	15.57	253.1	314.1	164.6
261.4	17.94	226.2	296.6	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



Predicción fuera de las observaciones.

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

$$\hat{y}_i = 71.518 + 1.090(1600) + 47.645(1) + 9.171(1) + 1.169(40) = 1919$$

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

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

```
## [1] 98.26
```

Intervalo de confianza para la predicción

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

Predicción fuera de las observaciones.

```
predict(mod_svy, newdata = datos_nuevos, type =  "link")
```

```
##   link   SE
## 1 1919 98.3
```

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

2.5 %	97.5 %
1726	2112