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

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

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

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

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

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

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

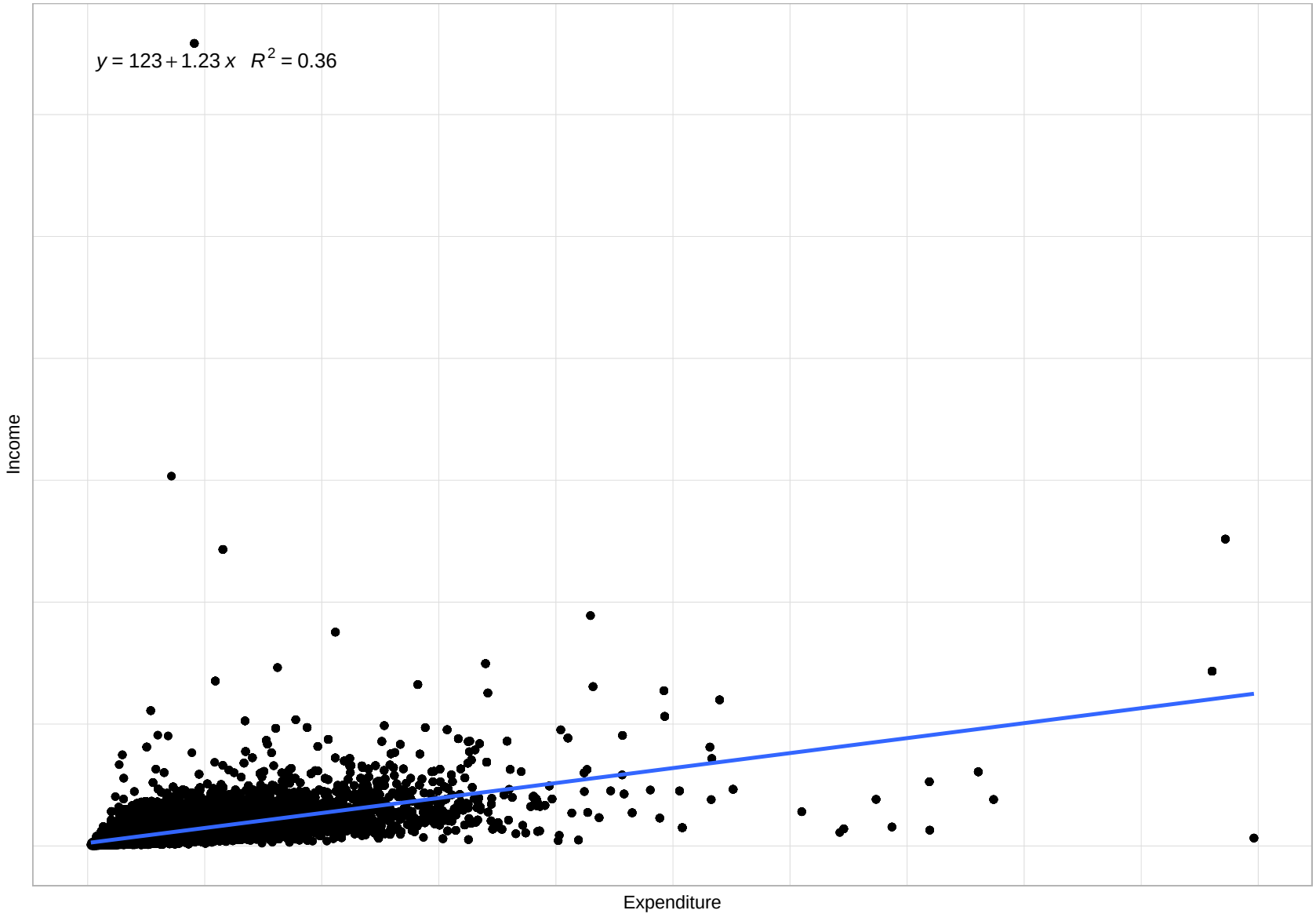
Modelo nulo (Q_W)

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

encuesta %<>% 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' )
```

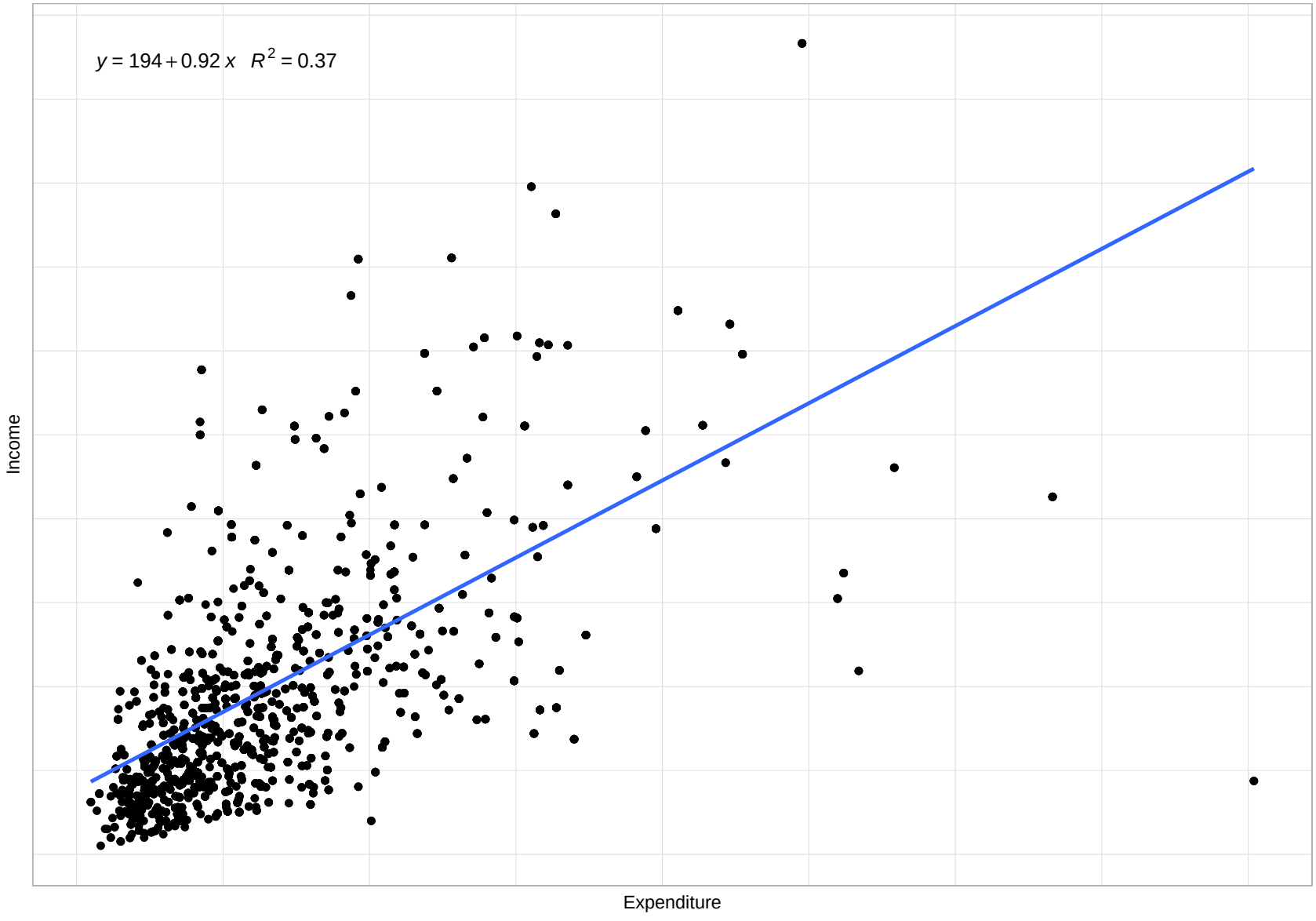
Table 1: Modelo BigCity

	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

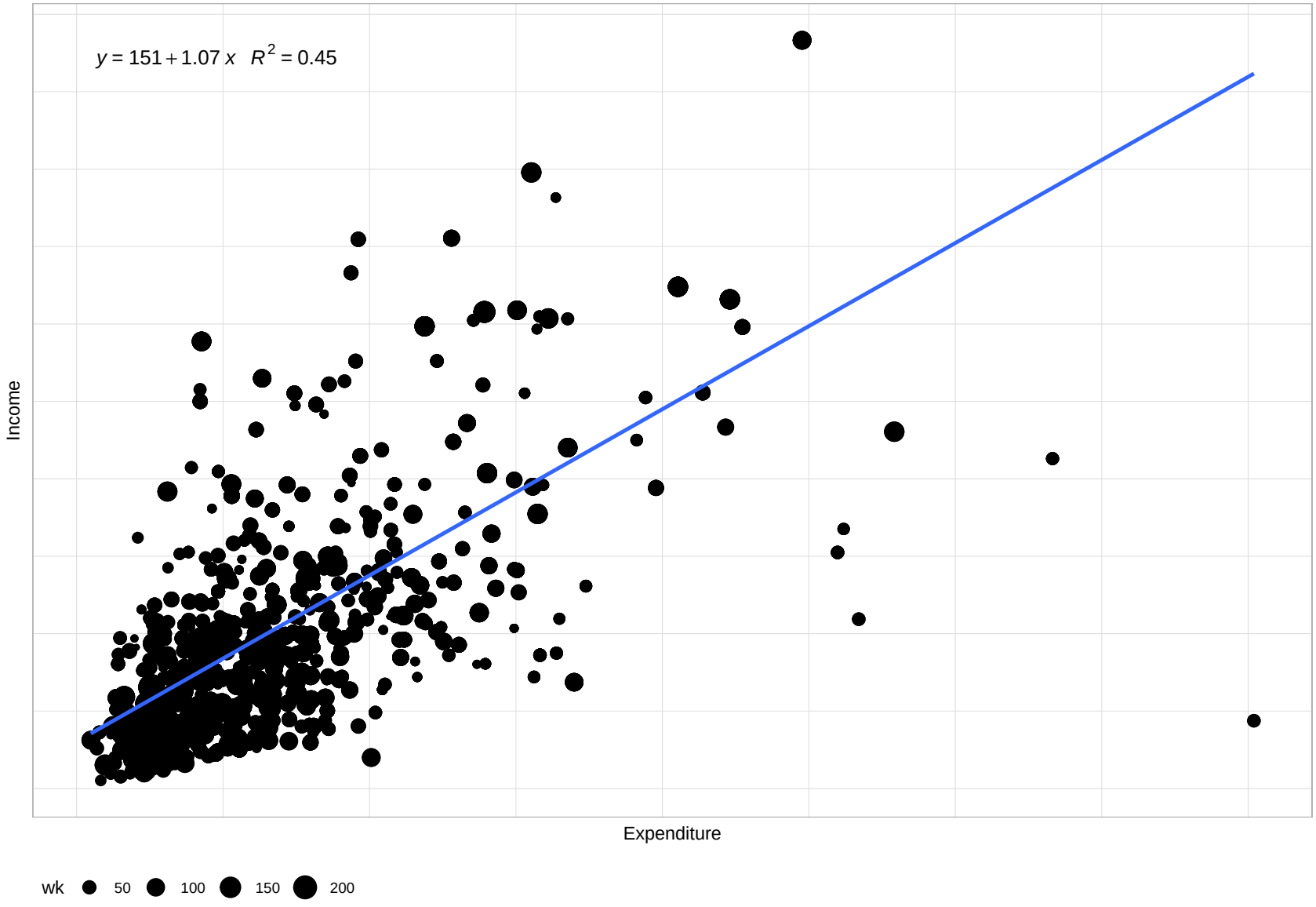
	Income
Expenditure	0.920^{***}
	(0.024)
Constant	194.200^{***}
	(10.420)
N	2422
R-squared	0.374
Adj. R-squared	0.374
Residual Std. Error	285.600 (df = 2420)
F Statistic	1447.000^{***} (df = 1; 2420)

***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	1.073***
	(0.024)
Constant	151.400***
	(10.580)
N	2422
R-squared	0.449
Adj. R-squared	0.449
Residual Std. Error	2289.000 (df = 2420)
F Statistic	1972.000*** (df = 1; 2420)

***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,]    84448 8472
```

```
s0$dispersion
```

```
##      variance    SE
## [1,]   153275 19642
```

```
(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

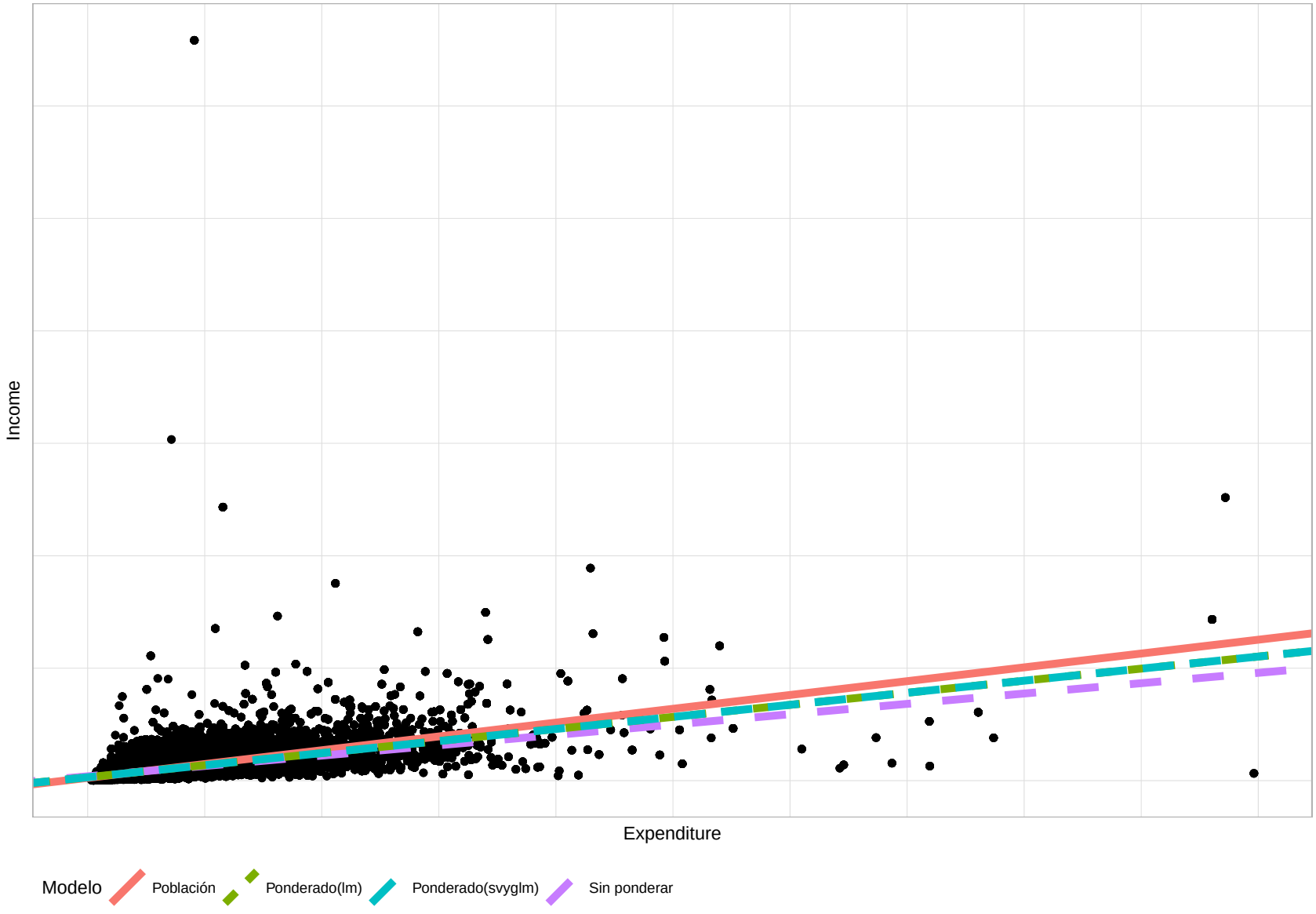
	Income
Expenditure	1.073***
	(0.099)
Constant	151.400***
	(33.010)
N	2422
AIC	34751.000

***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)	194.226	151.394	151.394
	p = (0.000)	p = (0.000)	p = (0.000)
Expenditure	0.920	1.073	1.073
	p = (0.000)	p = (0.000)	p = (0.000)
Num.Obs.	2422	2422	
R2	0.374	0.449	
R2 Adj.	0.374	0.449	
AIC	34268.1	34751.4	33.8
F	1446.794	1972.357	118.300
RMSE	285.59	2289.43	290.66

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)	149.248	32.2090	4.634	0
Expenditure	1.079	0.0963	11.203	0

Calculo del R^2

```
s1$dispersion
```

```
##      variance   SE
## [1,]    82893 8263
```

```
s0$dispersion
```

```
##      variance    SE
## [1,]   149888 18748
```

```
(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)	151.394	149.248
	p = (0.000)	p = (0.000)
Expenditure	1.073	1.079
	p = (0.000)	p = (0.000)
AIC	33.8	31.6
F	118.300	125.509
RMSE	290.66	287.97

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	1.053^{***} (0.105)
ZoneUrban	35.240 (28.830)
SexMale	2.030 (12.380)
Age2	-0.002 (0.004)
Constant	142.400^{***} (31.680)
N	2422
AIC	34704.000

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

Calculo del R^2

```
s1$dispersion
```

```
##      variance   SE
## [1,]    82616 8170
```

```
s0$dispersion
```

```
##      variance    SE
## [1,]   149888 18748
```

```
(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

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

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

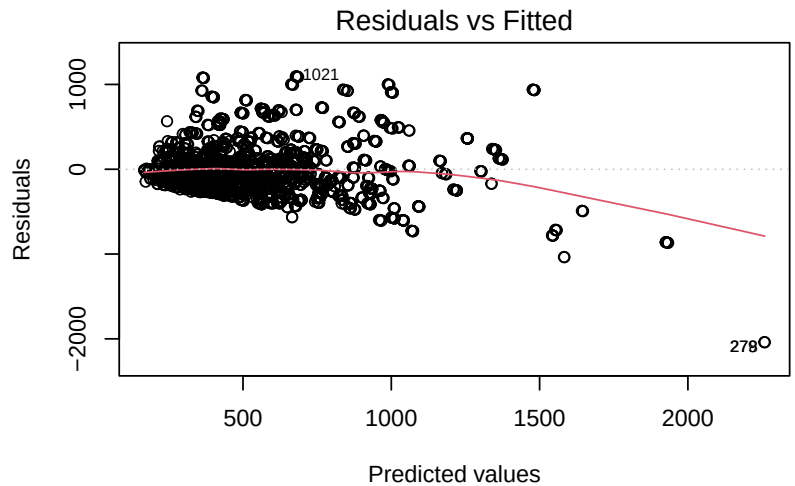
donde

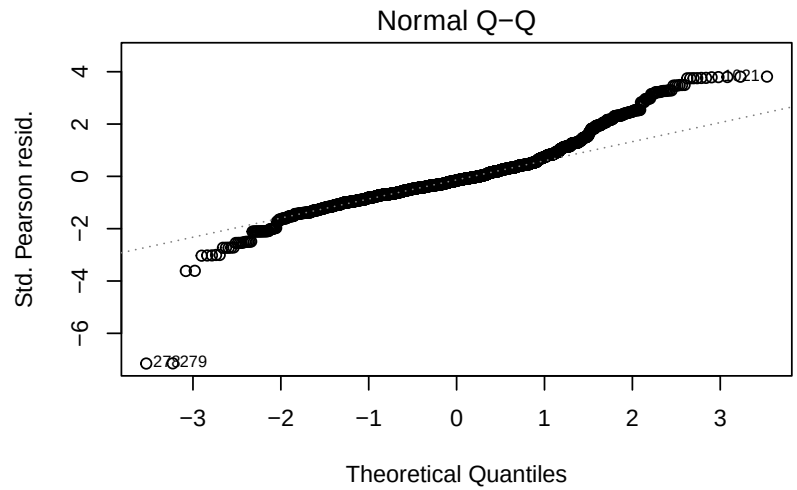
$$W = diag\left\{\frac{w_1}{V\left(\hat{\mu}_1\right)\left[g'\left(\mu_1\right)\right]^2}, \ldots, \frac{w_n}{V\left(\hat{\mu}_1\right)\left[g'\left(\mu_n\right)\right]^2}\right\}$$

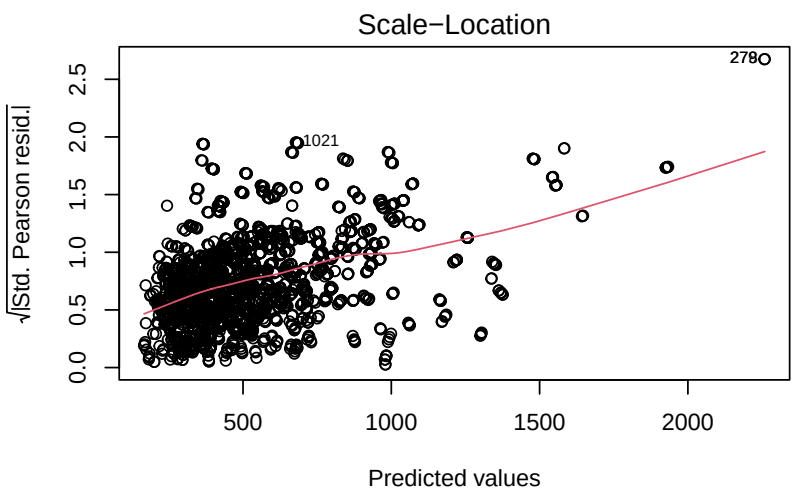
con g es una función de enlace que es especificada mediante un Modelo lineal generalizado.

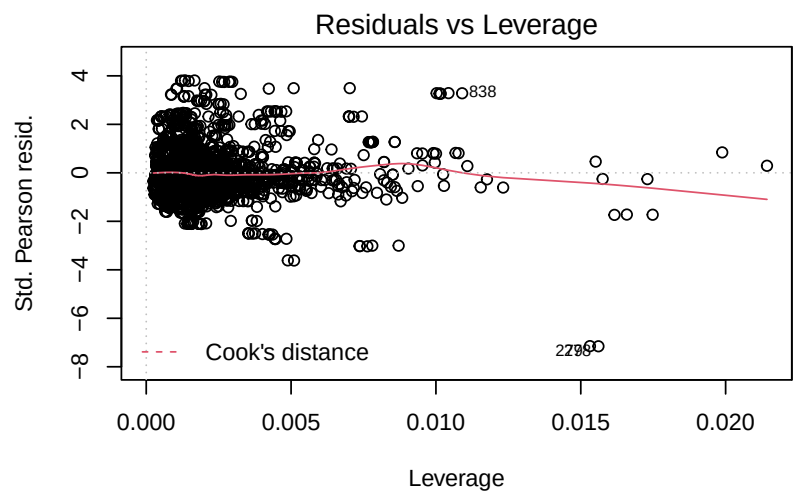
Diagnostico del Modelo

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

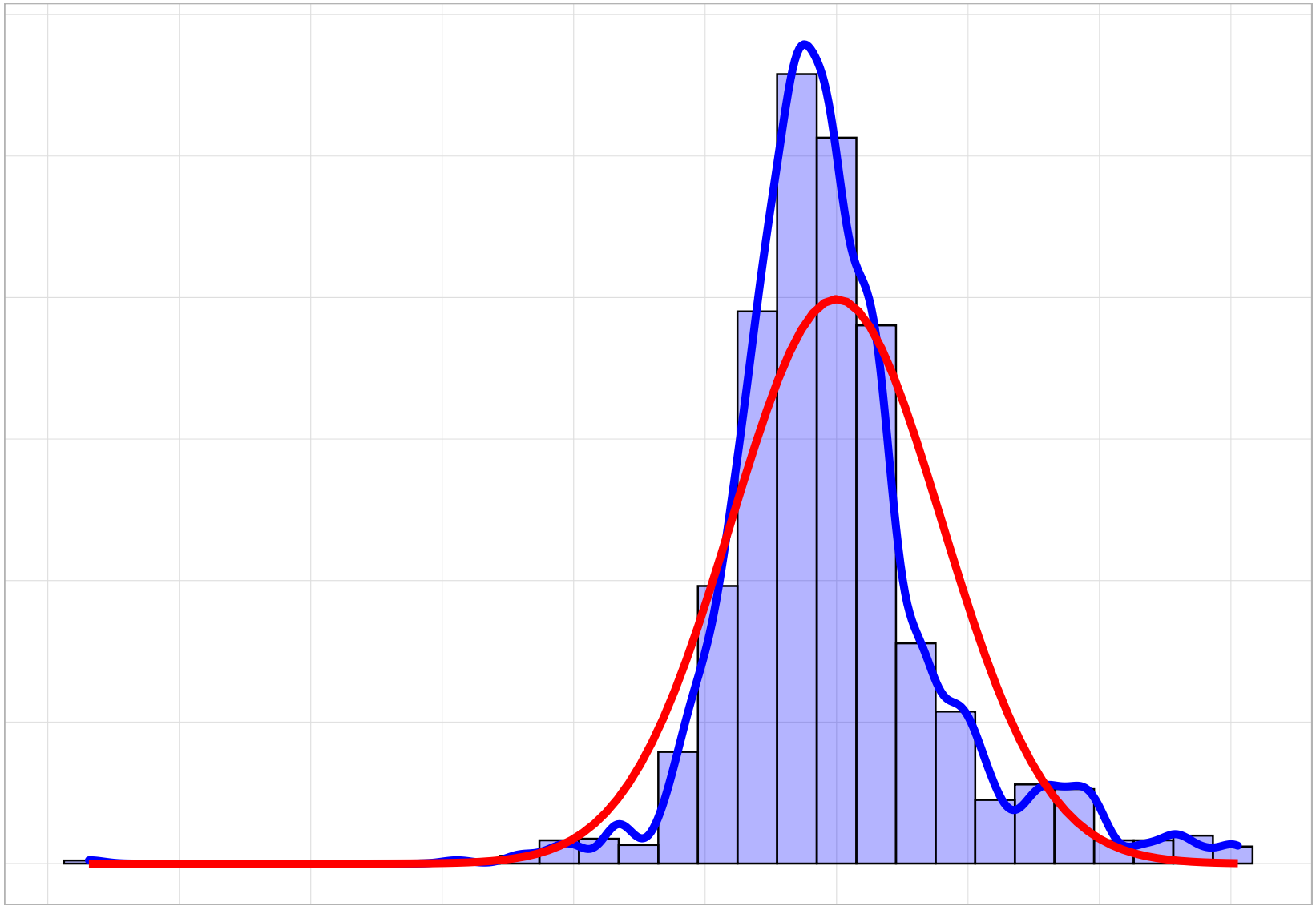
Extrayendo residuales standarizados

```
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..), 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 = 0.1, p-value <2e-16
```

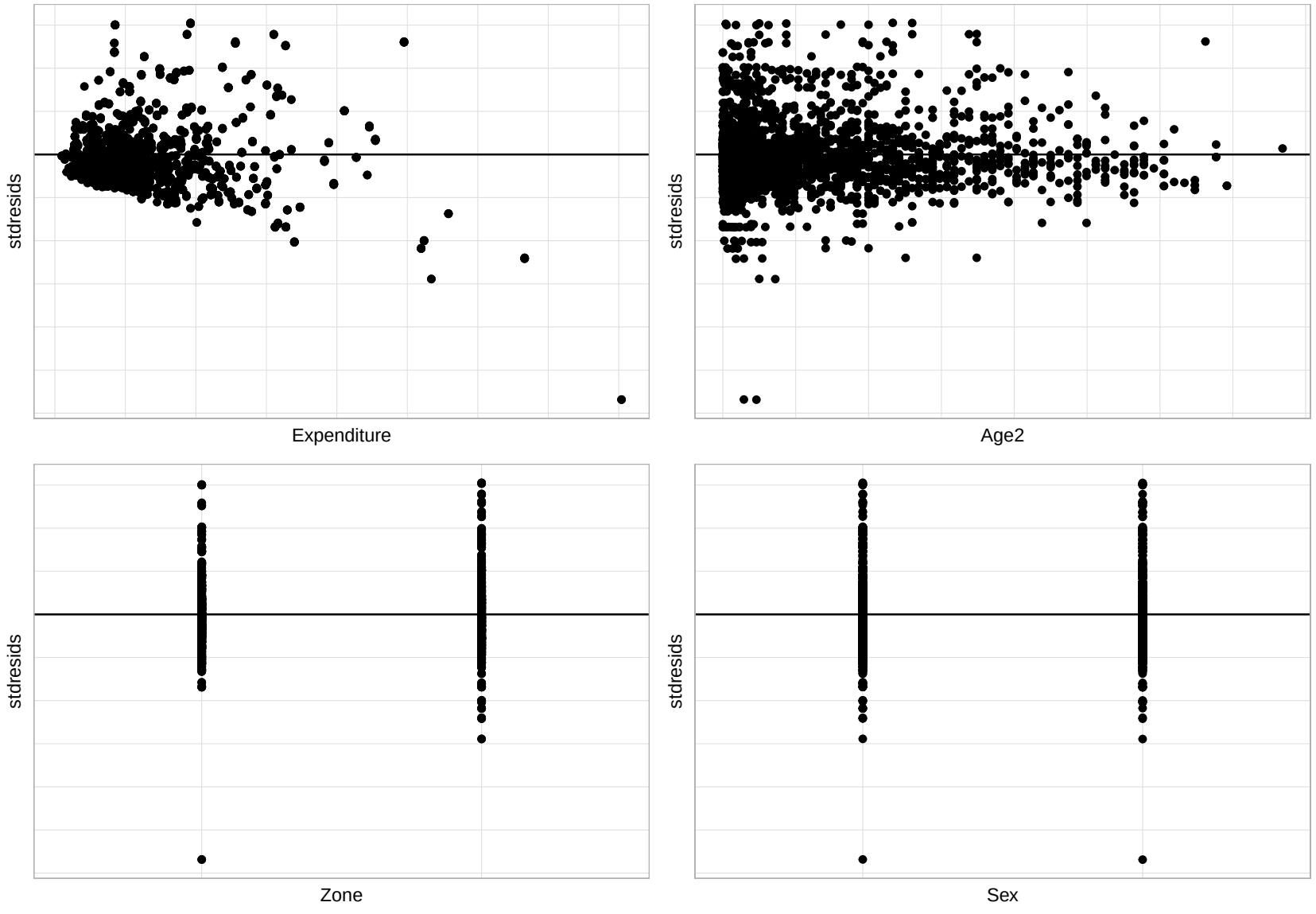
Varianza constante

```
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

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

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.
- $\phi =$ 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

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:

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

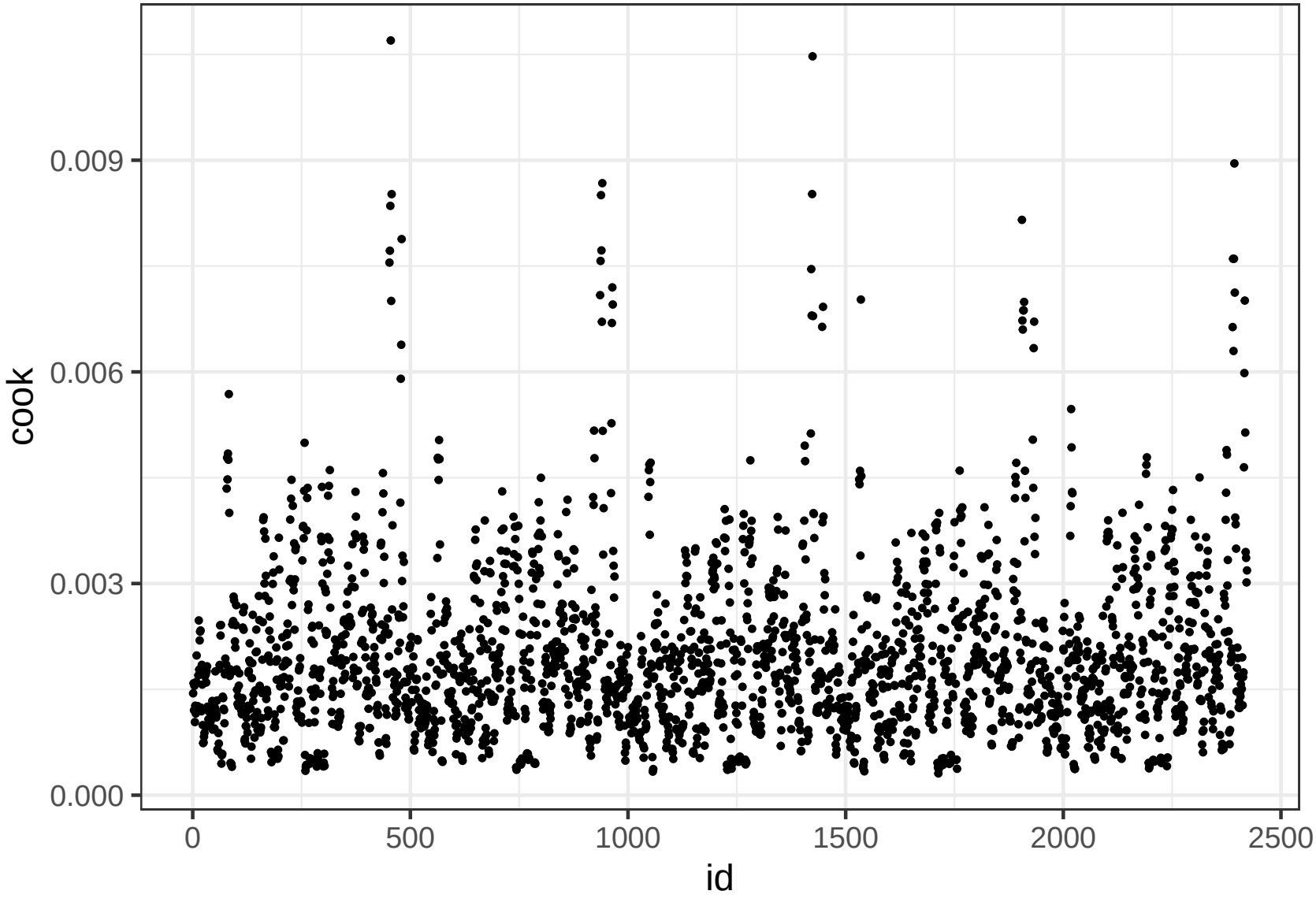
donde df = grados de liberta basados en el diseño.

Por otro lado, la literatura considera a las observaciones influyentes 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)



$Df\,Beta_{(i)}$

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

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

$D_f Betas_{(i)}$

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

donde:

- $c_{ji} =$ es el ji-estimo elemento de $\boldsymbol{A}^{-1} w_i^2 \boldsymbol{X}_{(i)} \boldsymbol{X}_{(i)}^t \boldsymbol{A}^{-1}$
- 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

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

```
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
-0.0037	0.0006	0.0019	0.0049	0.0013
-0.0042	0.0006	0.0018	0.0050	0.0035
-0.0023	0.0006	0.0019	-0.0054	0.0036
-0.0043	0.0007	0.0018	0.0050	0.0040
-0.0001	0.0001	0.0001	-0.0003	-0.0003
-0.0001	0.0000	0.0001	0.0002	-0.0002
-0.0003	0.0022	-0.0054	0.0107	0.0003
0.0037	0.0021	-0.0053	-0.0093	-0.0005
0.0049	0.0019	-0.0051	-0.0094	-0.0063
0.0049	0.0019	-0.0050	-0.0094	-0.0066

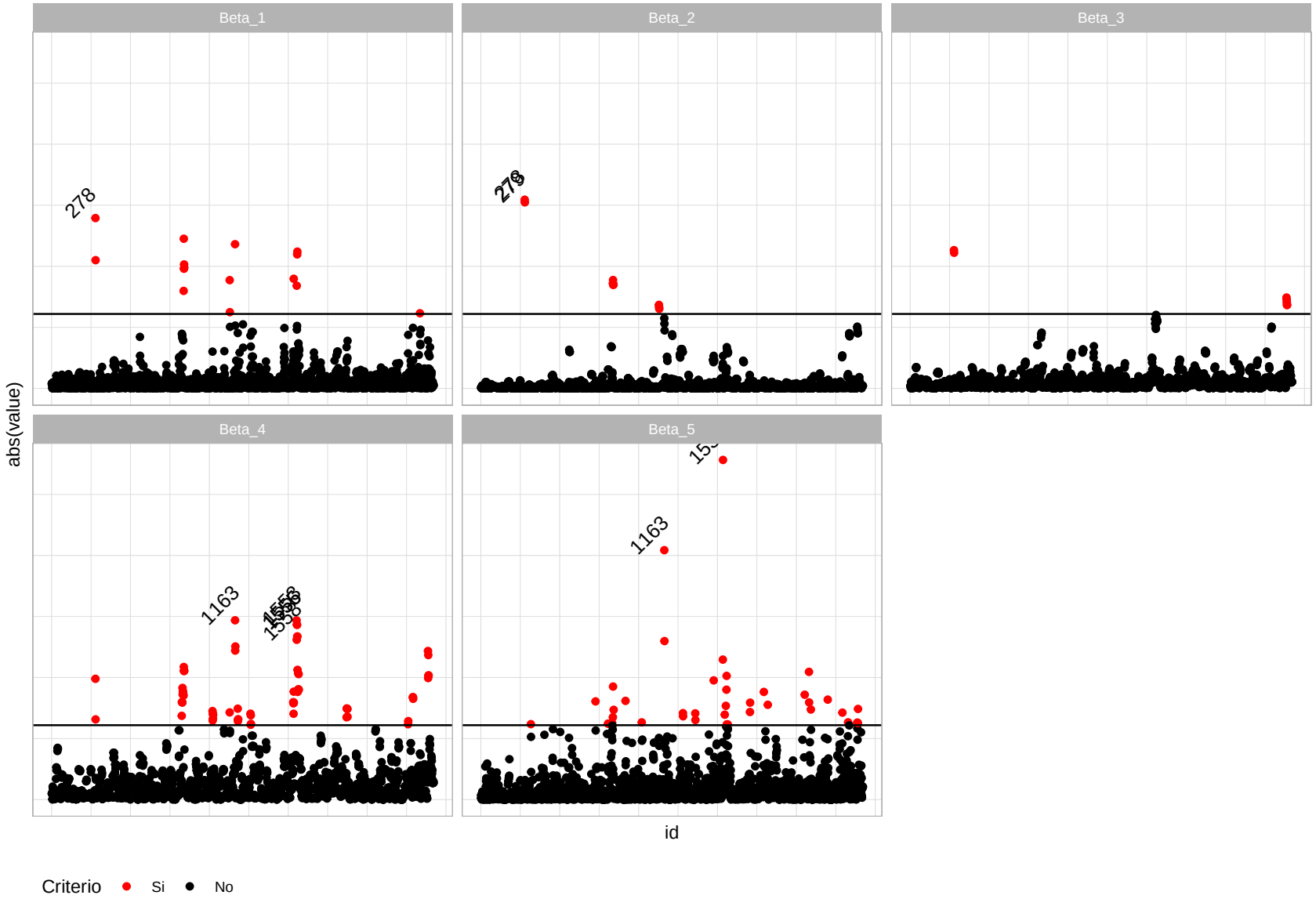
Detección de observaciones influyentes ($D_f Betas_{(i)j}$)

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



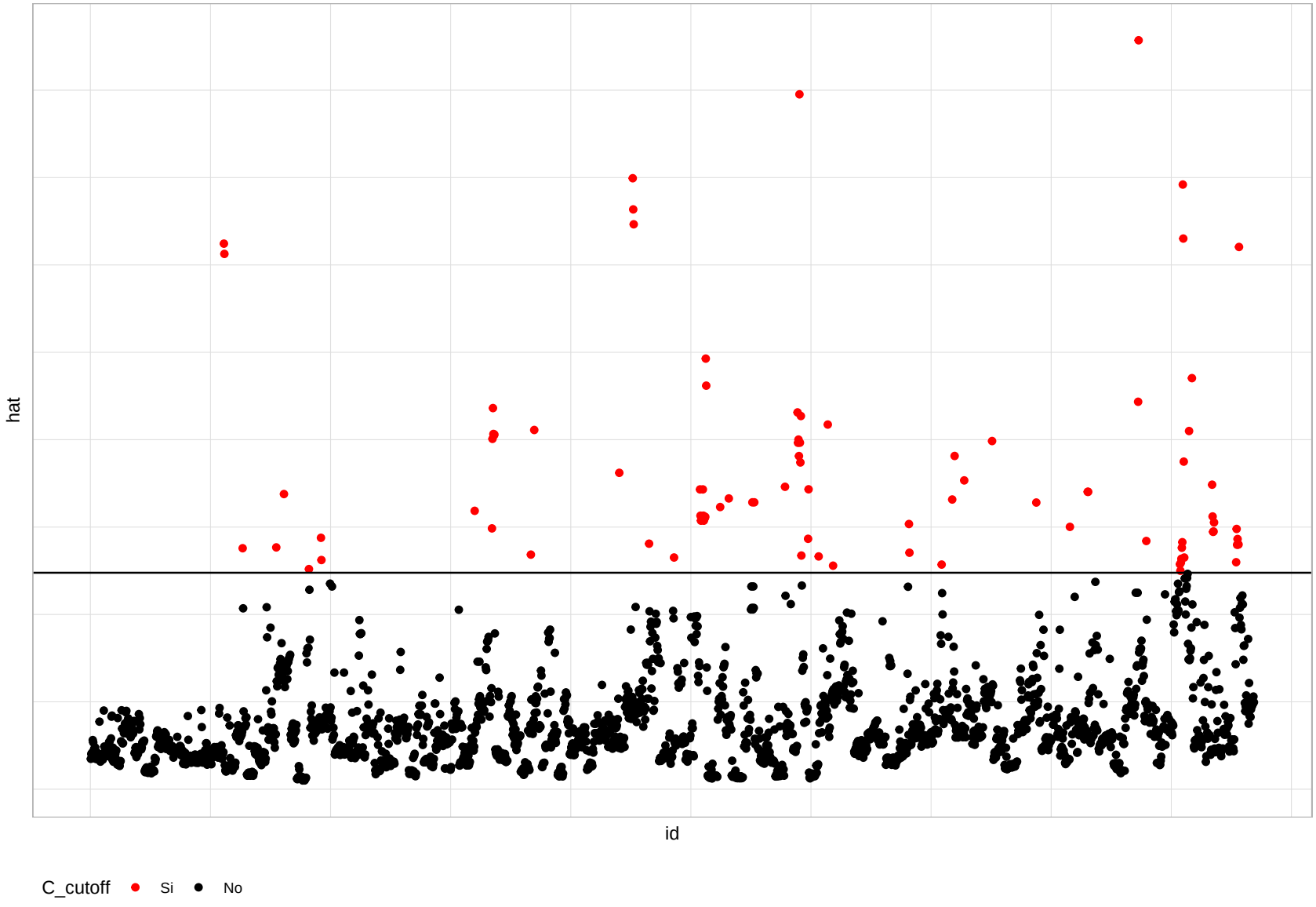
Matriz H asociada al PMLE

- 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 $h_{ii} = \boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{t}}\boldsymbol{A}^{-1}\boldsymbol{x}_{\boldsymbol{i}}{}^{-t}w_i$.
- Una observación puede ser grande y, como resultado, influir en las predicciones, cuando un x_i es considerablemente diferente del promedio ponderado $\bar{x}_w = \sum_{i\in s} w_i \boldsymbol{x}_{\boldsymbol{i}} / \sum_{i\in s} w_i$.

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})



Estadístico $D_f Fits_{(i)}$

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

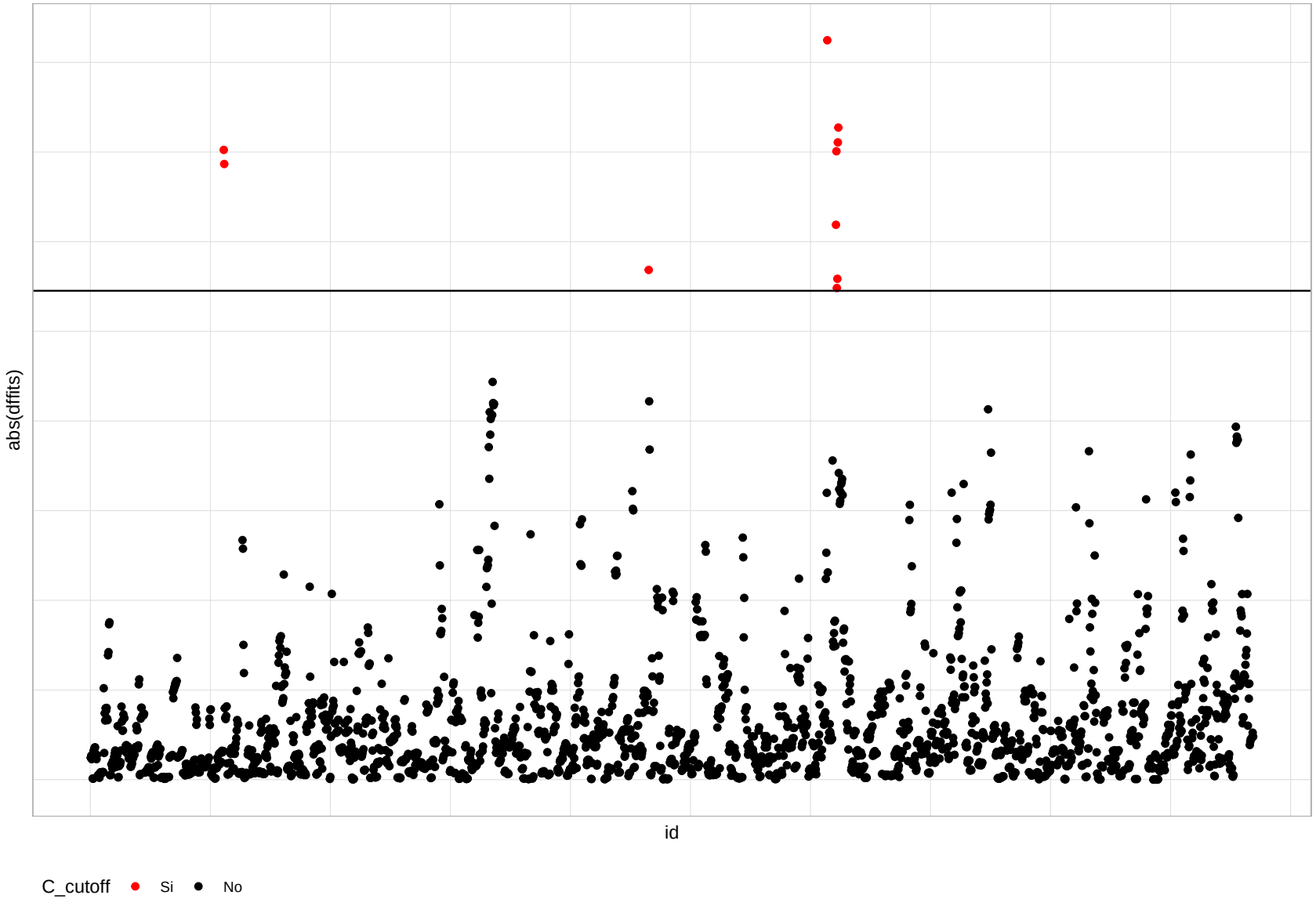
Donde, $\sqrt{v\left(\hat{\beta}_j\right)}$ puede ser aproximada por el diseño o el Modelo.

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

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 ($Df Fits_{(i)}$)



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

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

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

term	estimate	std.error	statistic	p.value
(Intercept)	142.368	31.6755	4.4946	0.0000
Expenditure	1.053	0.1047	10.0605	0.0000
ZoneUrban	35.239	28.8336	1.2221	0.2242
SexMale	2.030	12.3800	0.1640	0.8700
Age2	-0.002	0.0038	-0.5161	0.6068

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

Estimación del dato

(Intercept)	Expenditure	ZoneUrban	SexMale	Age2
1	207.2	0	0	1024
1	207.2	0	0	169
1	207.2	0	1	81
1	207.2	0	0	9
1	86.8	0	1	3364
1	86.8	0	0	3364
1	494.5	0	1	1296

$$\hat{y}_i = 91.6319 + 1.0893(247.9) + 48.7667(0) + 8.0933(1) + 0.0115(55^2) = 404.6$$

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	Age2
(Intercept)	1003.3374	-2.7810	61.3195	17.1429	-0.0469
Expenditure	-2.7810	0.0110	-1.3884	-0.1016	0.0001
ZoneUrban	61.3195	-1.3884	831.3739	-43.2900	0.0128
SexMale	17.1429	-0.1016	-43.2900	153.2635	0.0048
Age2	-0.0469	0.0001	0.0128	0.0048	0.0000

Estimando el IC de predicción

```
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] 16.4
```

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)
```

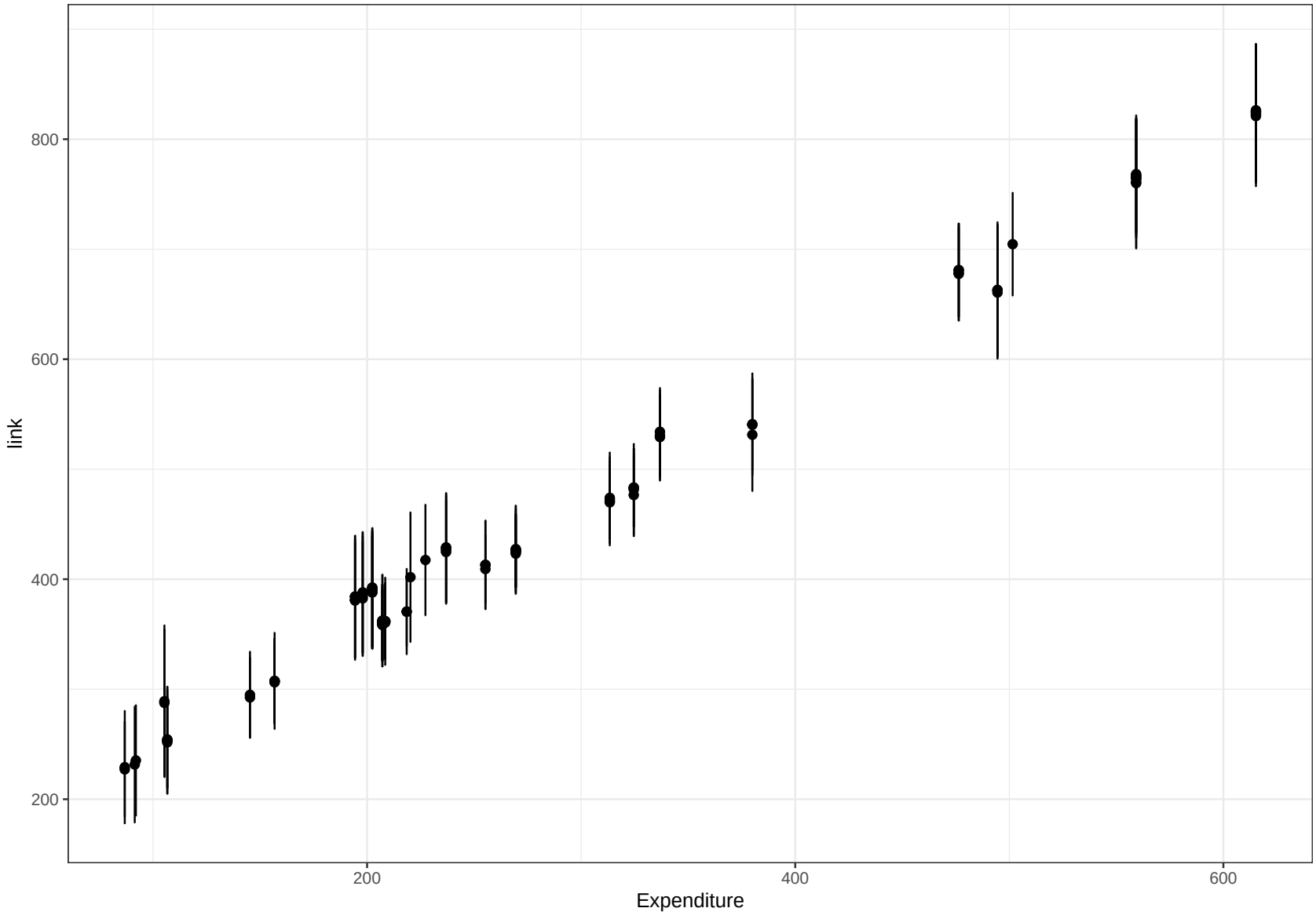
Utilizando la función predict

link	SE	Lim_Inf	Lim_Sup	Expenditure
358.5	16.40	326.3	390.6	207.2
360.1	17.62	325.6	394.7	207.2
362.3	21.50	320.2	404.5	207.2
360.5	17.91	325.4	395.6	207.2
229.1	26.36	177.5	280.8	86.8
227.1	22.20	183.6	270.6	86.8

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,
                           Age2 = 40^2, Sex = "Male",
                           Zone = "Urban")
```

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

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

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

```
## [1] 129.3
```

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 1861 129
```

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

2.5 %	97.5 %
1608	2115