Análisis de encuestas de hogares con R

Modulo 7: Modelos lineales generalizados (Variable categóricas)

CEPAL - Unidad de Estadísticas Sociales

Análisis de encuestas de hogares con R

Método de Pseudo máxima verosimilitud

Sea $\boldsymbol{y}_i$ el vector de observaciones los cuales provienen de los vectores aleatorios $\boldsymbol{Y}_i$ para $i \in U$. Suponga también que $\boldsymbol{Y_1}, \ldots, \boldsymbol{Y_N}$ son IID con función de densidad $f(\boldsymbol{y}, \theta)$. Si todos los elementos de la población finita U fueran conocidos la función de log-verosimilitud estaría dada por:

$$L_U(\theta) = \sum_{i \in U} \log\left[f\left(\boldsymbol{y}_i; \theta\right)\right]$$

y las ecuaciones de verosimilitud están dadas por:

$$\sum_{i \in U} \boldsymbol{u}_i(\theta) = \boldsymbol{0}$$

donde

$$\boldsymbol{u}_i(\theta) = \frac{\partial \log\left[f\left(\boldsymbol{y}_i; \theta\right)\right]}{\partial \theta}$$

Método de Pseudo máxima verosimilitud

Si se cumplen las condiciones de regularidad (Ver Pag 281 de Cox and Hinkley 1974[1]), es posible considerar a

$$\boldsymbol{T} = \sum_{i \in U} \boldsymbol{u}_i (\theta)$$

como un vector de totales. La estimación $\boldsymbol{T}$ se puede hacer mediante

$$\hat{\boldsymbol{T}} = \sum_{i \in U} w_i \boldsymbol{u}_i (\theta),$$

donde w_i son los pesos previamente definidos.

[1] Cox, D. R., & Hinkley, D. V. (1974). Theoretical Statistics Chapman and Hall, London. See Also.

Método de Pseudo máxima verosimilitud (Definición)

Un estimador de Máxima Pseudo Verosimilitud (MVP) $\hat{\theta}_{MPV}$ de θ_U será la solución de las ecuaciones de Pseudo-Verosimilitud dadas por

$$\hat{\boldsymbol{T}} = \sum_{i \in U} w_i \boldsymbol{u}_i (\theta) = 0,$$

Através de la Linealización de Taylor podemos obtener la varianza asintotica de $\hat{\theta}_{MPV}$ dada por:

$$V_p \left(\hat{\theta}_{MPV} \right) \approx \left[J (\theta_U) \right]^{-1} V_p \left[\sum_{i \in s} w_i \boldsymbol{u}_i (\theta_U) \right] \left[J (\theta_U) \right]^{-1}$$

$$\hat{V}_p \left(\hat{\theta}_{MPV} \right) = \left[\hat{J} \left(\hat{\theta}_{MPV} \right) \right]^{-1} \hat{V}_p \left[\sum_{i \in s} w_i \boldsymbol{u}_i \left(\hat{\theta}_{MPV} \right) \right] \left[\hat{J} \left(\hat{\theta}_{MPV} \right) \right]^{-1}$$

Método de Pseudo máxima verosimilitud (Definición)

Con

$$J(\theta_U) = \left.\frac{\partial T(\theta)}{\partial \theta}\right|_{\theta=\theta_U} = \sum_{i \in U} \left.\frac{\partial \boldsymbol{u}_i(\theta)}{\partial \theta}\right|_{\theta=\theta_U}$$

$$\hat{J}\left(\hat{\theta}_{MPV}\right) = \left.\frac{\partial \hat{T}(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}_{MPV}} = \sum_{i \in s} w_i \left.\frac{\partial \boldsymbol{u}_i(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}_{MPV}}$$

$\hat{V}_p\left[\sum_{i \in s} w_i \boldsymbol{u}_i(\theta_U)\right]$ es la matriz de varianza estimada y $\hat{V}_p\left[\sum_{i \in s} w_i \boldsymbol{u}_i(\theta_{MPV})\right]$ es un estimador consistente para la varianza.

Introducción al GLM

Un modelo lineal generalizado tiene tres componentes básicos:

- **Componente aleatoria**: Identifica la variable respuesta $(y_1, \ldots, y_N)$ y su distribución de probabilidad.

Introducción al GLM

Un modelo lineal generalizado tiene tres componentes básicos:

- **Componente aleatoria**: Identifica la variable respuesta $(y_1, \ldots, y_N)$ y su distribución de probabilidad.
- **Componente sistemática**: Especifica las variables explicativas (independientes o predictoras) utilizadas en la función predictora lineal.
  Las covariables $x_1, \ldots, x_k$ producen un predictor lineal η_i que resulta de la combinación lineal $\eta_i = \sum_{j=1}^{k} x_{ij}\beta_j$ donde x_{ij} es el valor del j-ésimo predictor en el i-ésimo individuo, e $i = 1, \ldots, N$.

Introducción al GLM

- **Función link**: Es una función del valor esperado de Y , $E(Y)$, como una combinación lineal de las variables predictoras.
  Se denota el valor esperado Y como $\mu = E(Y)$, entonces la función *link* especifica una función

$$g(\mu) = \sum_{j=1}^{k} x_{ij}\beta_j.$$

  Así, la función $g(\cdot)$ realciona las componentes aleatoria y sistemática. De este modo, para $i = 1, \ldots, N$

$$\mu_i = E(Y_i)$$

$$\eta_i = g(\mu_i) = \sum_j \beta_j x_{ij}$$

Introducción al GLM

- Todos los modelos se pueden incluir dentro de la llamada familia exponencial de distribuciones

$$f\left(y_i \mid \theta_i\right) = a\left(\theta_i\right) b\left(\theta_i\right) \exp\left[y_i Q\left(\theta_i\right)\right]$$

de modo que $Q\left(\theta\right)$ recibe el nombre de *parámetro natural*. Además, $a(\cdot)$ y $b(\cdot)$ son funciones conocidas.

Introducción al GLM

- Todos los modelos se pueden incluir dentro de la llamada familia exponencial de distribuciones

$$f\left(y_i \mid \theta_i\right) = a\left(\theta_i\right) b\left(\theta_i\right) \exp\left[y_i Q\left(\theta_i\right)\right]$$

  de modo que $Q\left(\theta\right)$ recibe el nombre de *parámetro natural*. Además, $a(\cdot)$ y $b(\cdot)$ son funciones conocidas.
- Los modelos de regresión lineal típicos para respuestas continuas son un caso particular de los *GLM*.

Lectura de la base y definición del diseño muestral


```
library(survey)
library(srvyr)
encuesta <- readRDS("../Data/encuesta.rds")
diseno <- encuesta %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = T
  )
```

Definir nueva variable

Creando nuevas variables en la base de datos.

```
diseno <- diseno %>%
  mutate(
  pobreza = ifelse(Poverty != "NotPoor", 1, 0),
  desempleo = ifelse(Employment == "Unemployed", 1, 0
```

Modelo para el ingreso

Estimador de momentos de la distribución gamma

```
library(ggplot2)

x <- encuesta$Income
n = length(x)
shape1 = (n*mean(x)^2)/sum((x-mean(x))^2)
rate1 = (n*mean(x))/sum((x-mean(x))^2)
c(shape1 = shape1, rate1 = rate1)
```

```
##   shape1    rate1
## 1.442897 0.002494
```

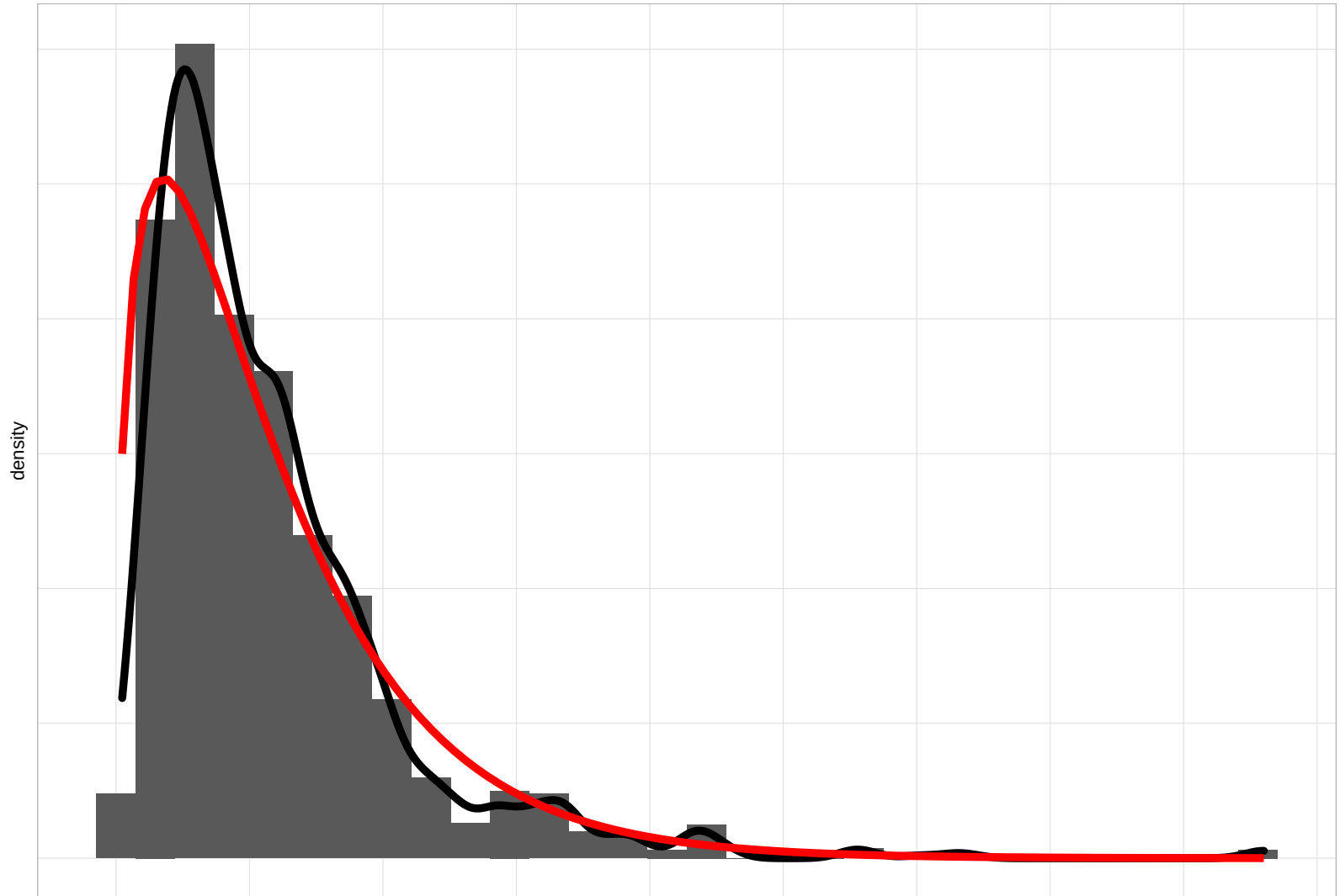
Modelo para el ingreso

La densidad empirica para el ingreso.

```
ggplot(data = encuesta, aes(x = Income) ) +
  geom_histogram(aes(y =..density..), bins = 30) +
  geom_density(aes(y =..density..), size = 2)+
  geom_function(fun = dgamma,
  args = list(shape = shape1, rate = rate1),
  col = "red", size = 2)  +
  theme_cepal()
```

Modelo para el ingreso

La linea roja se obtiene con la estimación de los parámetros, la linea negra es la densidad empírica.



Modelo gamma para variable continua.

La función de enlace $g(\cdot)$ para el GLM con una variable dependiente distribuida por Gamma es el recíproco, $\frac{1}{\mu_i}$. Eso significa que el valor esperado de y_i observado, $(E(y_i) = \mu_i)$, está relacionado con sus variables de entrada como, por ejemplo,

$$\frac{1}{\mu_i} = B_0 + B_1 x_1$$

o

$$\mu_i = \frac{1}{B_0 + B_1 x_1}$$

Modelo gamma ingreso


```
mod_qw <- lm(wk ~ Age + Sex + Region + Zone,
             data = encuesta)

encuesta$wk2 <-   encuesta$wk/predict(mod_qw)
diseno <- encuesta %>%
  as_survey_design( strata = Stratum,
    ids = PSU, weights = wk2,
    nest = T)

modelo <- svyglm(formula = Income ~ Age + Sex +
                   Region + Zone,
                   design = diseno,
                  family = Gamma(link = "inverse"))
broom::tidy(modelo)
```

Modelo gamma

Estimación de los parámetro del modelo.

term	estimate	std.error	statistic	p.value
(Intercept)	0.0024	2e-04	10.9726	0.0000
Age	0.0000	0e+00	-1.2838	0.2019
SexMale	-0.0001	0e+00	-1.8423	0.0681
RegionSur	-0.0001	2e-04	-0.2316	0.8173
RegionCentro	0.0000	2e-04	0.1383	0.8903
RegionOccidente	0.0002	2e-04	1.0196	0.3101
RegionOriente	0.0000	3e-04	0.0319	0.9746
ZoneUrban	-0.0009	2e-04	-4.8762	0.0000

Modelo gamma

Es útil la estimación de la dispersión que ofrece *svyglm* de forma predeterminada dado que no tiene en cuenta la información especial sobre la dispersión que se puede calcular utilizando la distribución Gamma. **No todos los GLM tienen una forma mejorada y específica del modelo para estimar**.

```
(alpha = MASS::gamma.dispersion(modelo))
```

```
## [1] 0.4831
```

```
mod_s <- summary(modelo, dispersion = alpha)
mod_s$dispersion
```

```
##      variance   SE
## [1,]    0.591 0.09
```

Modelo Gamma


```
mod_s$coefficients
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0024	2e-04	10.9726	0.0000
Age	0.0000	0e+00	-1.2838	0.2019
SexMale	-0.0001	0e+00	-1.8423	0.0681
RegionSur	-0.0001	2e-04	-0.2316	0.8173
RegionCentro	0.0000	2e-04	0.1383	0.8903
RegionOccidente	0.0002	2e-04	1.0196	0.3101
RegionOriente	0.0000	3e-04	0.0319	0.9746
ZoneUrban	-0.0009	2e-04	-4.8762	0.0000

Utilizando la función predict

Estimando los intervalos de confianza para la predicción.

```
pred <- data.frame(
  predict(modelo, type = "response", se = T))
pred_IC <- data.frame(
  confint(predict(modelo, type = "response", se = T))
colnames(pred_IC) <- c("Lim_Inf", "Lim_Sup")
pred <- bind_cols(pred, pred_IC)
pred$Income <- encuesta$Income
pred$Age <- encuesta$Age
pred %>% slice(1:6L)
```

Utilizando la función predict

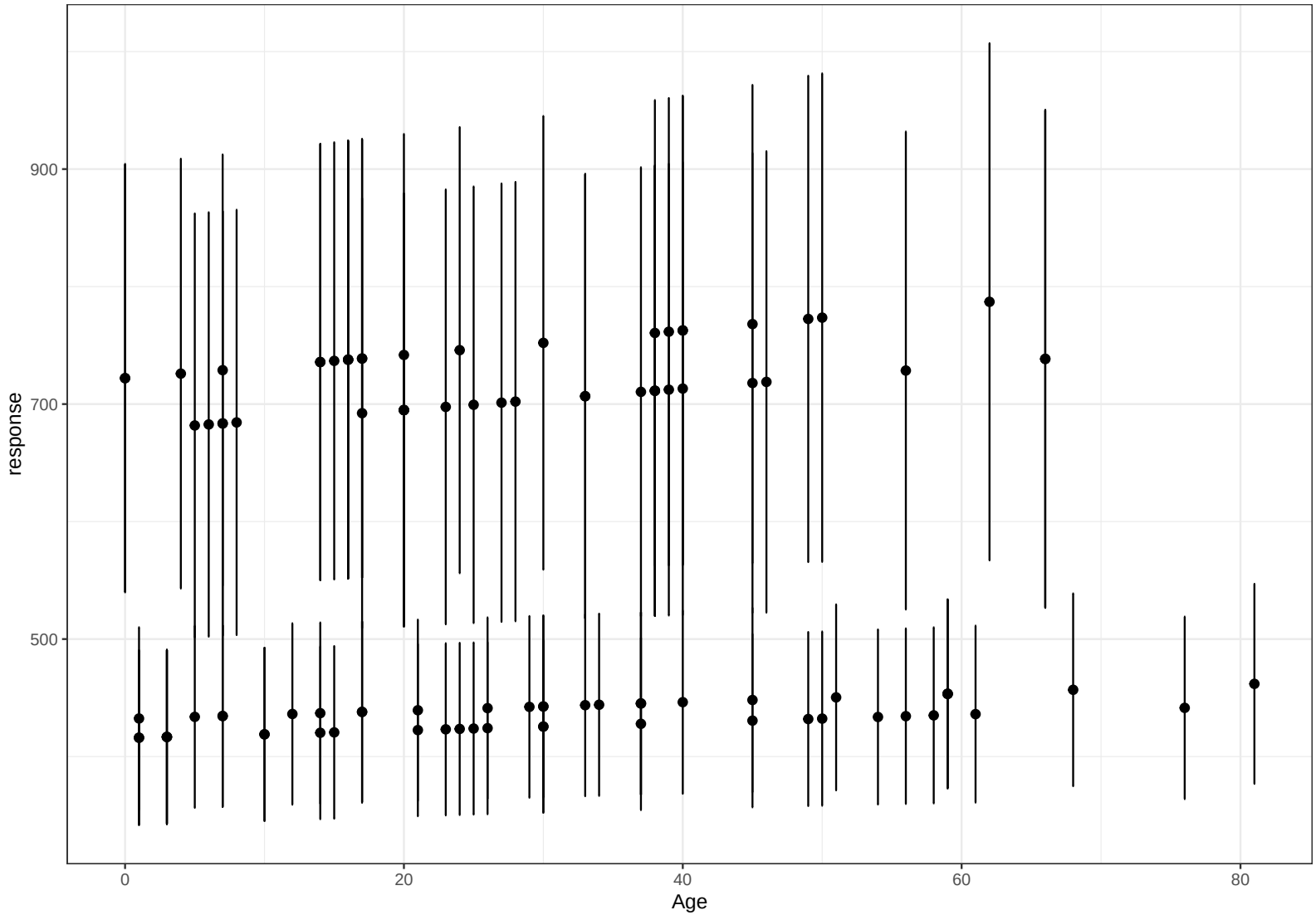
response	SE	Lim_Inf	Lim_Sup	Income	Age
456.8	41.80	374.9	538.8	409.9	68
434.4	38.07	359.8	509.0	409.9	56
423.5	37.33	350.3	496.7	409.9	24
441.2	39.35	364.1	518.3	409.9	26
416.7	37.78	342.6	490.7	409.9	3
436.1	38.36	360.9	511.3	823.8	61

Scaterplot de la predicción

Intervalos de confizan para la predicción en cada punto.

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

Utilizando la función predict

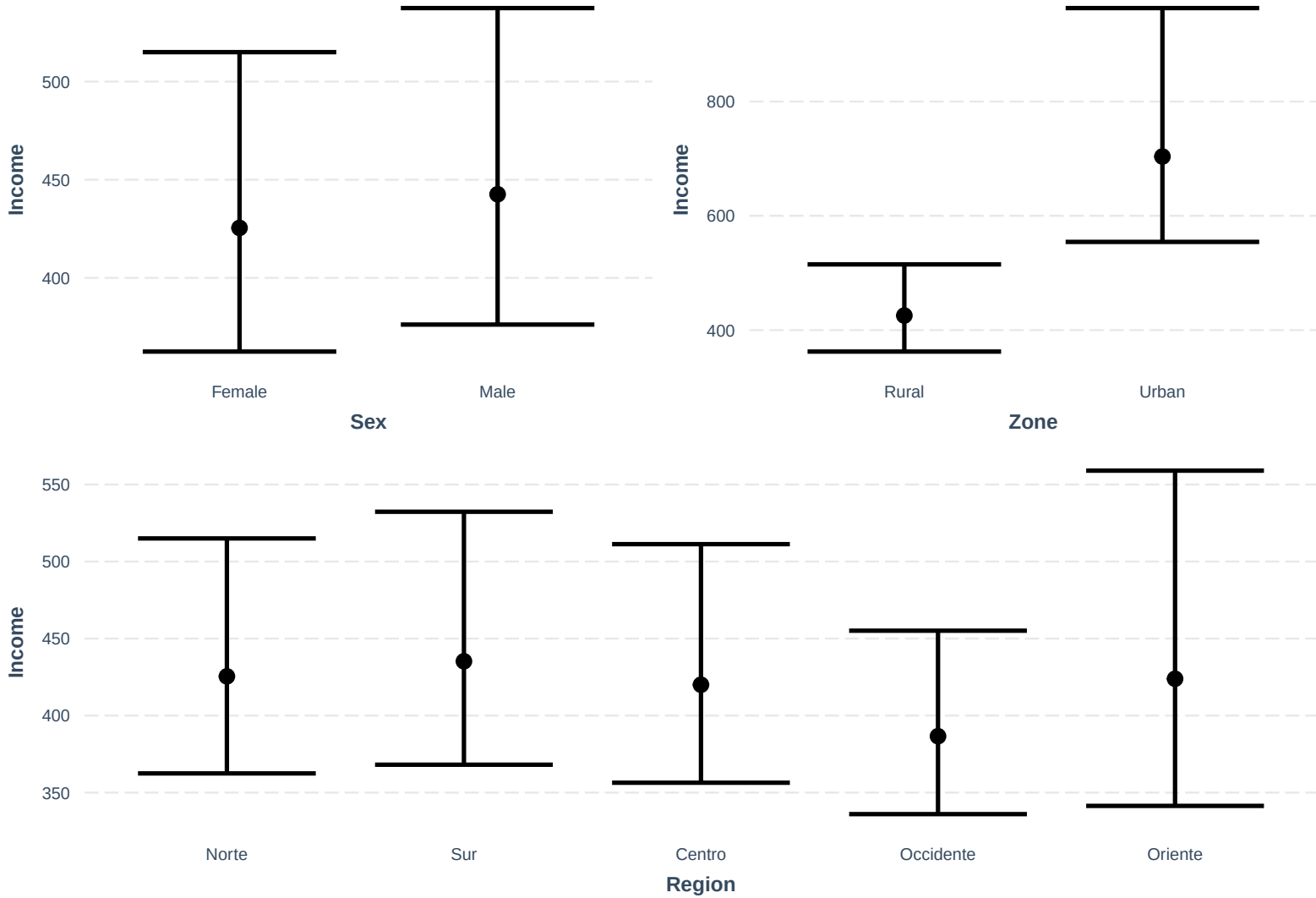


Efecto del modelo.

Efecto de los variables en el modelo.

```
effe_sex <- effect_plot(modelo, pred = Sex,
                        interval = TRUE)
effe_Zona <-effect_plot(modelo, pred = Zone,
                        interval = TRUE)
effe_Region <- effect_plot(modelo, pred = Region,
                           interval = TRUE)
(effe_sex |effe_Zona)/effe_Region
```

Efecto del modelo.



Modelos multinomial

El modelo de regresión logit multinomial es la extensión natural del modelo de regresión logística binomial simple para encuestar respuestas que tienen tres o más categorías distintas. Esta técnica es más apropiada para variables de encuesta con categorías de respuesta nominales.

Modelo multinomial

Para ajustar el modelo debemos tener presente que:

- Su variable dependiente debe medirse en el nivel nominal.

Modelo multinomial

Para ajustar el modelo debemos tener presente que:

- Su variable dependiente debe medirse en el nivel nominal.
- Tiene una o más variables independientes que son continuas , ordinales o nominales (incluidas las variables dicotómicas).

Modelo multinomial

Para ajustar el modelo debemos tener presente que:

- Su variable dependiente debe medirse en el nivel nominal.
- Tiene una o más variables independientes que son continuas , ordinales o nominales (incluidas las variables dicotómicas).
- Tener independencia de las observaciones y la variable dependiente debe tener categorías mutuamente excluyentes y exhaustivas

Modelo multinomial

- No debe haber **multicolinealidad.** La multicolinealidad ocurre cuando tiene dos o más variables independientes que están altamente correlacionadas entre sí.

Modelo multinomial

- No debe haber **multicolinealidad.** La multicolinealidad ocurre cuando tiene dos o más variables independientes que están altamente correlacionadas entre sí.
- Debe haber una relación lineal entre cualquier variable independiente continua y la transformación logit de la variable dependiente

Modelo multinomial

- No debe haber **multicolinealidad.** La multicolinealidad ocurre cuando tiene dos o más variables independientes que están altamente correlacionadas entre sí.
- Debe haber una relación lineal entre cualquier variable independiente continua y la transformación logit de la variable dependiente
- No debe haber valores atípicos, valores de apalancamiento elevados o puntos muy influyentes .

Modelo multinomial

El modelo múltinomial esta dado como:

$$Pr\left(Y_{ik}\right) = Pr\left(y_i = k \mid \boldsymbol{x}_i : \beta_1, \ldots \beta_m\right) = \frac{\exp\left(\beta_{0k} + \boldsymbol{\beta}_k \boldsymbol{x}_i\right)}{\sum_{j=1}^{m} \exp\left(\beta_{0j} + \boldsymbol{\beta}_j \boldsymbol{x}_i\right)}$$

donde $\boldsymbol{\beta}_k$ es el vector de coeficiente de $\boldsymbol{X}$ para la k-ésima categoría de Y.

Modelo multinomial


```
diseno %>% group_by(Employment) %>%
  summarise(Prop = survey_mean(vartype = c("se", "ci"
```

Employment	Prop	Prop_se	Prop_low	Prop_upp
Unemployed	0.0323	0.0057	0.0210	0.0435
Inactive	0.2734	0.0107	0.2523	0.2946
Employed	0.4142	0.0139	0.3867	0.4417
NA	0.2801	0.0140	0.2524	0.3078

Modelo multinomial


```
diseno %>% filter(Age >= 15)%>% group_by(Employment)
  summarise(Prop = survey_mean(vartype = c("se", "ci"
```

Employment	Prop	Prop_se	Prop_low	Prop_upp
Unemployed	0.0448	0.0078	0.0294	0.0602
Inactive	0.3798	0.0150	0.3501	0.4096
Employed	0.5754	0.0131	0.5495	0.6013

Modelo multinomial


```
library(svyVGAM)
diseno_15 <- diseno %>% filter(Age >= 15)
model_mul <- svy_vglm(
    formula = Employment ~ Age + Sex + Region + Zone,
                    design = diseno_15,
     crit = "coef",
    family = multinomial(refLevel = "Unemployed"))
```

La función `broom::tidy()`, que normalmente usamos para limpiar y estandarizar la salida del modelo, no puede ser empleada en este caso, sin embargo, en el link[2] encuentra la función que utilizamos a continuación.

[2] https://tech.popdata.org/pma-data-hub/posts/2021-08-15-covid-analysis/

Modelo multinomial


```
tab_model <- tidy.svyVGAM(model_mul,
                          exponentiate = FALSE,
                          conf.int = FALSE)
tab_model
```

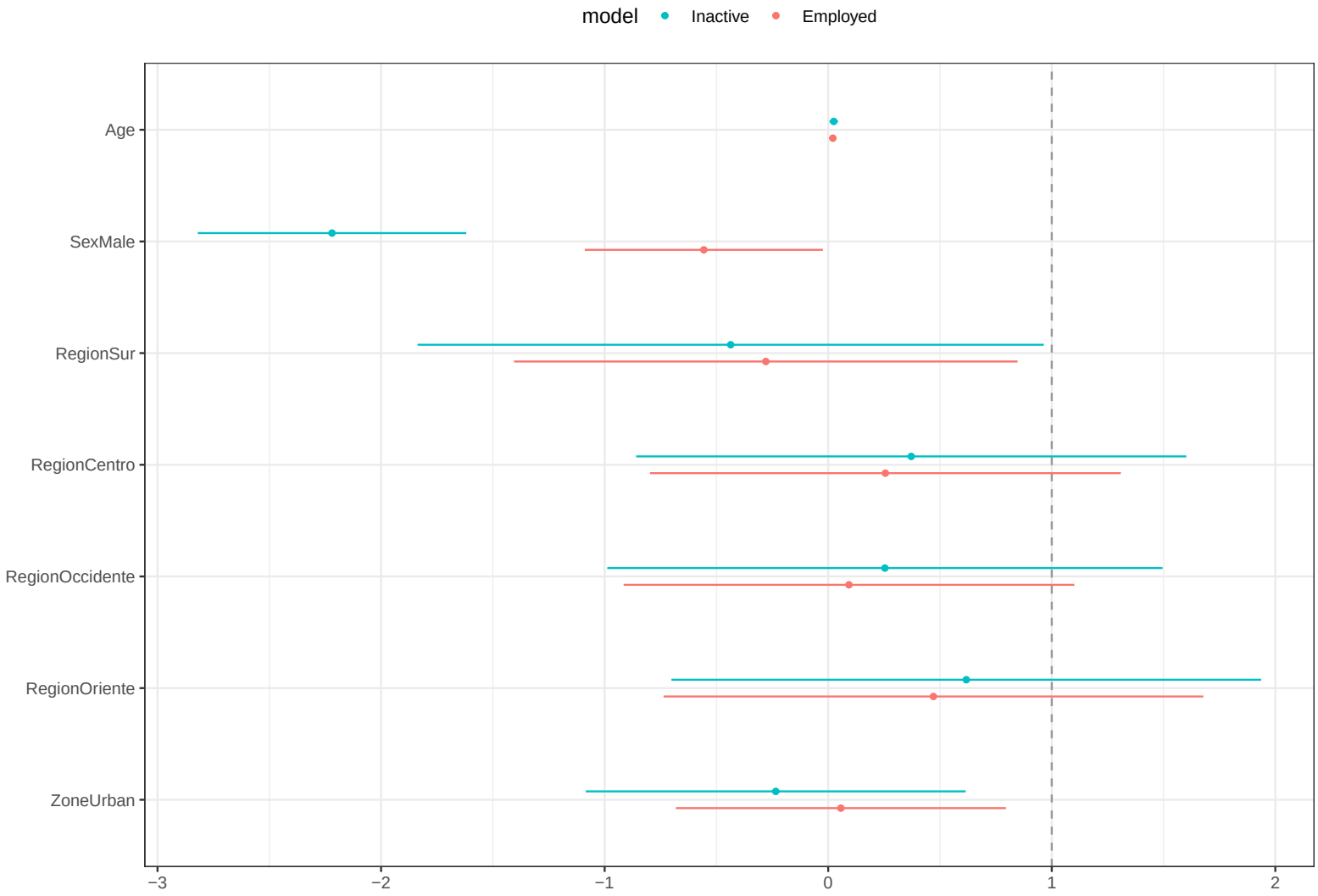
Modelo multinomial

y.level	term	estimate	std.error	statistic	p.value
1	(Intercept)	2.2904	0.7846	2.9193	0.0035
1	Age	0.0247	0.0098	2.5132	0.0120
1	SexMale	-2.2195	0.3063	-7.2454	0.0000
1	RegionSur	-0.4362	0.7146	-0.6105	0.5415
1	RegionCentro	0.3713	0.6277	0.5916	0.5541
1	RegionOccidente	0.2536	0.6336	0.4002	0.6890
1	RegionOriente	0.6176	0.6730	0.9177	0.3588
1	ZoneUrban	-0.2346	0.4335	-0.5412	0.5884
2	(Intercept)	2.0931	0.6322	3.3108	0.0009
2	Age	0.0207	0.0084	2.4672	0.0136
2	SexMale	-0.5563	0.2718	-2.0470	0.0407
2	RegionSur	-0.2791	0.5746	-0.4857	0.6272
2	RegionCentro	0.2558	0.5373	0.4760	0.6341
2	RegionOccidente	0.0928	0.5143	0.1804	0.8568
2	RegionOriente	0.4706	0.6159	0.7640	0.4449

Plot del IC para los coeficientes.


```
tab_model %>%
  mutate(
    model = if_else(
      y.level == 1,
      "Inactive",
      "Employed",
    ),
    sig = gtools::stars.pval(p.value)
  )  %>%
  dotwhisker::dwplot(
    dodge_size = 0.3,
    vline = geom_vline(xintercept = 1, colour = "grey
                       linetype = 2)
  ) +
  guides(color = guide_legend(reverse = TRUE)) +
  theme_bw() + theme(legend.position = "top")
```

Plot del IC para los coeficientes.



modelo multinomial función alternativa.

La función `svy_vglm` realiza la estimación de los parámatros, sin embargo, presenta limitaciones para hacer las predicciones con el modelo, por lo tanto, podemos usar como alternativa.

```
library(CMAverse)
model_mul2 <- svymultinom(
  formula = Employment ~ Age + Sex + Region + Zone,
  weights = diseno_15$variables$wk2,
  data = diseno_15$variables
)
summary(model_mul2)$summarydf
```

Modelo multinomial función alternativa.

Parámetros estimados

	Estimate	Std. Error	t value	Pr(>|t|)
Inactive:(Intercept)	2.2901	0.5587	4.0992	0.0000
Inactive:Age	0.0248	0.0100	2.4721	0.0135
Inactive:SexMale	-2.2195	0.3162	-7.0182	0.0000
Inactive:RegionSur	-0.4361	0.4258	-1.0243	0.3058
Inactive:RegionCentro	0.3715	0.4910	0.7566	0.4494
Inactive:RegionOccidente	0.2537	0.4553	0.5573	0.5774
Inactive:RegionOriente	0.6175	0.5158	1.1973	0.2314
Inactive:ZoneUrban	-0.2346	0.2907	-0.8071	0.4197
Employed:(Intercept)	2.0929	0.5427	3.8563	0.0001
Employed:Age	0.0207	0.0096	2.1496	0.0317
Employed:SexMale	-0.5563	0.3053	-1.8219	0.0686
Employed:RegionSur	-0.2790	0.4106	-0.6794	0.4969
Employed:RegionCentro	0.2559	0.4679	0.5471	0.5844
Employed:RegionOccidente	0.0929	0.4439	0.2093	0.8342
Employed:RegionOriente	0.4705	0.5051	0.9314	0.3517
Employed:ZoneUrban	0.0567	0.2792	0.2033	0.8390

Predicción del modelo

El hacer uso de esta función podemos obtener de forma simple la predicción de las probabilidades

```
tab_pred <- predict(model_mul2, type = "probs") %>%
  data.frame()
tab_pred %>% slice(1:15)
```

Predicción del modelo

Unemployed	Inactive	Employed
0.0387	0.2237	0.7376
0.0151	0.5948	0.3901
0.0310	0.5551	0.4139
0.0908	0.1854	0.7238
0.0134	0.6005	0.3861
0.0317	0.5537	0.4146
0.0467	0.2157	0.7376
0.0173	0.5878	0.3949
0.0791	0.1921	0.7289
0.0295	0.2350	0.7355
0.0095	0.6170	0.3735
0.0621	0.2032	0.7347
0.0687	0.1986	0.7327
0.0839	0.1892	0.7268
0.0730	0.1958	0.7312

Predicción del modelo


```
diseno_15$variables  %<>%
  mutate(predicion = predict(model_mul2))

diseno_15 %>% group_by(Employment) %>%
  summarise(Prop = survey_mean(vartype = c("se", "ci"
```

Employment	Prop	Prop_se	Prop_low	Prop_upp
Unemployed	0.0448	0.0078	0.0294	0.0602
Inactive	0.3798	0.0150	0.3501	0.4096
Employed	0.5754	0.0131	0.5495	0.6013

Predicción del modelo


```
diseno_15 %>% group_by(predicion) %>%
  summarise(Prop = survey_mean(vartype = c("se", "ci"
```

predicion	Prop	Prop_se	Prop_low	Prop_upp
Inactive	0.4649	0.0096	0.4458	0.4840
Employed	0.5351	0.0096	0.5160	0.5542

¡Gracias!

Email: andres.gutierrez@cepal.org