Modelos lineales generalizados

CEPAL

7/3/2022

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:

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

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 consitente 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.
- **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.

- Los modelos de regresión lineal típicos para respuestas continuas son un caso particular de los *GLM*.

Lectura de la base

```
encuesta <- readRDS("../Data/encuesta.rds")
```

Definir diseño de la muestra con `srvyr`

```
library(srvyr)

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

definir nuevas variables

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

Modelo para el ingreso

```
library(ggplot2)
## Estimador de momentos de la distribución gamma
x <- encuesta$Income
n = length(x)
(shape1 = (n*mean(x)^2)/sum((x-mean(x))^2))
```

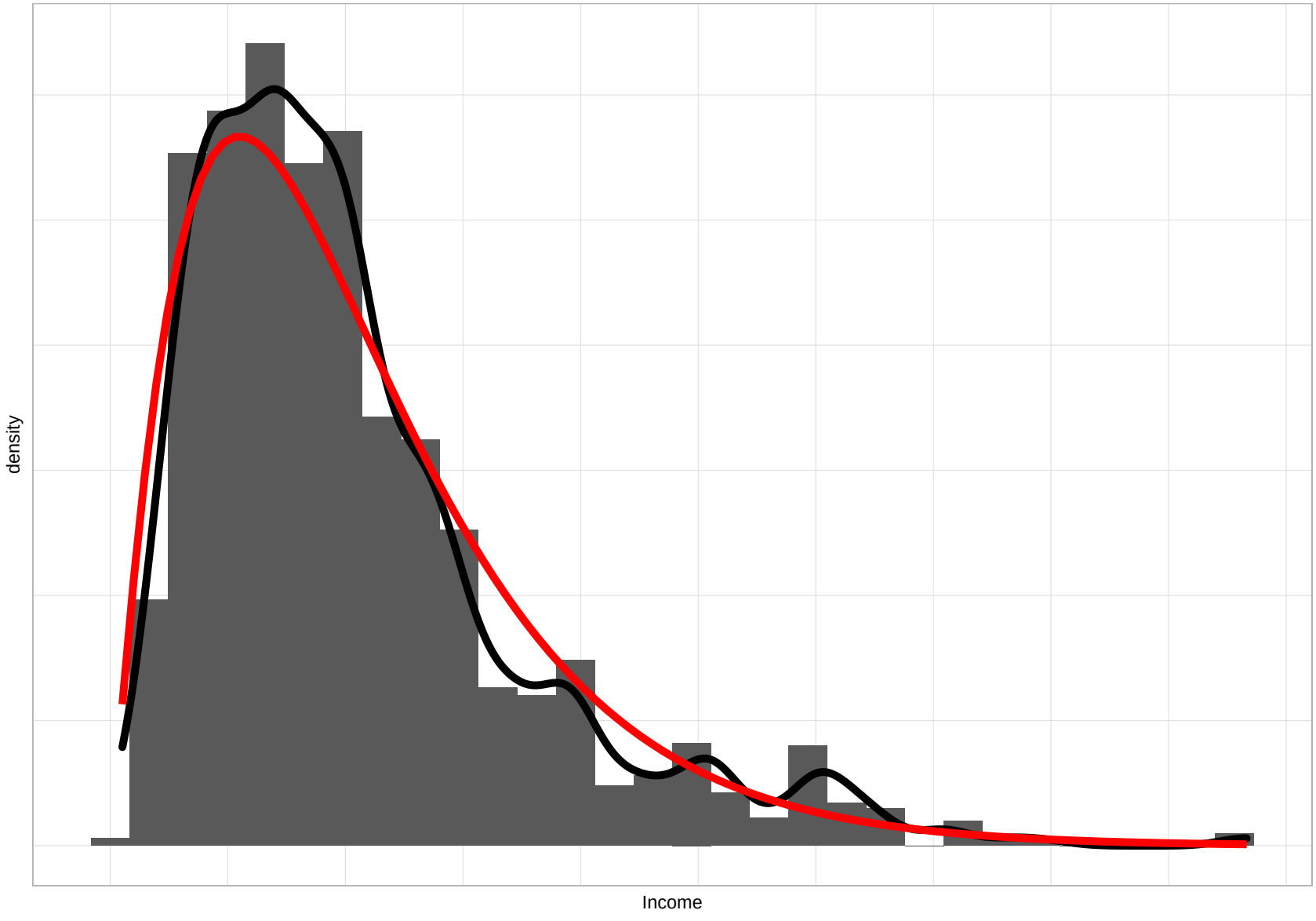
```
## [1] 2.105
```

```
(rate1 = (n*mean(x))/sum((x-mean(x))^2))
```

```
## [1] 0.004021
```

```
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



Modelo gamma

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

```
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

term	estimate	std.error	statistic	p.value
(Intercept)	0.0026	2e-04	14.0753	0.0000
Age	0.0000	0e+00	1.5518	0.1235
SexMale	0.0000	1e-04	0.5330	0.5951
RegionSur	-0.0005	2e-04	-2.0324	0.0445
RegionCentro	-0.0008	2e-04	-4.2642	0.0000
RegionOccidente	-0.0007	2e-04	-3.4736	0.0007
RegionOriente	-0.0001	2e-04	-0.3338	0.7392
ZoneUrban	-0.0007	1e-04	-4.9858	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**.

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

```
## [1] 0.3754
```

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

```
##      variance   SE
## [1,]    0.443 0.05
```

Modelo Gamma

```
mod_s$coefficients
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0026	2e-04	14.0753	0.0000
Age	0.0000	0e+00	1.5518	0.1235
SexMale	0.0000	1e-04	0.5330	0.5951
RegionSur	-0.0005	2e-04	-2.0324	0.0445
RegionCentro	-0.0008	2e-04	-4.2642	0.0000
RegionOccidente	-0.0007	2e-04	-3.4736	0.0007
RegionOriente	-0.0001	2e-04	-0.3338	0.7392
ZoneUrban	-0.0007	1e-04	-4.9858	0.0000

Utilizando la función predict

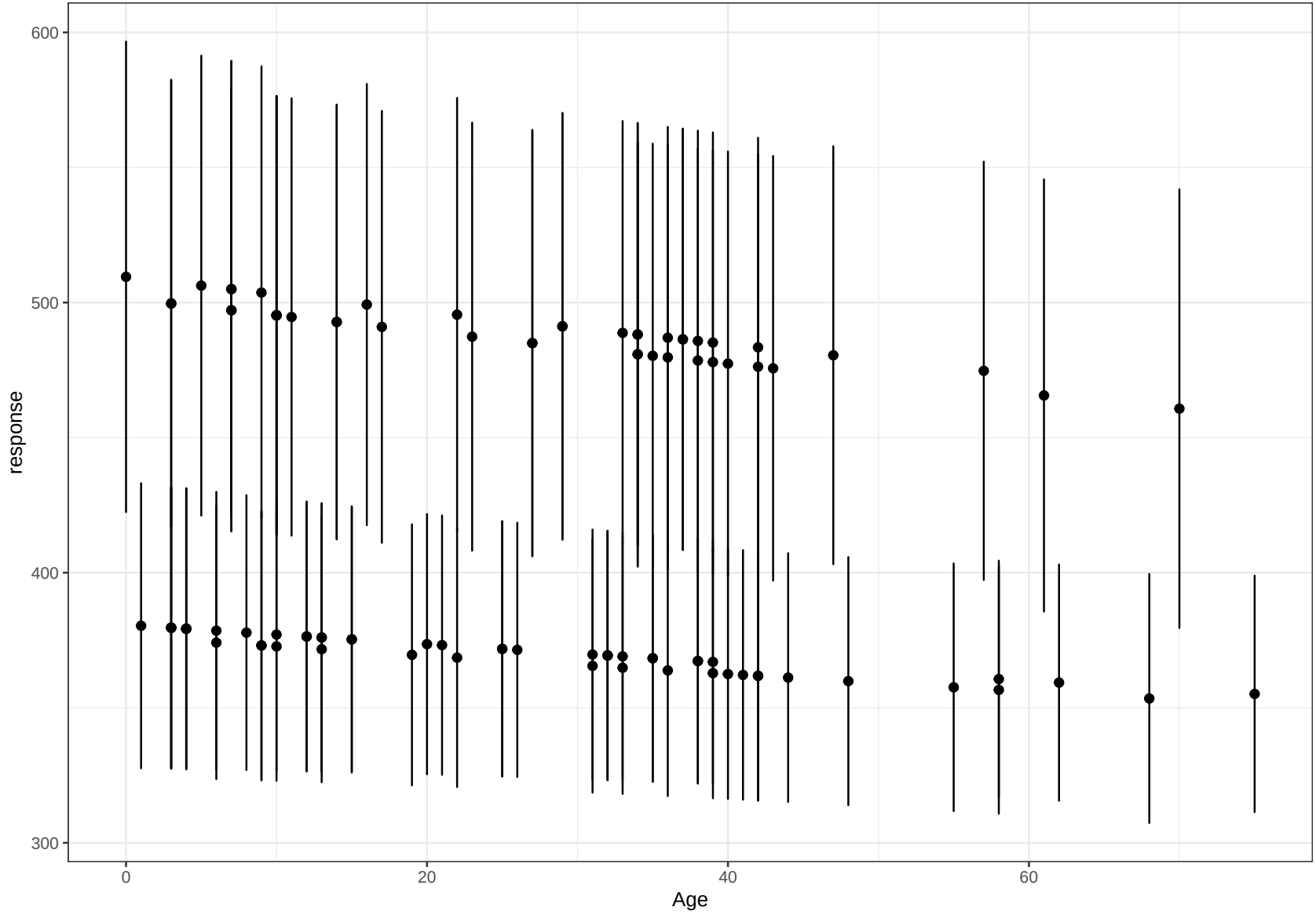
```
pred <- data.frame(
  predict(modelo, type = "response", se = T))
pred_IC <- data.frame(
  confint(predict(modelo, type = "response", se = T)))
colnames(pred_IC) <- c("Lim_Inf", "Lim_Sup")
pred <- bind_cols(pred, pred_IC)
pred$Income <- encuesta$Income
pred$Age <- encuesta$Age
pred %>% slice(1:6L)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

response	SE	Lim_Inf	Lim_Sup	Income	Age
369.4	23.51	323.3	415.4	243.2	32
376.0	25.33	326.4	425.7	243.2	13
373.1	25.47	323.1	423.0	243.2	9
379.6	26.62	327.4	431.8	243.2	3
356.6	23.38	310.8	402.4	223.0	58
360.6	22.35	316.8	404.4	223.0	58

Scaterplot de la predicción

```
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()
```

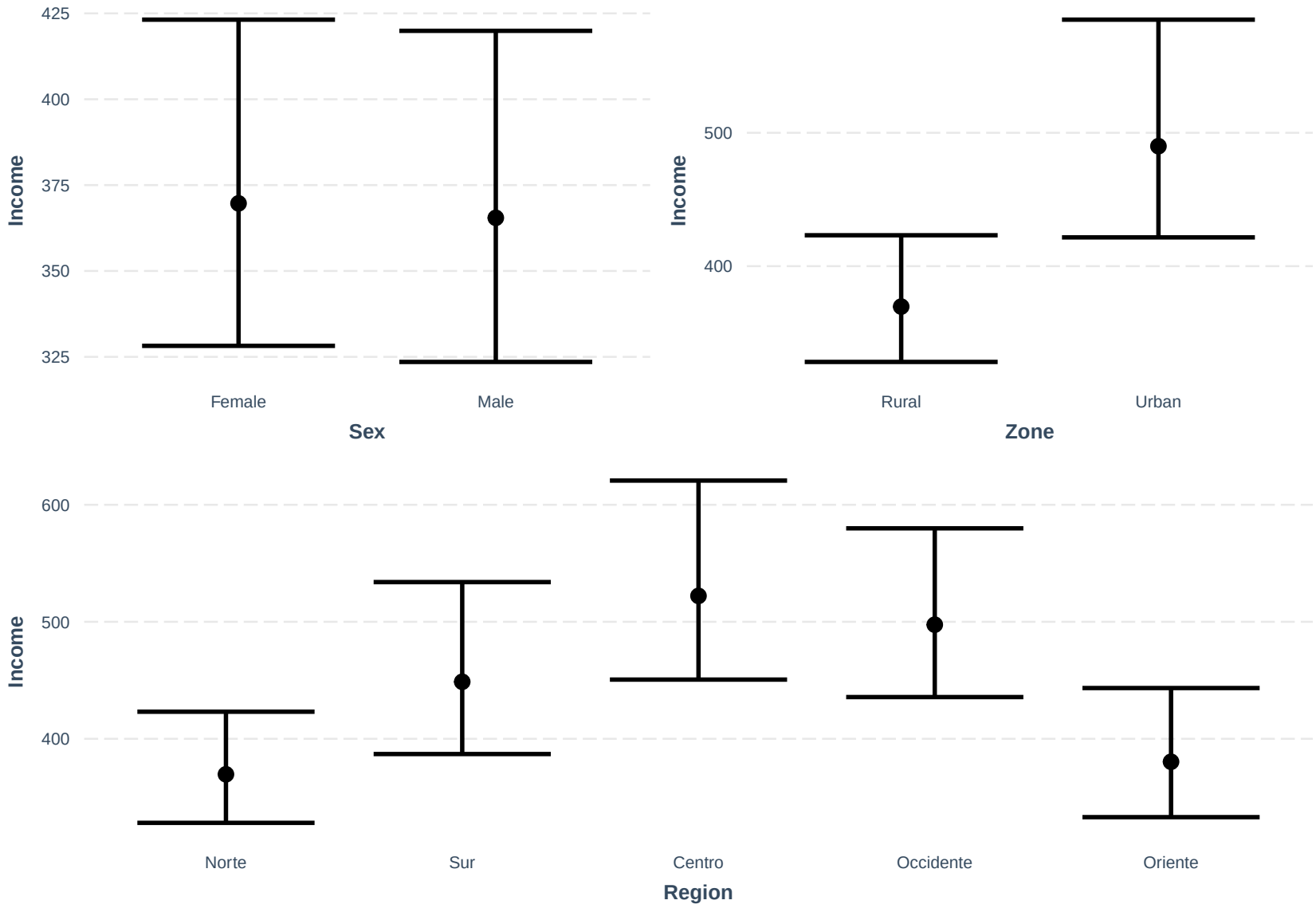
Utilizando la función predict



Efecto del modelo.

```
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.
- 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
- 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

$$Pr\left(Y_{ik}\right) = Pr\left(y_i = k \mid \boldsymbol{x}_i : \boldsymbol{\beta}_1, \ldots \boldsymbol{\beta}_m\right) = \frac{\exp\left(\beta_{0k} + \boldsymbol{\beta}_k \boldsymbol{x}_i\right)}{\sum_{j=1}^{m} \exp\left(\beta_{0j} + \boldsymbol{\beta}_j \boldsymbol{x}_i\right)}$$

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	0.0362	0.0054	0.0255	0.0469
Inactive	0.2900	0.0102	0.2697	0.3102
Employed	0.4062	0.0107	0.3850	0.4275
NA	0.2676	0.0110	0.2459	0.2893

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

Employment	Prop	Prop_se	Prop_low	Prop_upp
Unemployed	0.0494	0.0073	0.0350	0.0639
Inactive	0.3959	0.0120	0.3722	0.4196
Employed	0.5547	0.0132	0.5285	0.5808

Modelo multinomial

```
diseno_15 <- diseno %>% filter(Age >= 15)
library(svyVGAM)
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
```

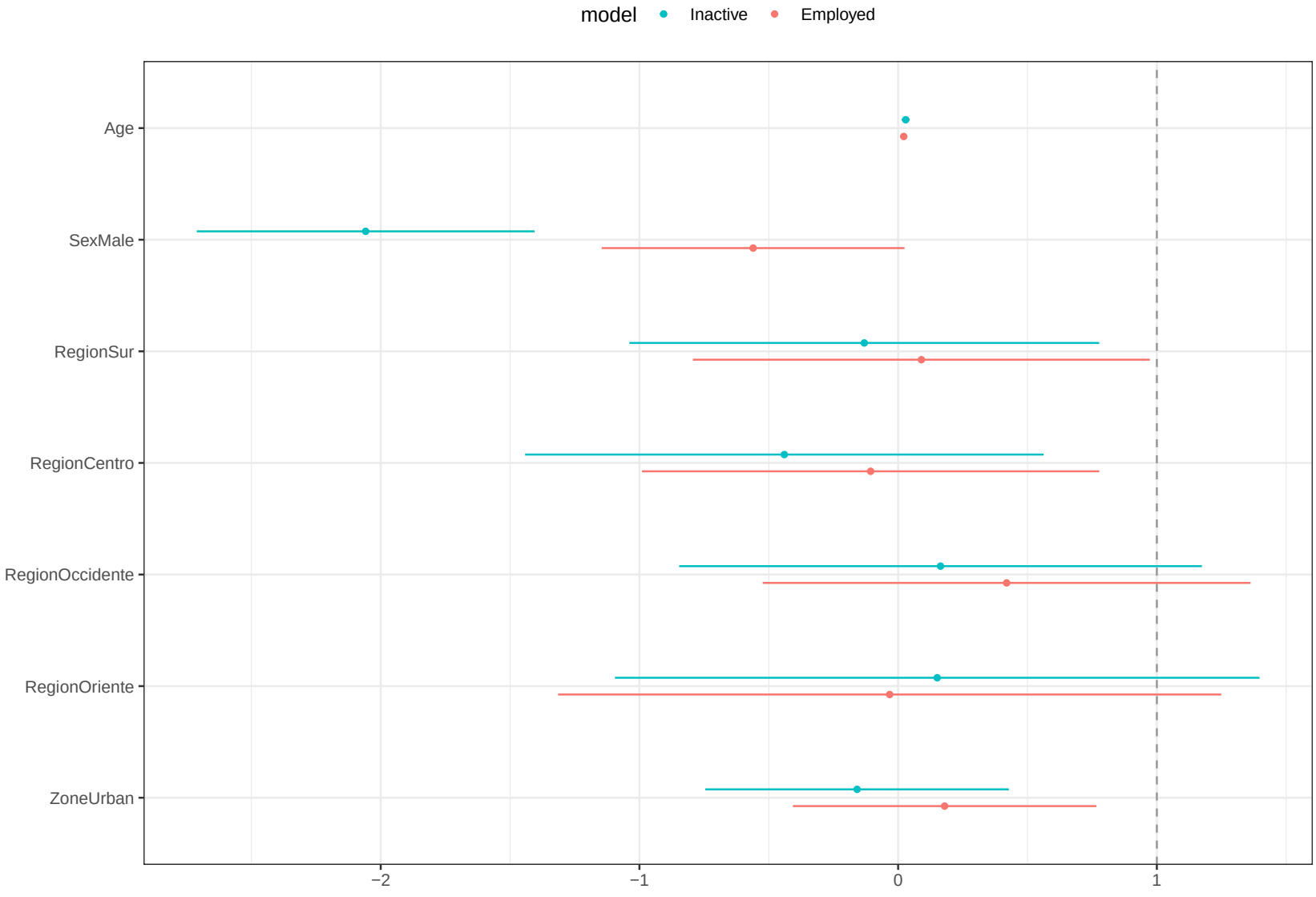
Modelo multinomial

y.level	term	estimate	std.error	statistic	p.value
1	(Intercept)	2.0850	0.5432	3.8385	0.0001
1	Age	0.0292	0.0076	3.8447	0.0001
1	SexMale	-2.0582	0.3331	-6.1798	0.0000
1	RegionSur	-0.1308	0.4633	-0.2823	0.7777
1	RegionCentro	-0.4400	0.5112	-0.8606	0.3894
1	RegionOccidente	0.1638	0.5154	0.3178	0.7506
1	RegionOriente	0.1511	0.6356	0.2377	0.8121
1	ZoneUrban	-0.1587	0.2992	-0.5304	0.5958
2	(Intercept)	1.8328	0.4917	3.7272	0.0002
2	Age	0.0217	0.0062	3.5039	0.0005
2	SexMale	-0.5606	0.2986	-1.8770	0.0605
2	RegionSur	0.0899	0.4506	0.1995	0.8418
2	RegionCentro	-0.1062	0.4509	-0.2356	0.8137
2	RegionOccidente	0.4194	0.4811	0.8717	0.3834
2	RegionOriente	-0.0327	0.6539	-0.0500	0.9602
2	ZoneUrban	0.1796	0.2992	0.6003	0.5483

Plot 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 = "grey60",
                       linetype = 2)
  ) +
  guides(color = guide_legend(reverse = TRUE)) +
  theme_bw() + theme(
    legend.position = "top"
  )
```

Plot coeficientes.



modelo multinomial función alternativa.

La función `svy_vglm` realiza la estimación de los paramatros, 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.

	Estimate	Std. Error	t value	Pr(>|t|)
Inactive:(Intercept)	2.0849	0.4522	4.6104	0.0000
Inactive:Age	0.0292	0.0076	3.8566	0.0001
Inactive:SexMale	-2.0582	0.3059	-6.7294	0.0000
Inactive:RegionSur	-0.1308	0.4185	-0.3127	0.7546
Inactive:RegionCentro	-0.4399	0.4256	-1.0337	0.3014
Inactive:RegionOccidente	0.1638	0.4377	0.3742	0.7083
Inactive:RegionOriente	0.1511	0.4938	0.3059	0.7597
Inactive:ZoneUrban	-0.1587	0.2756	-0.5761	0.5646
Employed:(Intercept)	1.8328	0.4425	4.1424	0.0000
Employed:Age	0.0217	0.0071	3.0677	0.0022
Employed:SexMale	-0.5606	0.2941	-1.9063	0.0568
Employed:RegionSur	0.0899	0.4063	0.2212	0.8249
Employed:RegionCentro	-0.1062	0.4039	-0.2629	0.7927
Employed:RegionOccidente	0.4193	0.4291	0.9773	0.3285
Employed:RegionOriente	-0.0327	0.4877	-0.0671	0.9465
Employed:ZoneUrban	0.1796	0.2652	0.6771	0.4984

Predicción del modelo

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

Predicción del modelo

Unemployed	Inactive	Employed
0.0295	0.6023	0.3682
0.0523	0.2917	0.6560
0.0150	0.6553	0.3296
0.0853	0.2504	0.6642
0.0273	0.6088	0.3639
0.0295	0.6023	0.3682
0.0911	0.2448	0.6641
0.0343	0.5890	0.3767
0.0416	0.3105	0.6479
0.0782	0.2579	0.6639
0.0253	0.6152	0.3595
0.1150	0.2243	0.6608
0.1224	0.2187	0.6590
0.0452	0.5635	0.3913
0.0352	0.5868	0.3780

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	0.0494	0.0073	0.0350	0.0639
Inactive	0.3959	0.0120	0.3722	0.4196
Employed	0.5547	0.0132	0.5285	0.5808

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

predicion	Prop	Prop_se	Prop_low	Prop_upp
Inactive	0.413	0.0117	0.3898	0.4361
Employed	0.587	0.0117	0.5639	0.6102