Modelos lineales generalizados

CEPAL

7/3/2022

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
  )
```

definir nuevas variables

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

Tablas de doble entrada para el tamaño

```
library(survey)
(tab_pobreza_sexo <- svyby(~factor(pobreza), ~Sex,
      FUN = svytotal, design = as.svrepdesign(diseno),
      se=F, na.rm=T, ci=T, keep.var=TRUE))
```

	Sex	factor(pobreza)0	factor(pobreza)1	se1	se2
Female	Female	48010	31180	2206	3001
Male	Male	42310	28766	2093	2239

```
(tab <- svytable(~pobreza + Sex, design = diseno))
```

pobreza/Sex	Female	Male
0	48010	42310
1	31180	28766

Tablas de doble entrada para el proporción

```
(tab_pobreza_sexo <- svyby(~factor(pobreza), ~Sex,
      FUN = svymean, design = as.svrepdesign(diseno),
      se=F, na.rm=T, ci=T, keep.var=TRUE))
```

	Sex	factor(pobreza)0	factor(pobreza)1	se1	se2
Female	Female	0.6063	0.3937	0.0304	0.0304
Male	Male	0.5953	0.4047	0.0270	0.0270

```
prop.table(tab, margin = 2)
```

pobreza/Sex	Female	Male
0	0.6063	0.5953
1	0.3937	0.4047

Prueba de independencia F

```
summary(tab, statistic = "F")
```

```
##          Sex
## pobreza Female  Male
##       0  48010 42310
##       1  31180 28766
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  NextMethod()
## F = 0.35, ndf = 1, ddf = 119, p-value = 0.6
```

Prueba de independencia ChiSq

```
summary(tab, statistic = "Chisq")
```

```
##          Sex
## pobreza Female  Male
##       0  48010 42310
##       1  31180 28766
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  NextMethod()
## X-squared = 0.33, df = 1, p-value = 0.6
```

Prueba de independencia Wald

```
summary(tab, statistic = "Wald")
```

```
##          Sex
## pobreza Female  Male
##       0  48010 42310
##       1  31180 28766
##
##  Design-based Wald test of association
##
## data:  NextMethod()
## F = 0.35, ndf = 1, ddf = 119, p-value = 0.6
```

Prueba de independencia adjWald

```
summary(tab, statistic = "adjWald")
```

```
##          Sex
## pobreza Female  Male
##       0  48010 42310
##       1  31180 28766
##
##  Design-based Wald test of association
##
## data:  NextMethod()
## F = 0.35, ndf = 1, ddf = 119, p-value = 0.6
```

Prueba de independencia lincom

```
summary(tab, statistic = "lincom")
```

```
##          Sex
## pobreza Female  Male
##       0  48010 42310
##       1  31180 28766
##
##  Pearson's X^2: asymptotic exact distribution
##
## data:  NextMethod()
## X-squared = 0.33, p-value = 0.6
```

Prueba de independencia saddlepoint

```
summary(tab, statistic = "saddlepoint")
```

```
##          Sex
## pobreza Female  Male
##       0  48010 42310
##       1  31180 28766
##
##  Pearson's X^2: saddlepoint approximation
##
## data:  NextMethod()
## X-squared = 0.33, p-value = 0.6
```

Modelo log lineal para tablas de contingencia

$$\log(p_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY},$$

donde:

- p_{ijk} = la proporción esperada en la celda bajo el modelo.
- $\mu = \log(p_0) = \frac{1}{\#\ de\ celdas}$

Modelo log lineal para tablas de contingencia

```
mod1 <- svyloglin(~pobreza+Sex + pobreza:Sex , diseno)
(s1 <- summary(mod1))
```

```
## Loglinear model: svyloglin(~pobreza + Sex + pobreza:Sex
##                      coef      se          p
## pobreza1          0.20437 0.05676 0.0003172
## Sex1              0.05174 0.02119 0.0146150
## pobreza1:Sex1     0.01144 0.01933 0.5538109
```

Modelo log lineal para tablas de contingencia

```
mod2 <- svyloglin(~pobreza+Sex, diseno)
(s2 <- summary(mod2))
```

```
## Loglinear model: svyloglin(~pobreza + Sex, diseno)
##                 coef      se          p
## pobreza1 0.20497 0.05690 0.0003157
## Sex1     0.05405 0.01919 0.0048540
```

Modelo log lineal para tablas de contingencia

```
anova(mod1, mod2)
```

```
## Analysis of Deviance Table
##  Model 1: y ~ pobreza + Sex
## Model 2: y ~ pobreza + Sex + pobreza:Sex
## Deviance= 0.3254 p= 0.555
## Score= 0.3254 p= 0.5549
```

Modelo de regresión logistica

$$g(\pi(x)) = logit(\pi(x)) = z = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = B_0 + B_1 x_1 + \cdots + B_p x_p$$

Tablas de contingencia

Sex	pobreza	se	ci_l	ci_u
Female	0.3937	0.0304	0.3342	0.4533
Male	0.4047	0.0270	0.3518	0.4576

Zone	pobreza	se	ci_l	ci_u
Rural	0.4206	0.0485	0.3255	0.5157
Urban	0.3789	0.0269	0.3261	0.4317

Region	pobreza	se	ci_l	ci_u
Norte	0.4213	0.0522	0.3189	0.5236
Sur	0.3079	0.0539	0.2023	0.4136
Centro	0.2509	0.0461	0.1606	0.3411
Occidente	0.3980	0.0385	0.3226	0.4734
Oriente	0.5453	0.0768	0.3948	0.6957

Prueba de independencia ChiSq

```
## Pearson's X^2: Rao & Scott adjustment
pobreza_sex <-  svychisq(
  formula = ~pobreza + Sex, design = diseno)
pobreza_Zona <- svychisq(
  formula = ~pobreza + Zone, design = diseno)
pobreza_Region <- svychisq(
  formula = ~pobreza + Region, design = diseno)
bind_rows( tidy( pobreza_sex),
           tidy(pobreza_Zona),
           tidy(pobreza_Region)) %>%
  dplyr::select(-method)
```

ndf	ddf	statistic	p.value
1.000	119.0	0.3506	0.5549
1.000	119.0	0.5730	0.4505
3.194	380.1	4.2241	0.0049

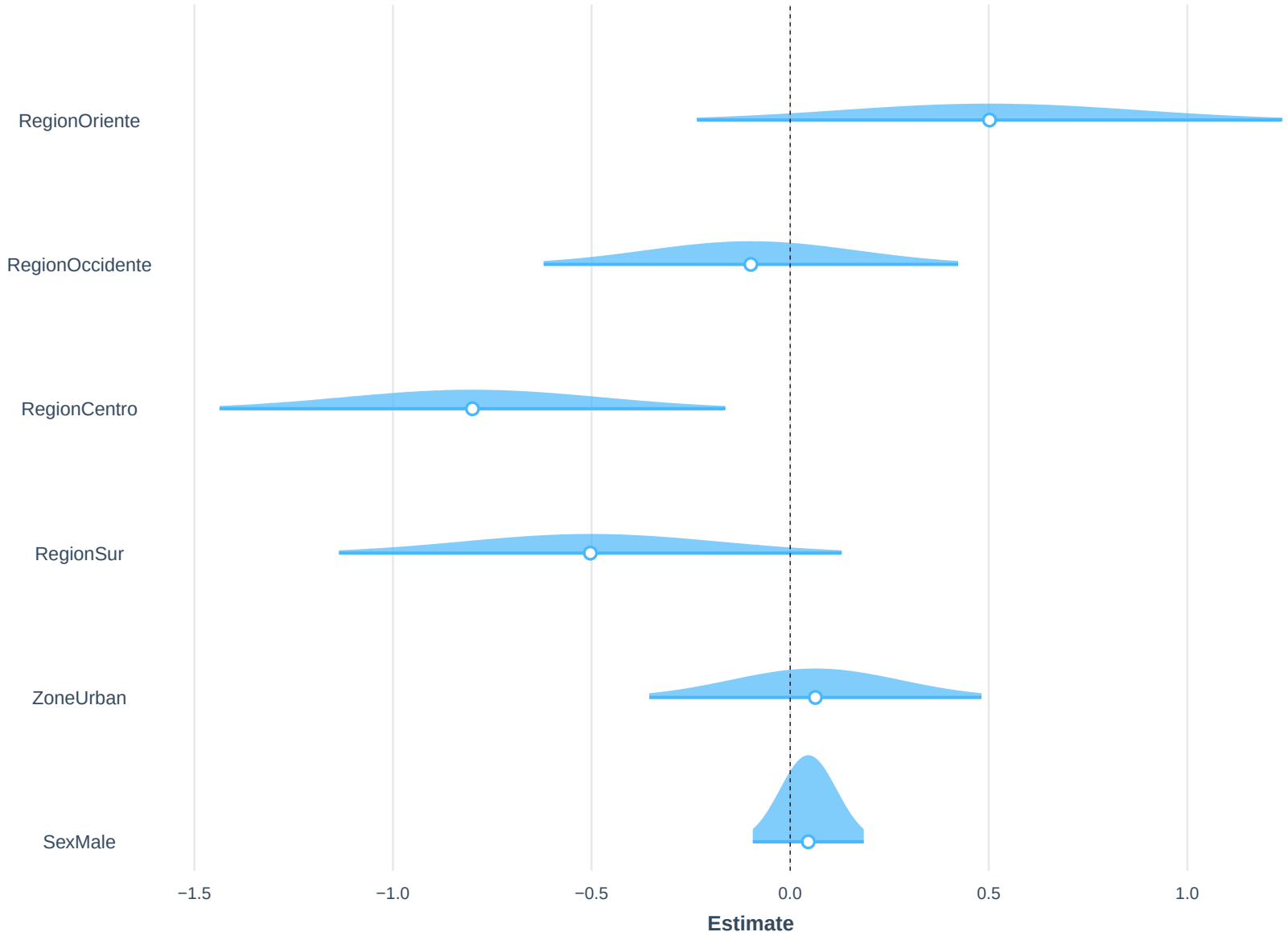
Modelo log lineal ajustado

```
mod_loglin <- svyglm(pobreza ~ Sex + Zone + Region,
                     family=quasibinomial, design=diseno)
tidy(mod_loglin)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.3664	0.2486	-1.4735	0.1434
SexMale	0.0455	0.0713	0.6384	0.5245
ZoneUrban	0.0634	0.2134	0.2970	0.7670
RegionSur	-0.5031	0.3229	-1.5582	0.1220
RegionCentro	-0.7999	0.3249	-2.4619	0.0153
RegionOccidente	-0.0990	0.2663	-0.3719	0.7107
RegionOriente	0.5019	0.3759	1.3349	0.1846

Plot de la distribución de los betas

```
plot_summs(mod_loglin,
           scale = TRUE, plot.distributions = TRUE)
```



Modelo log lineal ajustado

```
bind_cols(
data.frame(exp_estimado = exp(coef(mod_loglin))),
as.data.frame(exp(confint(mod_loglin)))
)
```

	exp_estimado	2.5 %	97.5 %
(Intercept)	0.6933	0.4236	1.1345
SexMale	1.0466	0.9087	1.2054
ZoneUrban	1.0654	0.6981	1.6262
RegionSur	0.6046	0.3189	1.1463
RegionCentro	0.4494	0.2361	0.8554
RegionOccidente	0.9057	0.5344	1.5349
RegionOriente	1.6518	0.7843	3.4787

Estadístico de Wald sobre los parámetros

```
regTermTest(model = mod_loglin, ~Sex)
```

```
## Wald test for Sex
##  in svyglm(formula = pobreza ~ Sex + Zone + Region, desi
##      family = quasibinomial)
## F =  0.4076  on  1  and  113  df: p= 0.52
```

```
regTermTest(model = mod_loglin, ~Zone)
```

```
## Wald test for Zone
##  in svyglm(formula = pobreza ~ Sex + Zone + Region, desi
##      family = quasibinomial)
## F =  0.08823  on  1  and  113  df: p= 0.77
```

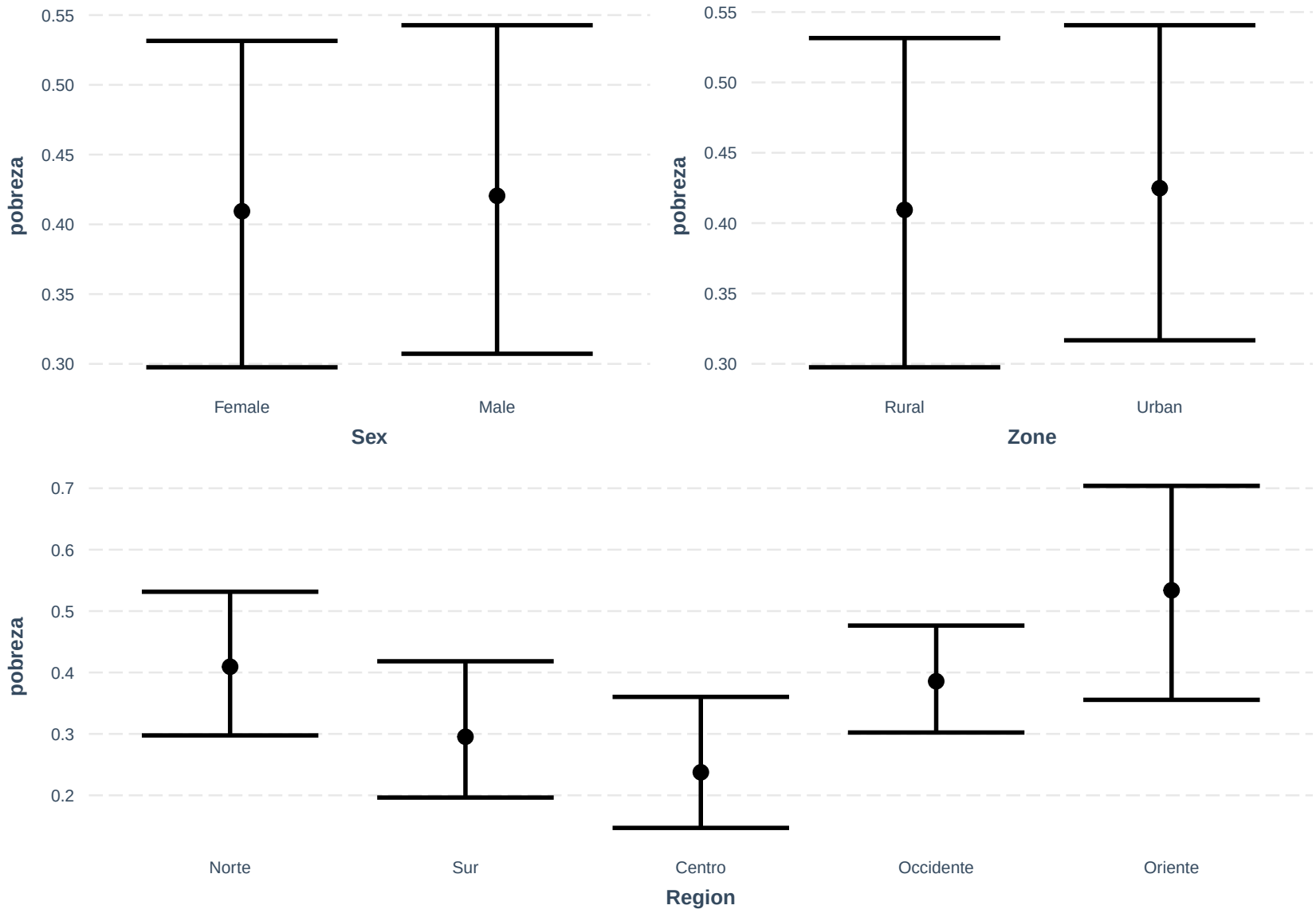
```
regTermTest(model = mod_loglin, ~Region)
```

```
## Wald test for Region
##  in svyglm(formula = pobreza ~ Sex + Zone + Region, desi
##      family = quasibinomial)
## F =  4.33  on  4  and  113  df: p= 0.0027
```

Efecto del modelo.

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

Efecto del modelo.



Modelo log lineal ajustado con interacciones

```
mod_loglin_int <- svyglm(pobreza ~ Sex + Zone + Region +
                                   Sex:Zone + Sex:Region,
                         family=quasibinomial, design=diseno)
tab_mod <- tidy(mod_loglin_int) %>% arrange(p.value)
tab_mod %>% slice(1:6)
```

term	estimate	std.error	statistic	p.value
RegionCentro	-0.7863	0.3539	-2.2217	0.0284
RegionSur	-0.5432	0.3776	-1.4383	0.1532
SexMale:RegionOriente	0.2616	0.1866	1.4022	0.1637
(Intercept)	-0.3196	0.2623	-1.2183	0.2257
RegionOriente	0.3780	0.3854	0.9809	0.3288
SexMale:RegionOccidente	0.1648	0.1779	0.9261	0.3564

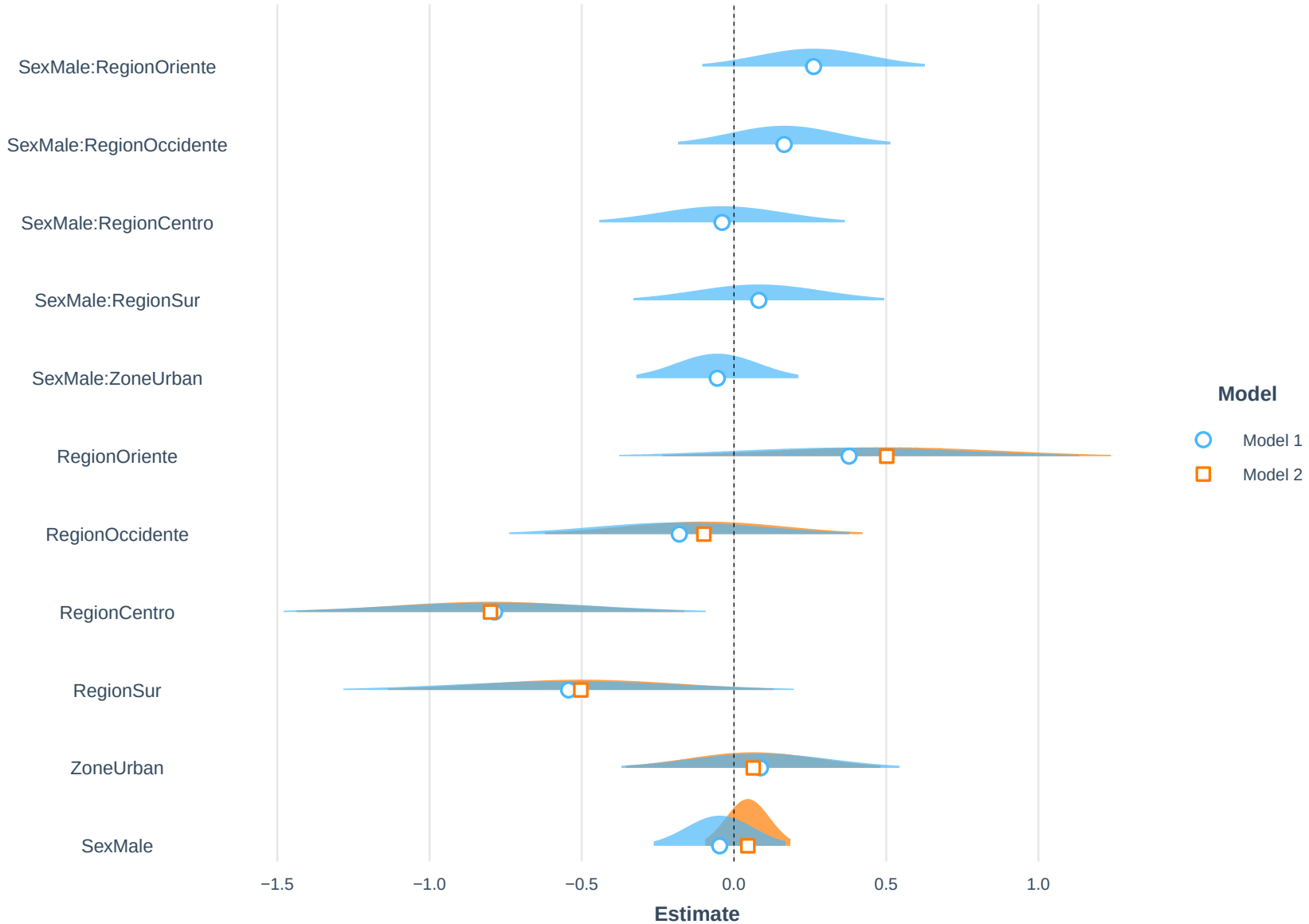
Modelo log lineal ajustado con interacciones

```
tab_mod %>% slice(7:12)
```

term	estimate	std.error	statistic	p.value
RegionOccidente	-0.1796	0.2849	-0.6304	0.5297
SexMale	-0.0470	0.1105	-0.4257	0.6712
SexMale:ZoneUrban	-0.0548	0.1357	-0.4037	0.6872
SexMale:RegionSur	0.0817	0.2103	0.3884	0.6985
ZoneUrban	0.0869	0.2330	0.3731	0.7098
SexMale:RegionCentro	-0.0391	0.2059	-0.1897	0.8499

Plot de la distribución de los betas

```
plot_summs(mod_loglin_int, mod_loglin, scale = TRUE, plot.c
```



Modelo log lineal ajustado

	exp_estimado	2.5 %	97.5 %
(Intercept)	0.7264	0.4319	1.2219
SexMale	0.9541	0.7665	1.1876
ZoneUrban	1.0908	0.6874	1.7312
RegionSur	0.5809	0.2748	1.2280
RegionCentro	0.4555	0.2259	0.9187
RegionOccidente	0.8356	0.4750	1.4698
RegionOriente	1.4594	0.6799	3.1328
SexMale:ZoneUrban	0.9467	0.7234	1.2389
SexMale:RegionSur	1.0851	0.7152	1.6463
SexMale:RegionCentro	0.9617	0.6395	1.4463
SexMale:RegionOccidente	1.1791	0.8287	1.6778
SexMale:RegionOriente	1.2990	0.8974	1.8802

Estadístico de Wald sobre los parámetros

```
regTermTest(model = mod_loglin_int, ~Sex)
```

```
## Wald test for Sex
##  in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex
##      design = diseno, family = quasibinomial)
## F =  0.1812  on  1  and  108  df: p= 0.67
```

```
regTermTest(model = mod_loglin_int, ~Zone)
```

```
## Wald test for Zone
##  in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex
##      design = diseno, family = quasibinomial)
## F =  0.1392  on  1  and  108  df: p= 0.71
```

```
regTermTest(model = mod_loglin_int, ~Region)
```

```
## Wald test for Region
##  in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex
##      design = diseno, family = quasibinomial)
## F =  3.14  on  4  and  108  df: p= 0.017
```

Estadístico de Wald sobre los parámetros

```
regTermTest(model = mod_loglin_int, ~Sex:Zone)
```

```
## Wald test for Sex:Zone
##  in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex
##     design = diseno, family = quasibinomial)
## F =  0.163  on  1  and  108  df: p= 0.69
```

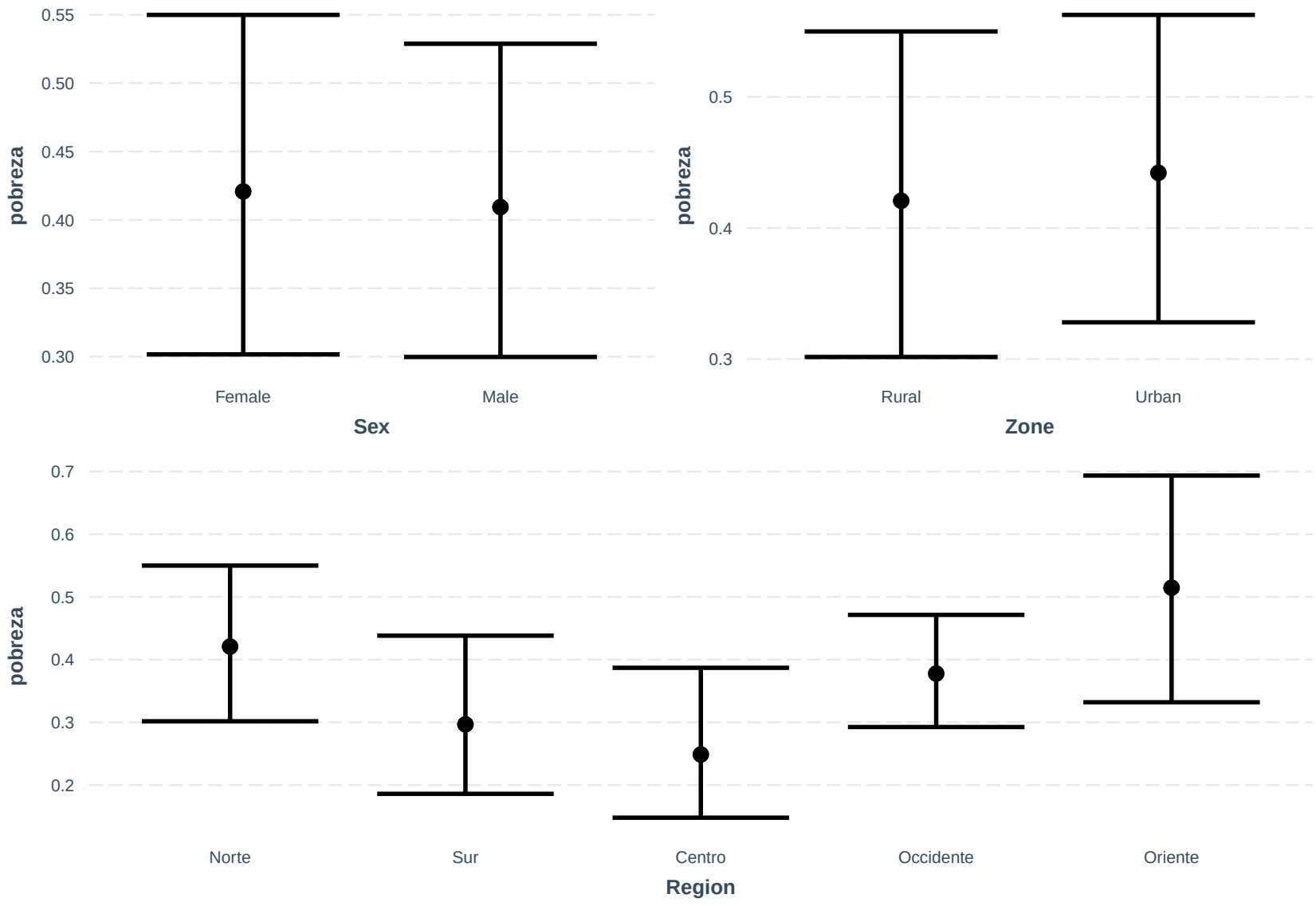
```
regTermTest(model = mod_loglin_int, ~Sex:Region)
```

```
## Wald test for Sex:Region
##  in svyglm(formula = pobreza ~ Sex + Zone + Region + Sex
##     design = diseno, family = quasibinomial)
## F =  0.7353  on  4  and  108  df: p= 0.57
```

Efecto del modelo.

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

Efecto del modelo.



Modelo log lineal ajustado con Q_Weighting

```
fit_wgt <- lm(wk ~  Sex + Zone + Region , 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
  )
```

Modelo log lineal ajustado con Q_Weighting

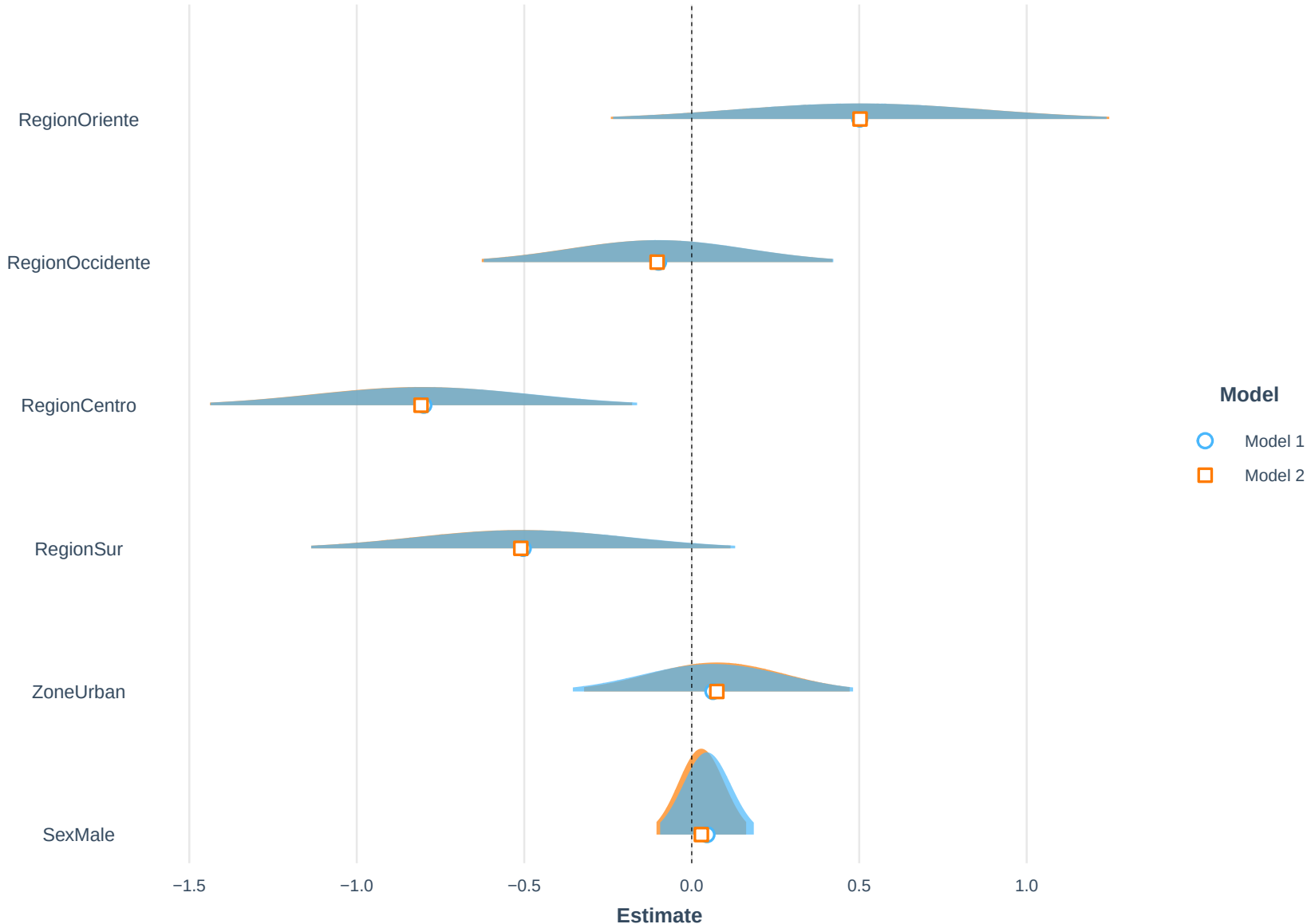
```
diseno_qwgt <- diseno_qwgt %>% mutate(
  pobreza = ifelse(Poverty != "NotPoor", 1, 0))

mod_loglin_qwgt <- svyglm(pobreza ~ Sex + Zone + Region,
                          family=quasibinomial,
                          design=diseno_qwgt)
(tab_mod <- tidy(mod_loglin_qwgt) )
```

Modelo log lineal ajustado con Q_Weighting

term	estimate	std.error	statistic	p.value
(Intercept)	-0.3611	0.2519	-1.4334	0.1545
SexMale	0.0287	0.0683	0.4200	0.6753
ZoneUrban	0.0750	0.2023	0.3706	0.7116
RegionSur	-0.5103	0.3194	-1.5976	0.1129
RegionCentro	-0.8081	0.3215	-2.5136	0.0134
RegionOccidente	-0.1033	0.2671	-0.3867	0.6997
RegionOriente	0.5025	0.3796	1.3238	0.1882

Plot de la distribución de los betas

```
plot_summs(mod_loglin, mod_loglin_qwgt,
           scale = TRUE, plot.distributions = TRUE)
```



Modelo log lineal ajustado

	exp_estimado	2.5 %	97.5 %
(Intercept)	0.6969	0.4231	1.1479
SexMale	1.0291	0.8989	1.1782
ZoneUrban	1.0779	0.7220	1.6092
RegionSur	0.6003	0.3188	1.1303
RegionCentro	0.4457	0.2357	0.8427
RegionOccidente	0.9019	0.5312	1.5310
RegionOriente	1.6528	0.7792	3.5062

Estadístico de Wald sobre los parámetros

```
regTermTest(model = mod_loglin_qwgt, ~Sex)
```

```
## Wald test for Sex
##  in svyglm(formula = pobreza ~ Sex + Zone + Region, desi
##      family = quasibinomial)
## F =  0.1764  on  1  and  113  df: p= 0.68
```

```
regTermTest(model = mod_loglin_qwgt, ~Zone)
```

```
## Wald test for Zone
##  in svyglm(formula = pobreza ~ Sex + Zone + Region, desi
##      family = quasibinomial)
## F =  0.1374  on  1  and  113  df: p= 0.71
```

```
regTermTest(model = mod_loglin_qwgt, ~Region)
```

```
## Wald test for Region
##  in svyglm(formula = pobreza ~ Sex + Zone + Region, desi
##      family = quasibinomial)
## F =  4.225  on  4  and  113  df: p= 0.0032
```

Efecto del modelo.

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

Efecto del modelo.