

Sesión 3. Reporte de resultados

Informe de resultados

Nombre

20-04-2022

Contents

El primer paso: la carga	2
Exploración de variables	2
Gráficos de dispersión	3
Gráfico de barras	6
Regresión	7
Información de modelos	10
Gráficar los modelos	11

El primer paso: la carga

El primero paso SIEMPRE será cargar los paquetes, sin embargo, este paso no es necesario mostrarlo en el reporte, para ello es que configuraremos el ‘chunk’ para que no se muestre en el reporte “renderizado”, para esto usaremos el argumento ‘include=FALSE’

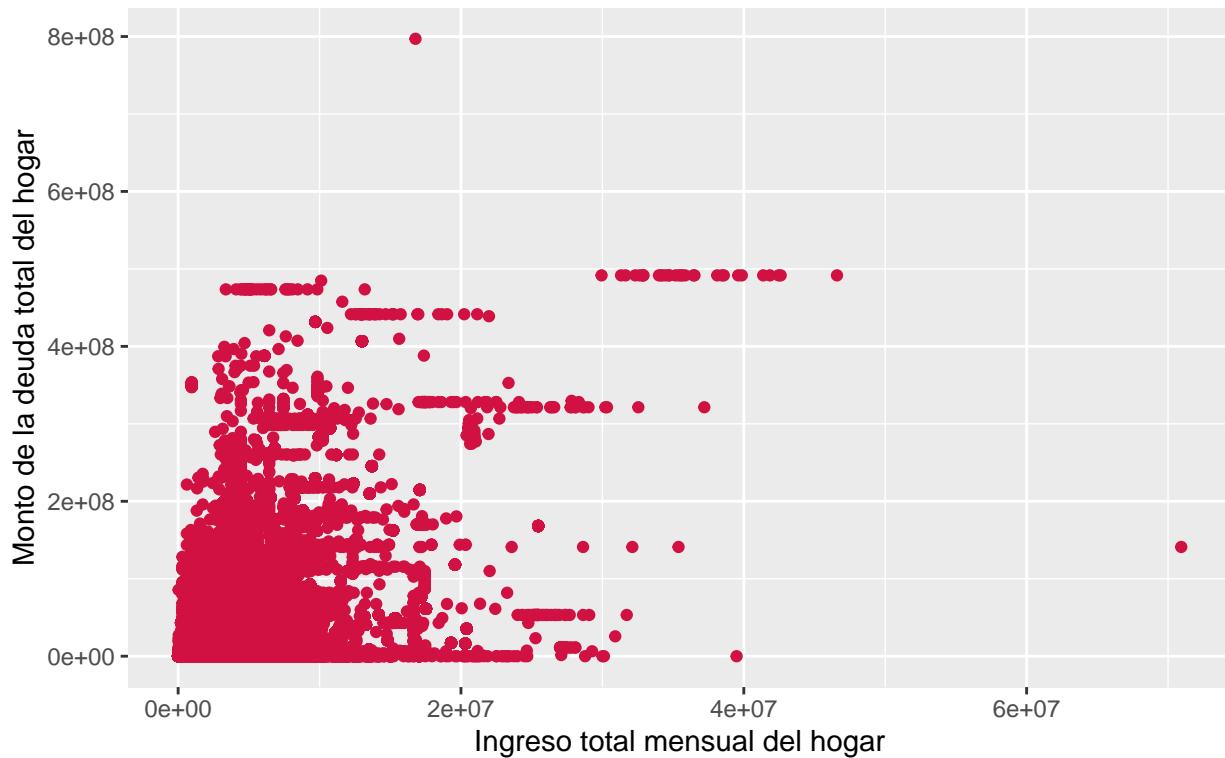
Haremos lo mismo con la carga de datos

Exploración de variables

Para explorar las variables a trabajar en el informe, usaremos el paquete ‘sjPlot’ para realizar un gráfico de dispersión, usando la función ‘plot_scatter’, podemos incorporar título y cambiar colores, para esto usamos los argumentos de la función.

```
plot_scatter(datos, ytoth, d_toth,
              title = "1. Gráfico de dispersión entre ingresos totales y gastos totales")
```

1. Gráfico de dispersión entre ingresos totales y gastos totales

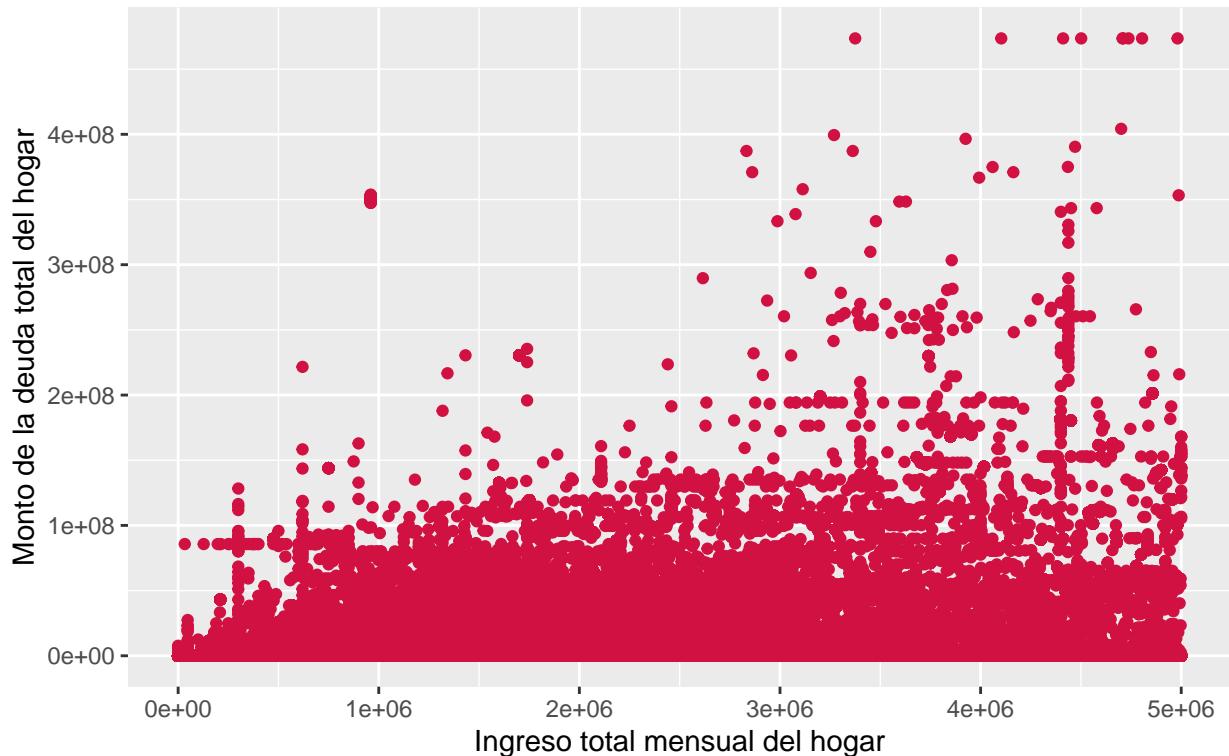


Como podemos ver, no existe una buena visualización de los datos, lo bueno de trabajar con ‘sjPlot’, es que es compatible con el universo ‘tidyverse’, por lo que podemos usar el operador pipeline. Entonces filtremos los casos y veamos como queda.

Gráficos de dispersión

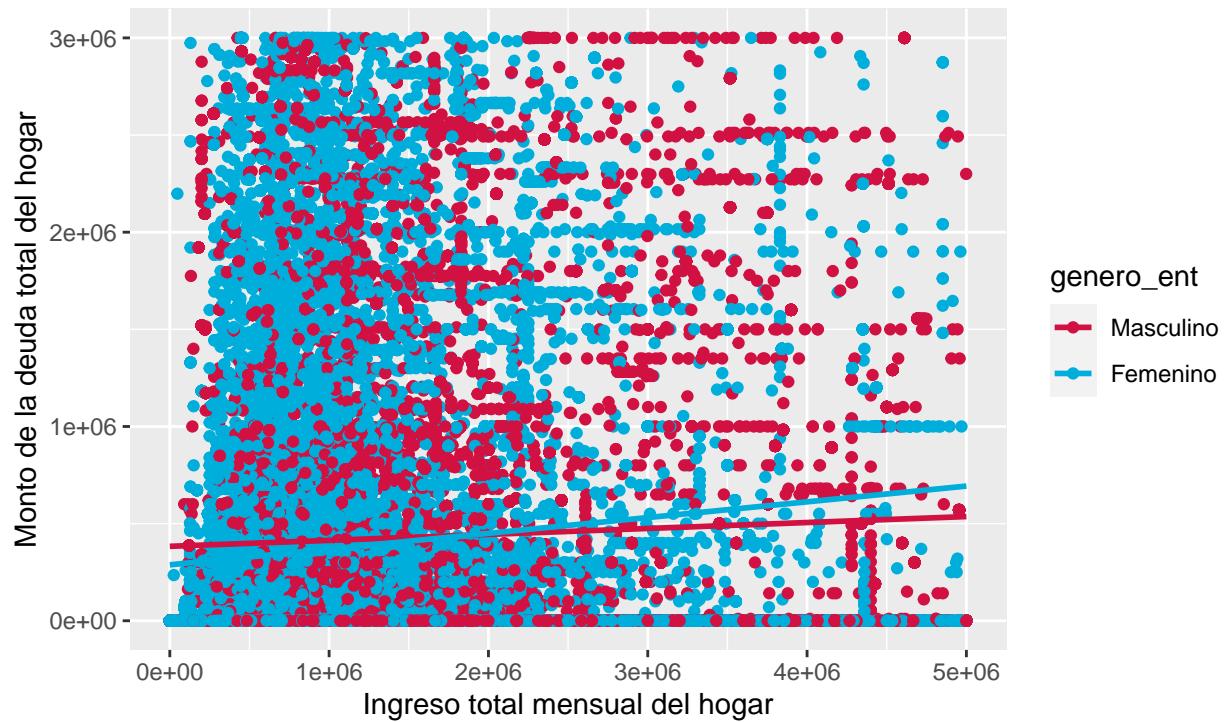
```
datos %>% filter(ytoth <= 5000000) %>%
  plot_scatter(., ytoth, d_toth,
               title = "2. Gráfico de dispersión entre ingresos totales y gastos totales, con casos filtrados")
```

2. Gráfico de dispersión entre ingresos totales y gastos totales, con casos filtrados



ya se visualiza mejor, hay muchas configuraciones que puedes explorar, así mismo con el argumento 'fit.grps = "lm"' puedes incorporar una recta según el método que prefieras, inclusive puedes añadir una categoría de diferenciación, veamos como queda con la variable género

3. Gráfico de dispersión entre ingresos totales y gastos totales, con casos filtrados y recta de regresión



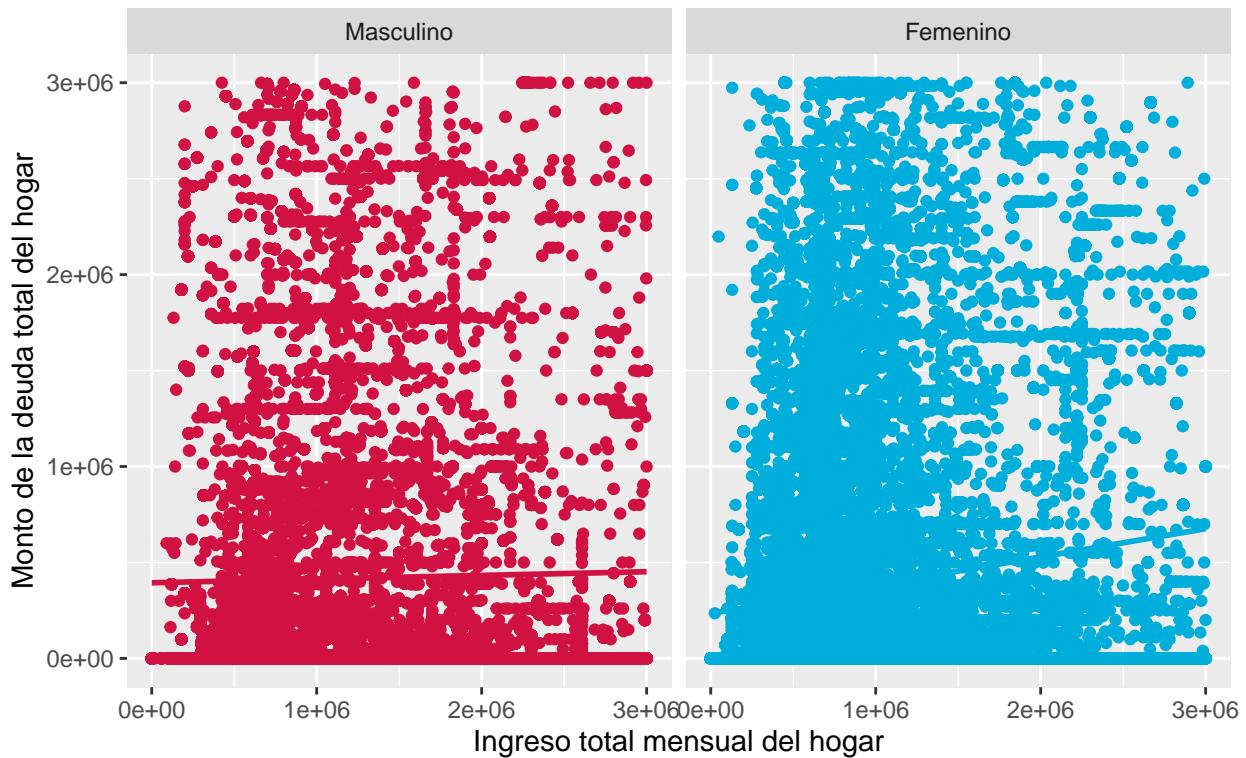
Nuevamente no hay buena visualización ¡pero nada que no se pueda arreglar!. Para ello usaremos el argumento 'grid', este nos separará el gráfico de dispersión para hombres y mujeres.

```

datos %>% filter(ytoth <= 3000000,
                     d_toth <= 3000000) %>%
  plot_scatter(., ytoth, d_toth, genero_ent,
               fit.grps = "lm", grid = TRUE,
               title = "4. Gráfico de dispersión entre ingresos totales y gastos totales, según género")

```

4. Gráfico de dispersión entre ingresos totales y gastos totales, según género

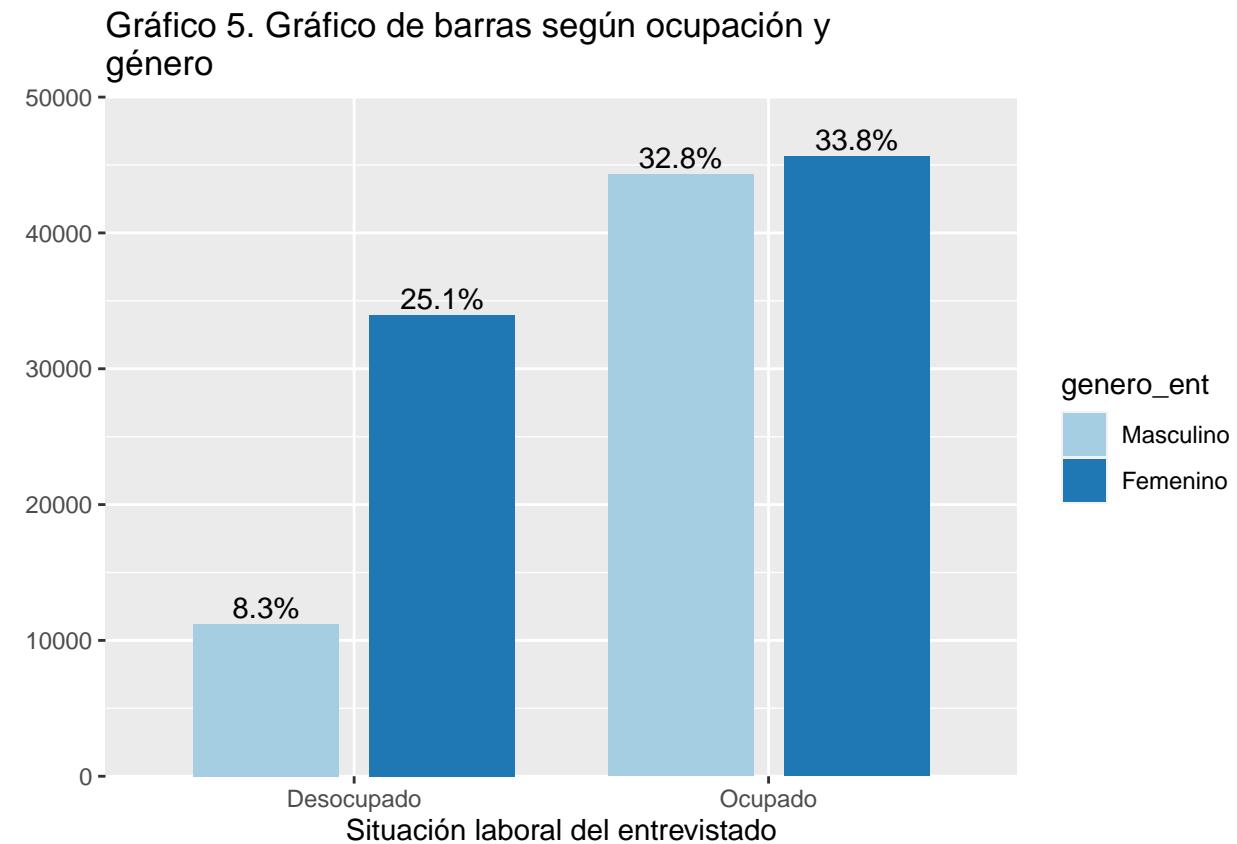


Ahora sólo te queda seguir investigando entre los argumentos de la función ¿qué otras cosas crees que son posibles de graficar?

Gráfico de barras

Efectivamente, con 'sjPlot' puedes crear más gráficos todavía. En esta ocasión usaremos la función 'plot_grpfrq', donde puedes generar barras hasta gráficos de caja, sólo debes explorar en los argumentos. Ahora generaremos un gráfico de barras para las variables ocupación y género.

```
plot_grpfrq(datos$ocup_ent, datos$genero_ent,  
             title = "Gráfico 5. Gráfico de barras según ocupación y género",  
             type = "bar", show.n = FALSE)
```



Regresión

Para generar una regresión utilizaremos la función 'lm' del paquete base, para crear un objeto con el nombre modelo0, este almacenará nuestro modelo.

Pero, ¿cómo lo visualizamos en el reporte? muy fácil, utilizaremos dos paquetes, uno llamado 'broom' y otro 'kableExtra', con el primero generaremos un reporte de los datos de la regresión, mientras que con el segundo incorporaremos un formato específico para la tabla a reportar

```
modelo0 <- lm(ytoth ~ 1,  
                 data = datos,  
                 weights = factor)
```

Table 1: Tabla 1. Modelo sin predictores

term	estimate	std.error	statistic	p.value
(Intercept)	1633805	5355.228	305.086	0

Note:

Elaboración propia

```
broom::tidy(modelo0) %>%
  kable(caption = "Tabla 1. Modelo sin predictores") %>%
  kable_styling(latex_options = c("striped")) %>%
  footnote(general = "Elaboración propia")
```

Table 2: Tabla 2. Modelo con predictores

term	estimate	std.error	statistic	p.value
(Intercept)	1.081695e+06	2.160171e+04	50.0744896	0.0000000
edad_ent	9.133413e+02	2.941627e+02	3.1048852	0.0019039
genero_entFemenino	-4.752229e+05	9.406942e+03	-50.5183171	0.0000000
numh	1.426028e+05	3.526778e+03	40.4343118	0.0000000
d_toth	3.138470e-02	1.555000e-04	201.8526177	0.0000000
t_dttothSI	5.614811e+03	1.027719e+04	0.5463373	0.5848350

Note:

Elaboración propia

```
modelo1 <- lm(ytoth ~ edad_ent + genero_ent + numh + d_toth + t_dttoth,
                 data = datos,
                 weights = factor)
```

```
broom::tidy(modelo1) %>%
  kable(caption = "Tabla 2. Modelo con predictores") %>%
  kable_styling(latex_options = c("striped")) %>%
  footnote(general = "Elaboración propia")
```

Para más información sobre kableExtra, ver aquí

Table 3: Tabla 3. Coeficientes de los predictores

	x
(Intercept)	1.081695e+06
edad_ent	9.133413e+02
genero_entFemenino	-4.752229e+05
numh	1.426028e+05
d_toth	3.138470e-02
t_dtotthSI	5.614811e+03

Note:

Elaboración propia

Información de modelos

En esta ocasión sólo extraeremos los elementos de los objetos creados, para ello usaremos el símbolo \$.

```
modelo1$coefficients %>%
  kable(caption = "Tabla 3. Coeficientes de los predictores") %>%
  kable_styling(latex_options = c("striped")) %>%
  footnote(general = "Elaboración propia")
```

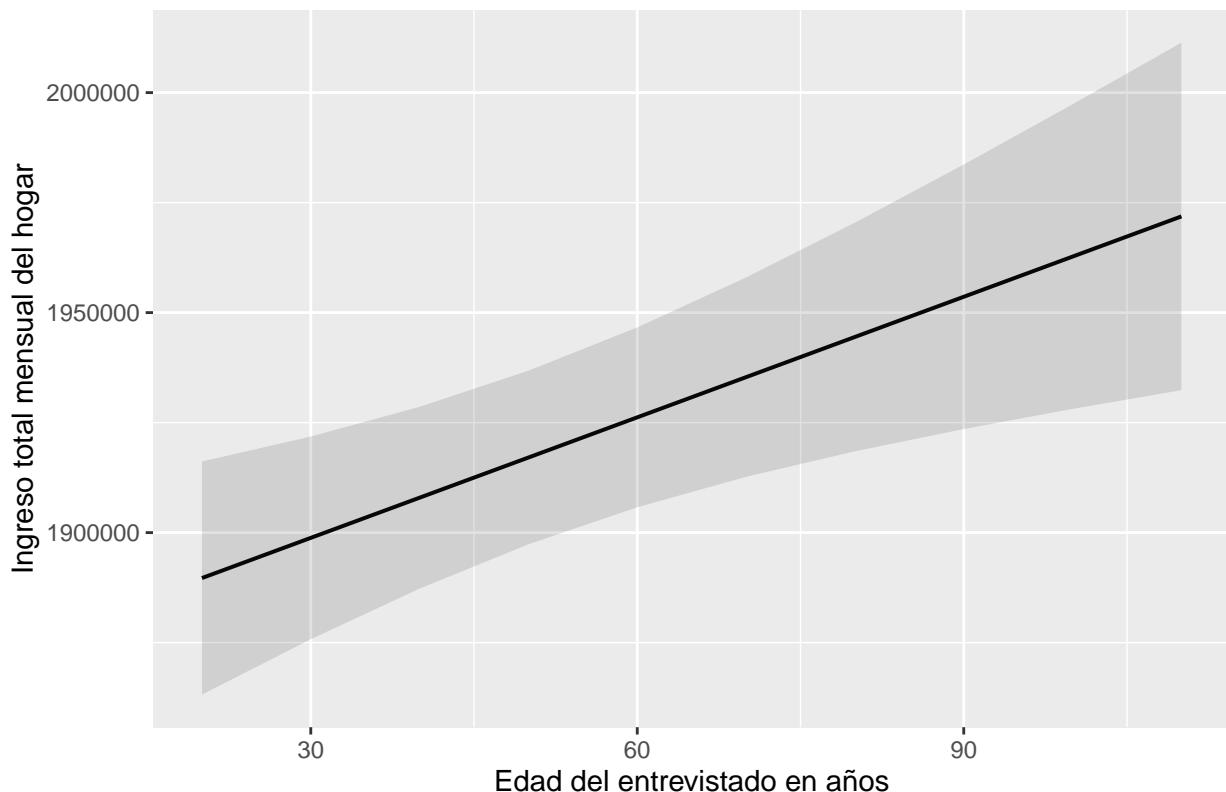
Gráficar los modelos

Nuevamente para la visualización utilizaremos 'sjPlot', con la función 'plot_model' graficaremos el modelo creado.

```
sjPlot::plot_model(modelo1, type = "pred",
                    show.p = T,
                    show.values = T,
                    ci.lvl = c(0.95),
                    title = "Estimación de predictores",
                    vline.color = "purple")
```

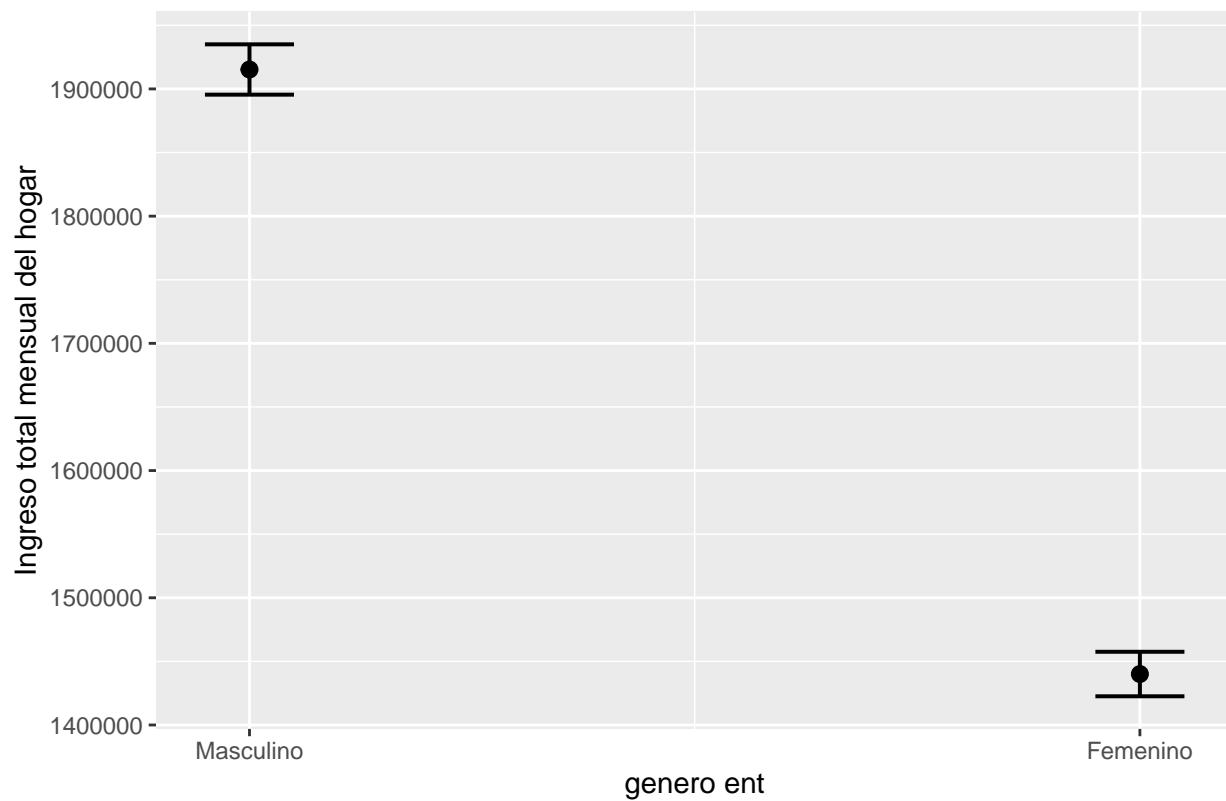
```
## $edad_ent
```

Estimación de predictores



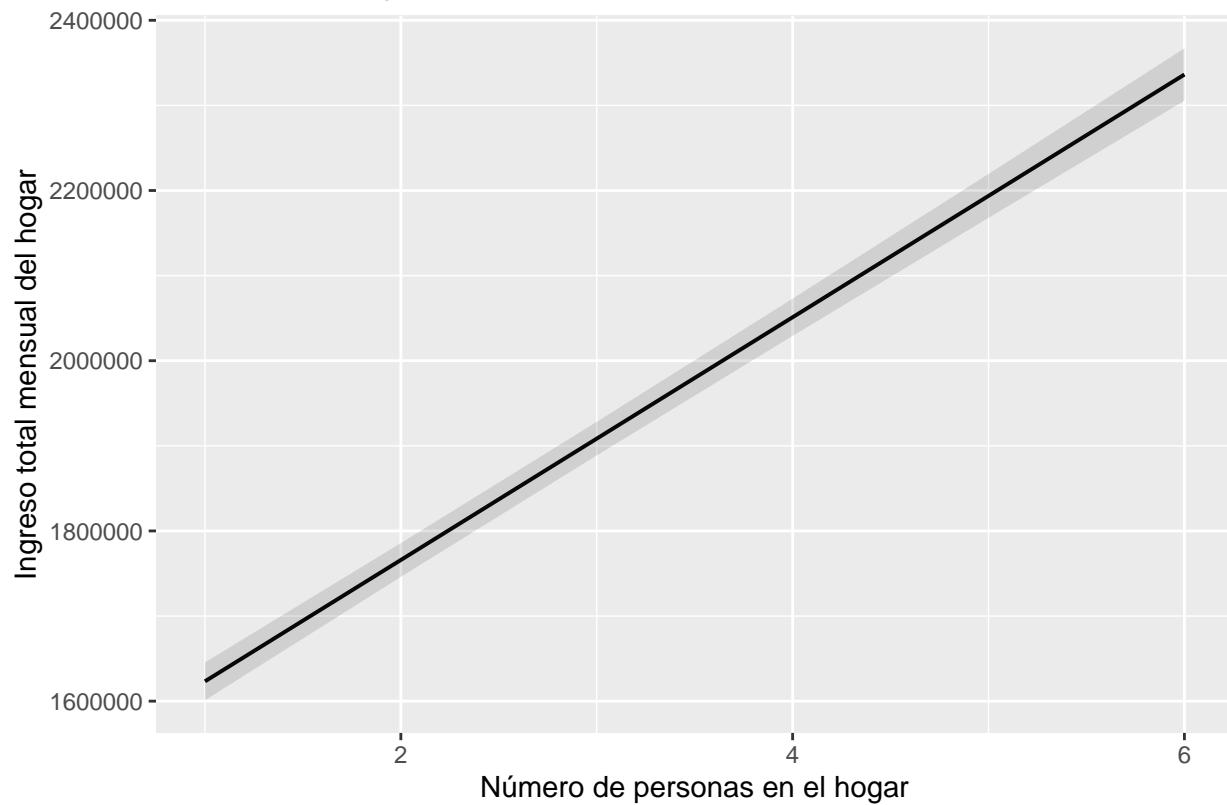
```
##  
## $genero_ent
```

Estimación de predictores



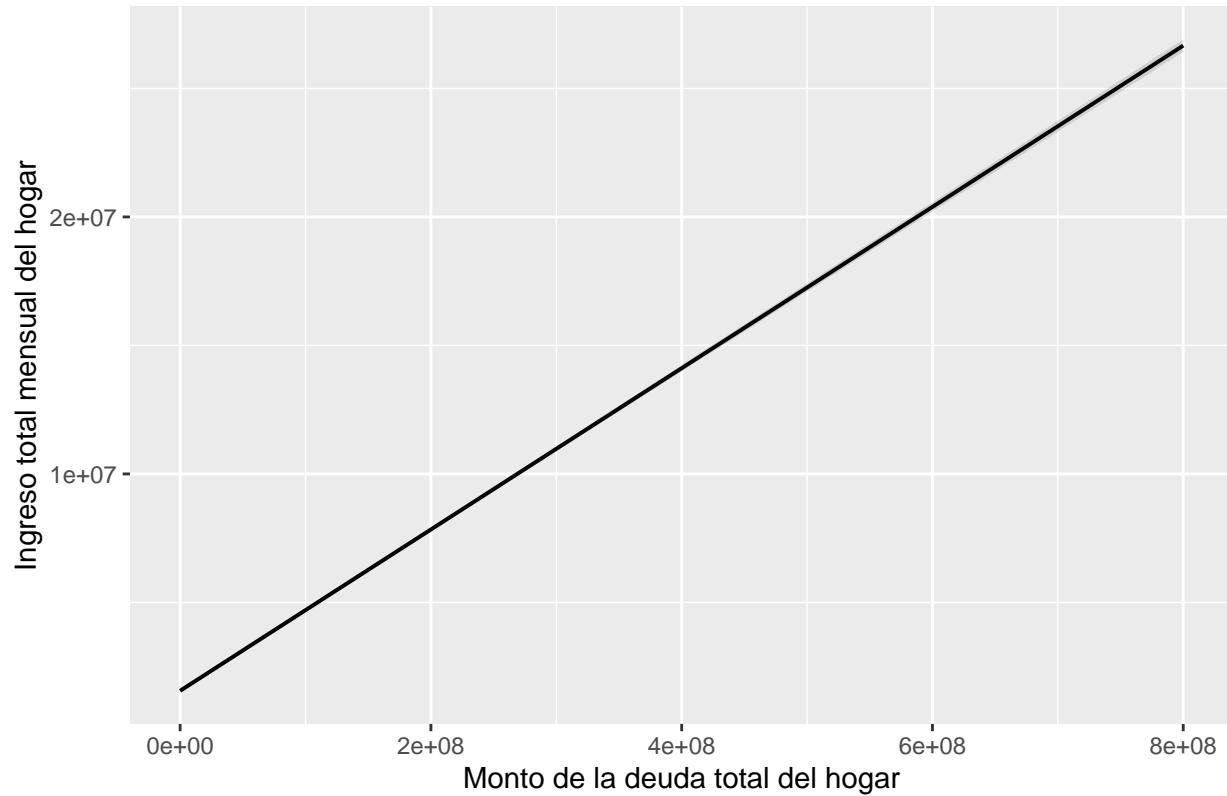
```
##  
## $numh
```

Estimación de predictores



```
##  
## $d_toth
```

Estimación de predictores



```
##  
## $t_dtotth
```

Estimación de predictores

