Práctica 4

Ana Escoto

23/06/2022

Contents

Paquetes

```r
if (!require("pacman")) install.packages("pacman")#instala pacman si se requiere
```

```
## Loading required package: pacman
```

```r
pacman::p_load(tidyverse,
               readxl,
               writexl,
               haven,
               sjlabelled,
               janitor,
               infer,
               ggpubr,
               magrittr,
               gt,
               GGally,
               broom,
               DescTools,
               wesanderson,
               gtsummary,
               srvyr,
               car,
               sjPlot,
               jtools,
               sandwich, huxtable)
```

Cargando los datos

```r
ags_t321 <- read_dta("./datos/AGS_SDEMT321.dta", encoding="latin1") %>%
  clean_names()
```

Hoy sí filtraremos toda nuestra base para quedarnos sólo con algunas variables y casos

```r
ags_t321 %<>%
  filter(r_def==0) %>%
  filter(!c_res==2) %>%
  filter(ing_x_hrs>0) %>%
  filter(clase2==1) %>%
  filter(anios_esc<99)
```

Prueba de hipótesis para la correlación

Una prueba de hipotésis sobe la correlación

```
cor_test<-ags_t321 %>%
    with(
      cor.test(ing_x_hrs,
               anios_esc,
               use = "pairwise")) # prueba de hipótesis.

#dos modos de visualizar el resultado
cor_test
```

```
##
##  Pearson's product-moment correlation
##
## data:  ing_x_hrs and anios_esc
## t = 19.855, df = 3205, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2997756 0.3614258
## sample estimates:
##       cor
## 0.3309538
```

```
tidy(cor_test)
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
0.331	19.9	7.81e-83	3205	0.3	0.361	Pearson's product-moment correlation	two.sided

Modelo simple

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

Donde los parámetros β_o y β_1 describen la pendiente y el intercepto de la población, respectivamente.

No está muy bien comportada, pero ligeramente es mejor con logaritmo

```
ags_t321 %<>%
  mutate(log_ing_x_hrs=log(ing_x_hrs))
```

Una vez transformada nuestra variable, corremos el modelo

```
modelo <- ags_t321 %>%
  with(lm(log_ing_x_hrs~anios_esc))

  summary(modelo) # resultado forma1
```

```
##
## Call:
## lm(formula = log_ing_x_hrs ~ anios_esc)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6773 -0.3277 -0.0305  0.2966  2.9543
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.920324   0.027155  107.54   <2e-16 ***
## anios_esc   0.061355   0.002476   24.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5438 on 3205 degrees of freedom
## Multiple R-squared:  0.1608, Adjusted R-squared:  0.1606
## F-statistic: 614.3 on 1 and 3205 DF,  p-value: < 2.2e-16
```

Con "tidy()"

```
tidy(modelo) # Pruebas de hipótesis de los coeficientes
```

term	estimate	std.error	statistic	p.value
(Intercept)	2.92	0.0272	108	0
anios_esc	0.0614	0.00248	24.8	3.21e-124

Para obtener los intervalos de confianza, podemos hacerlo a partir del siguiente comando:

```
confint(modelo)
```

```
##                  2.5 %     97.5 %
## (Intercept) 2.86708152 2.97356747
## anios_esc   0.05650163 0.06620911
```

Para el ajuste global del modelo, podemos utilzar el comando "glance()" sobre el objeto de nuestro modelo, ello nos dará la información correspondiente:

```
glance(modelo) # resultado ajuste global
```

ared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual
.161	0.161	0.544	614	3.21e-124	1	-2.6e+03	5.2e+03	5.22e+03	948	3205

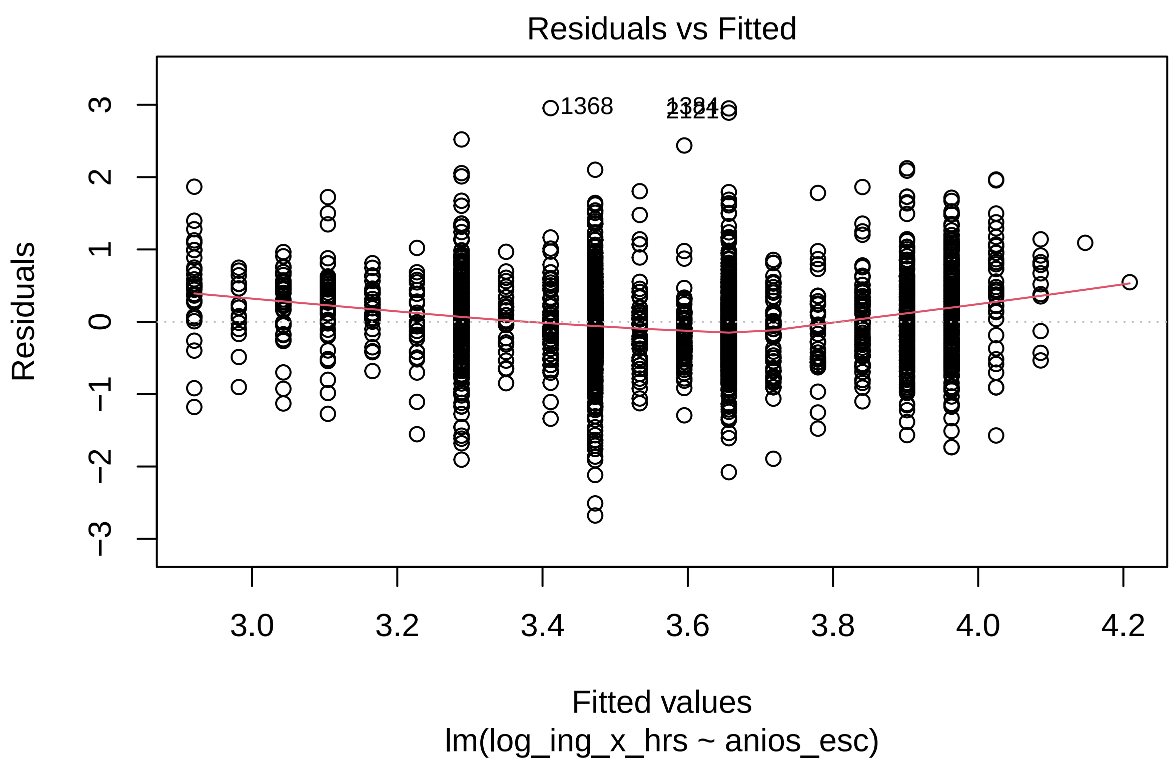
Otra manera de ver este ajuste es con el comando "anova()":

```
anova(modelo)
```

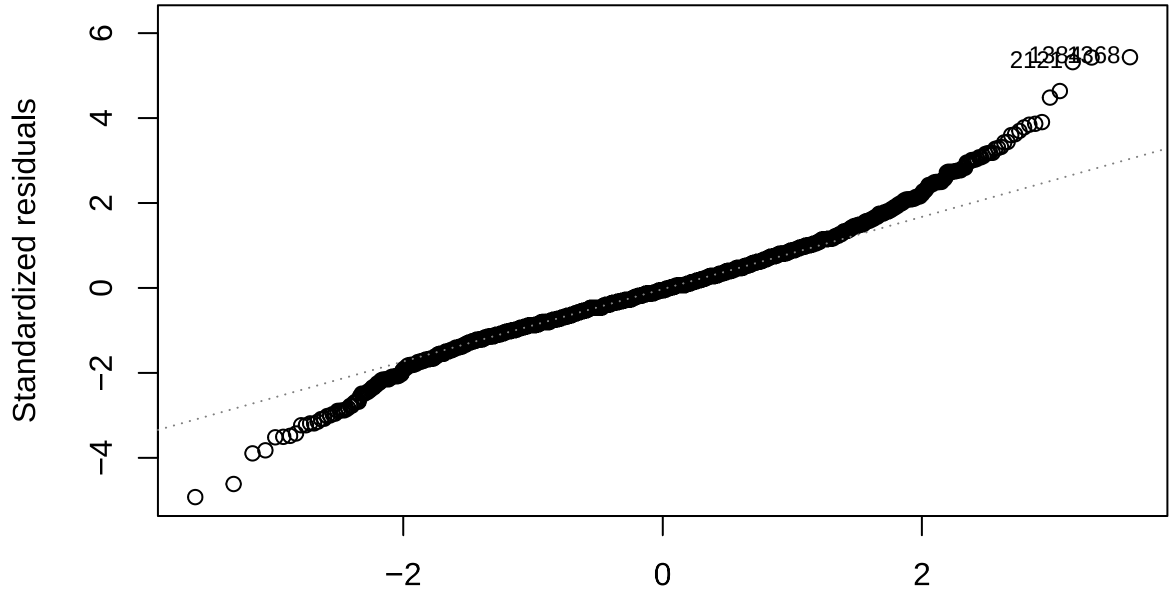
Diagnósticos

Df	Sum Sq	Mean Sq	F value	Pr(>F)
1	182	182	614	3.21e-124
3205	948	0.296		

```
plot(modelo)
```

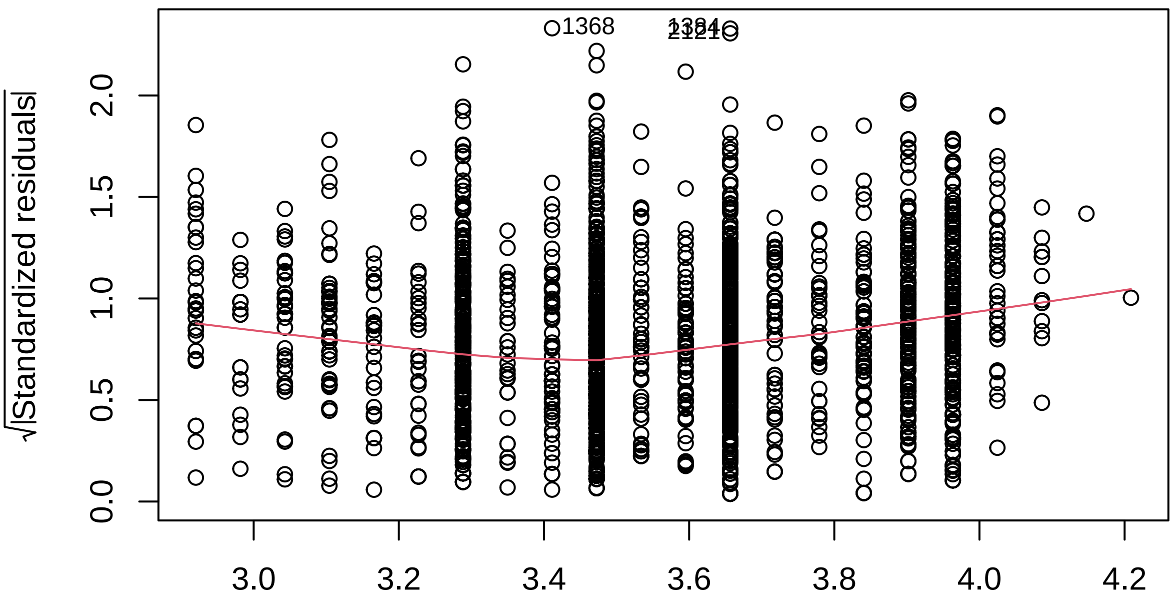


Normal Q−Q

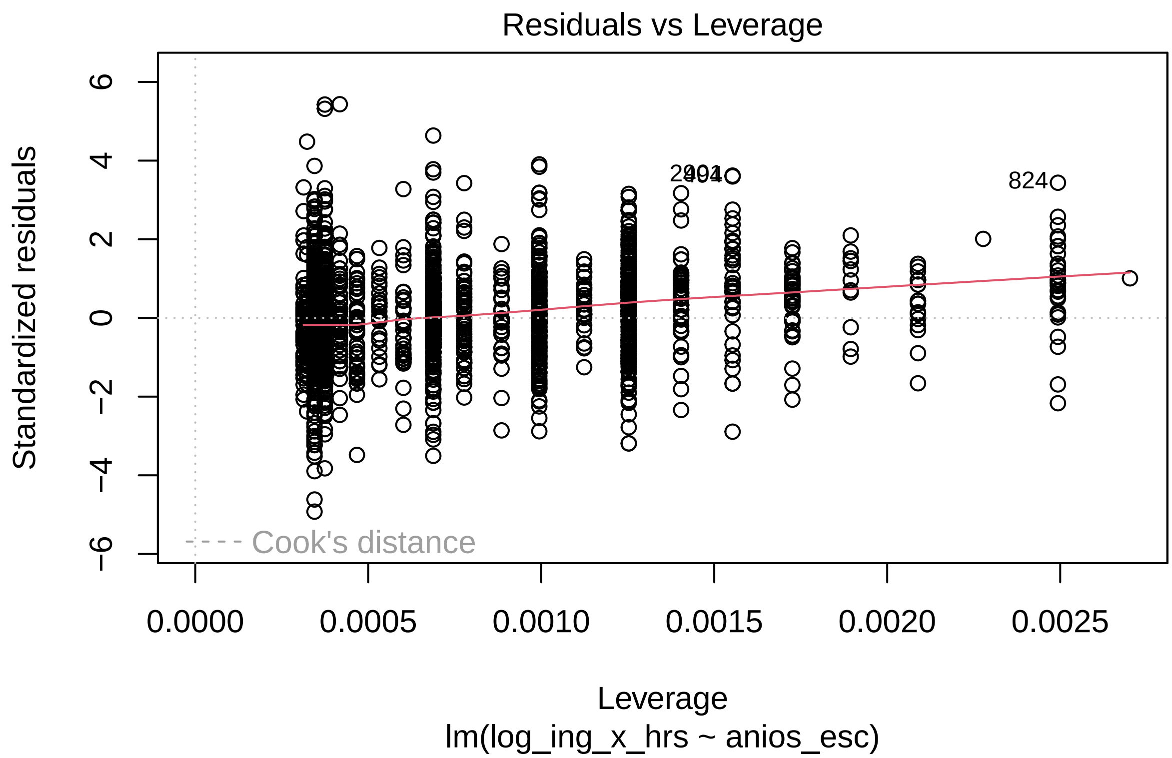


Theoretical Quantiles
lm(log_ing_x_hrs ~ anios_esc)

Scale−Location



Fitted values
lm(log_ing_x_hrs ~ anios_esc)

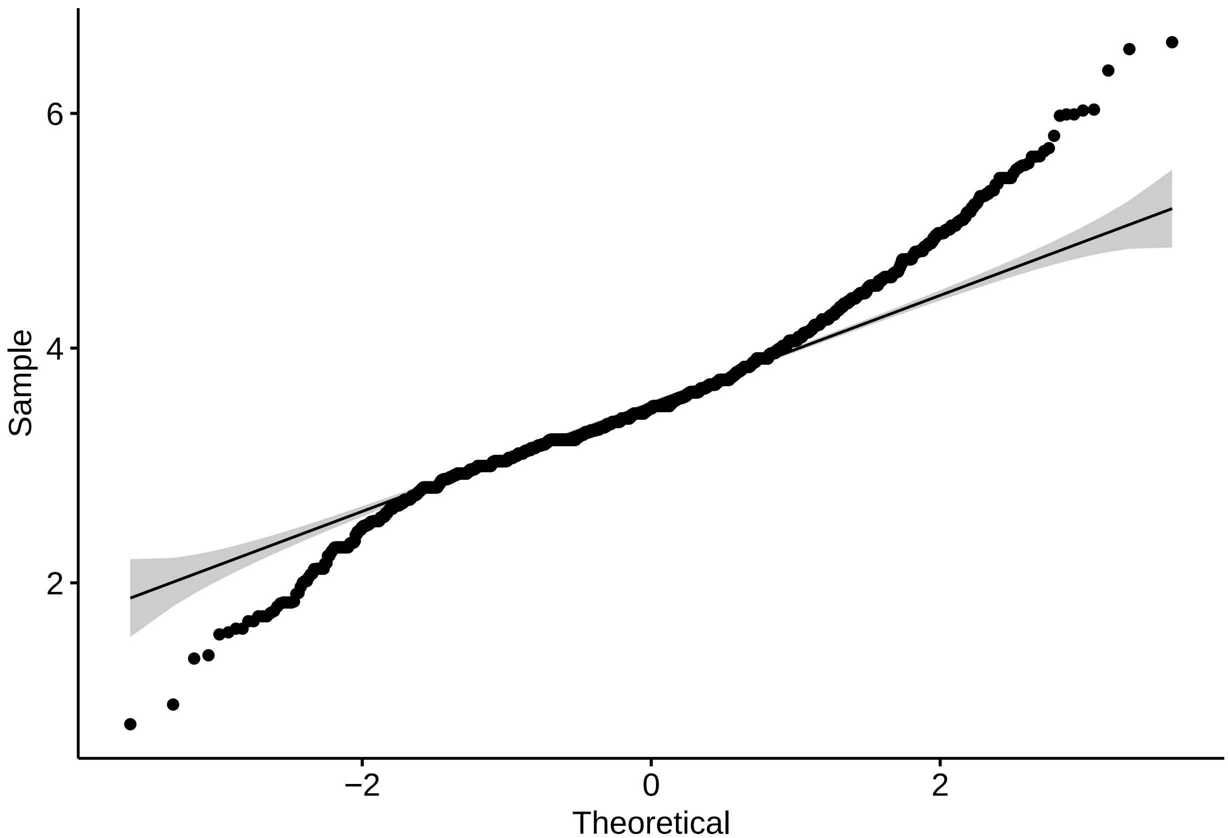


##1. Outliers y Normalidad

```r
# Assessing Outliers
car::outlierTest(modelo) # Bonferonni p-value for most extreme obs
```

```
##        rstudent unadjusted p-value Bonferroni p
## 1368   5.457985         5.1824e-08   0.00016620
## 1384   5.450004         5.4182e-08   0.00017376
## 2121   5.340243         9.9300e-08   0.00031845
## 236   -4.942090         8.1256e-07   0.00260590
## 2781   4.651710         3.4252e-06   0.01098500
## 1987  -4.630520         3.7925e-06   0.01216300
## 2853   4.495555         7.1848e-06   0.02304200
```

```r
ggpubr::ggqqplot(ags_t321$log_ing_x_hrs)
```

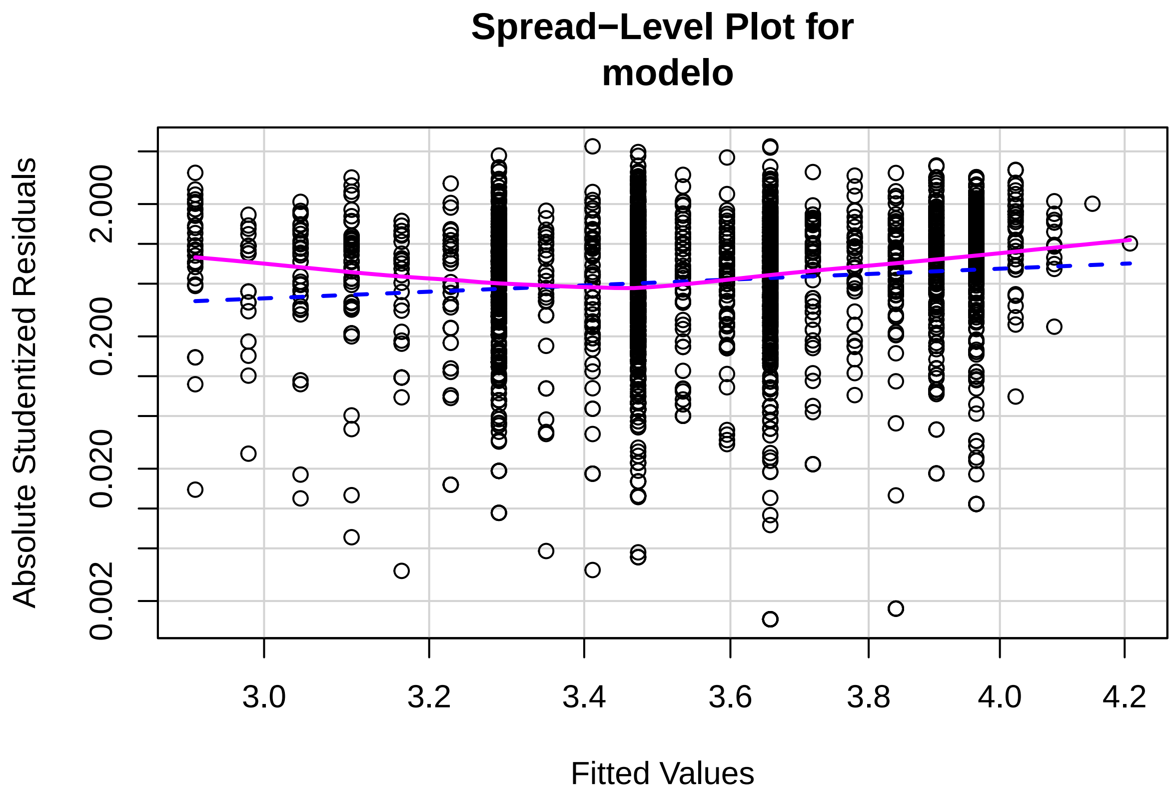


##2. Homocedasticidad

```
# non-constant error variance test
car::ncvTest(modelo)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 41.14906, Df = 1, p = 1.4105e-10
```

```
# plot studentized residuals vs. fitted values
car::spreadLevelPlot(modelo)
```




```
##
## Suggested power transformation:  -0.793166
```

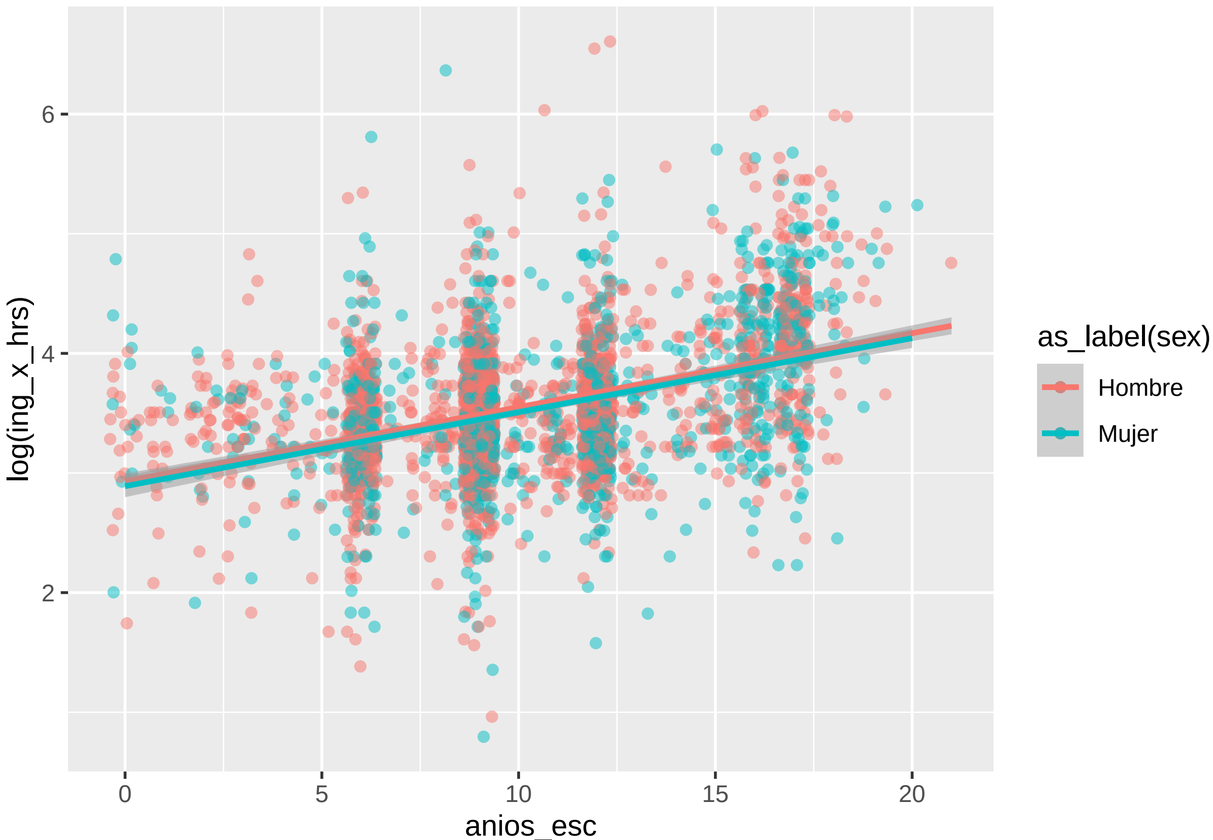
Regresión Lineal múltiple

Agregando una variable categórica

¿Es igual la relación entre hombres y mujeres con los ingresos y la escolaridad?

```
ags_t321 %>%
  ggplot() +
    aes(x=anios_esc, y=log(ing_x_hrs), alpha=I(0.5), color=as_label(sex)) +
  geom_jitter()+
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Cuando nosotros tenemos una variable categórica para la condición de sexo. [nota: seguimos haciendo el ejercicio, a pesar de que ya observamos en nuestro diagnóstico el modelo no cumple con los supuestos, pero lo haremos para fines ilustrativos]

```
modelo1<-ags_t321 %>%
  mutate(sex=as_label(sex)) %>%
  with(
    lm(log_ing_x_hrs ~anios_esc + sex)
  )

summary(modelo1)
```

```
##
## Call:
## lm(formula = log_ing_x_hrs ~ anios_esc + sex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.65148 -0.32711 -0.02543  0.29691  2.98062
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.931726   0.027671 105.951   <2e-16 ***
## anios_esc    0.061849   0.002485  24.887   <2e-16 ***
## sexMujer    -0.041708   0.019722  -2.115   0.0345 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.5435 on 3204 degrees of freedom
## Multiple R-squared:  0.162,  Adjusted R-squared:  0.1615
## F-statistic: 309.7 on 2 and 3204 DF,  p-value: < 2.2e-16
```

Este modelo tiene coeficientes que deben leerse "condicionados". Es decir, en este caso tenemos que el coeficiente asociado a la edad, mantiene constante el valor de sexo y viceversa.

¿Cómo saber is ha mejorado nuestro modelo? Podemos comparar el ajuste con la anova, es decir, una prueba F

```
pruebaf0<-anova(modelo, modelo1)
pruebaf0
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
3.20e+03	948				
3.2e+03	947	1	1.32	4.47	0.0345

Como puedes ver, el resultado muestra un Df de 1 (lo que indica que el modelo más complejo tiene un parámetro adicional) y un valor p muy pequeño (<.51). Esto significa que agregar el sexo al modelo lleva a un ajuste significativamente mejor sobre el modelo original.

Podemos seguir añadiendo variables sólo "sumando" en la función

```
modelo2<- ags_t321 %>%
  mutate(sex=as_label(sex)) %>%
  with(
    lm(log_ing_x_hrs ~ anios_esc + sex + eda)
    )
summary(modelo2)
```

```
##
## Call:
## lm(formula = log_ing_x_hrs ~ anios_esc + sex + eda)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67314 -0.31328 -0.01734  0.29021  3.12636
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.6076560  0.0423250  61.610  < 2e-16 ***
## anios_esc    0.0682538  0.0025299  26.979  < 2e-16 ***
## sexMujer    -0.0546051  0.0194668  -2.805  0.00506 **
## eda          0.0069996  0.0006995  10.007  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5353 on 3203 degrees of freedom
## Multiple R-squared:  0.1874, Adjusted R-squared:  0.1867
## F-statistic: 246.2 on 3 and 3203 DF,  p-value: < 2.2e-16
```

Y podemos ver si introducir esta variable afectó al ajuste global del modelo

```
pruebaf1<-anova(modelo1, modelo2)
pruebaf1
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
3.2e+03	947				
3.2e+03	918	1	28.7	100	3.1e-23

Hoy que tenemos más variables podemos hablar de revisar dos supuestos más.

Otros supuestos

Además de los supuestos de la regresión simple, podemos revisar estos otros. De nuevo, usaremos la librería "car",

1. Linealidad en los parámetros (será más díficil entre más variables tengamos)
2. La normalidad también, porque debe ser multivariada
3. Multicolinealidad La prueba más común es la de Factor Influyente de la Varianza (VIF) por sus siglas en inglés. La lógica es que la multicolinealidad tendrá efectos en nuestro R2, inflándolo. De ahí que observamos de qué variable(s) proviene este problema relacionado con la multicolinealidad.

Si el valor es mayor a 5, tenemos un problema muy grave.

```
car::vif(modelo2)
```

```
## anios_esc       sex       eda
##  1.077886  1.013336  1.070145
```

Jtools

Un solo modelo:

```
jtools::summ(modelo)
```

Observations	3207
Dependent variable	log_ing_x_hrs
Type	OLS linear regression

F(1,3205)	614.29
R^2	0.16
Adj. R^2	0.16

Si queremos errores robusto, estilo *STATA*:

	Est.	S.E.	t val.	p
(Intercept)	2.92	0.03	107.54	0.00
anios_esc	0.06	0.00	24.78	0.00

Standard errors: OLS

```
summ(modelo2,  robust = "HC1")
```

Observations	3207
Dependent variable	log_ing_x_hrs
Type	OLS linear regression

F(3,3203)	246.25
R^2	0.19
Adj. R^2	0.19

	Est.	S.E.	t val.	p
(Intercept)	2.61	0.05	57.74	0.00
anios_esc	0.07	0.00	23.54	0.00
sexMujer	-0.05	0.02	-2.75	0.01
eda	0.01	0.00	9.29	0.00

Standard errors: Robust, type = HC1

Si queremos estandarizar nuestras escalas:

```
summ(modelo2,  scale=T)
```

Observations	3207
Dependent variable	log_ing_x_hrs
Type	OLS linear regression

F(3,3203)	246.25
R^2	0.19
Adj. R^2	0.19

	Est.	S.E.	t val.	p
(Intercept)	3.57	0.01	293.17	0.00
anios_esc	0.26	0.01	26.98	0.00
sexMujer	-0.05	0.02	-2.81	0.01
eda	0.10	0.01	10.01	0.00

Standard errors: OLS; Continuous predictors are mean-centered and scaled by 1 s.d.

También se pueden comparar modelos:

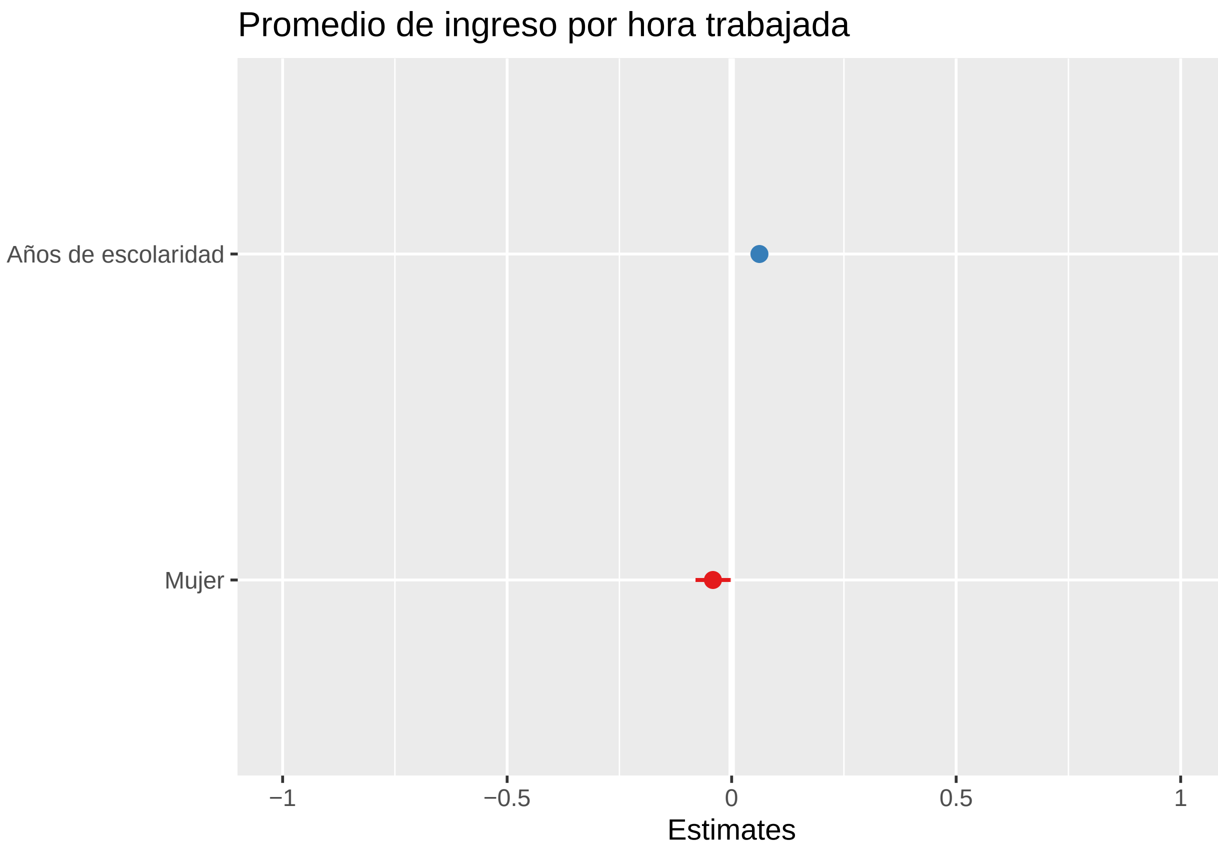
```
export_summs(modelo, modelo1, modelo2)
```

	Model 1	Model 2	Model 3
(Intercept)	2.92 ***	2.93 ***	2.61 ***
	(0.03)	(0.03)	(0.04)
anios_esc	0.06 ***	0.06 ***	0.07 ***
	(0.00)	(0.00)	(0.00)
sexMujer		-0.04 *	-0.05 **
		(0.02)	(0.02)
eda			0.01 ***
			(0.00)
N	3207	3207	3207
R2	0.16	0.16	0.19

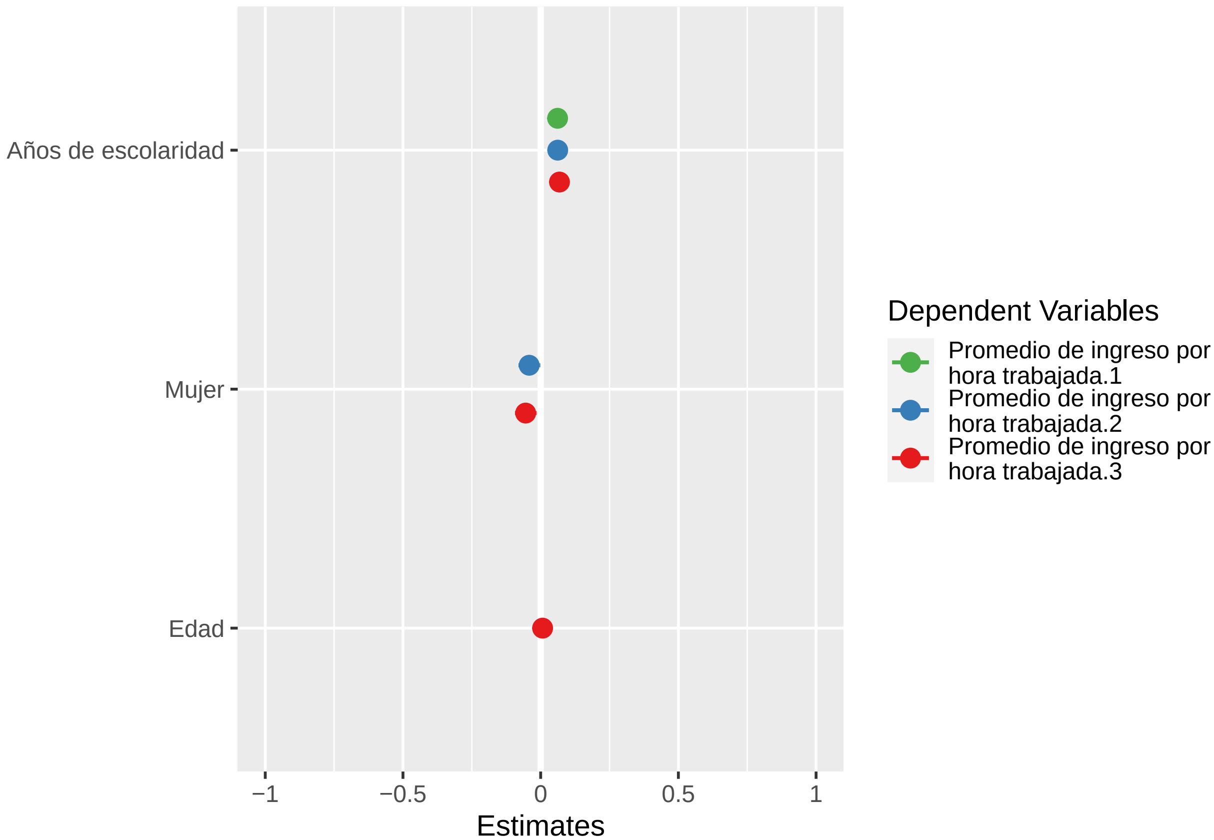
*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

También el paquete "sjPlot" tiene el comando "plot_model()"

```
sjPlot::plot_model(modelo1)
```




```
sjPlot::plot_models(modelo, modelo1, modelo2)
```

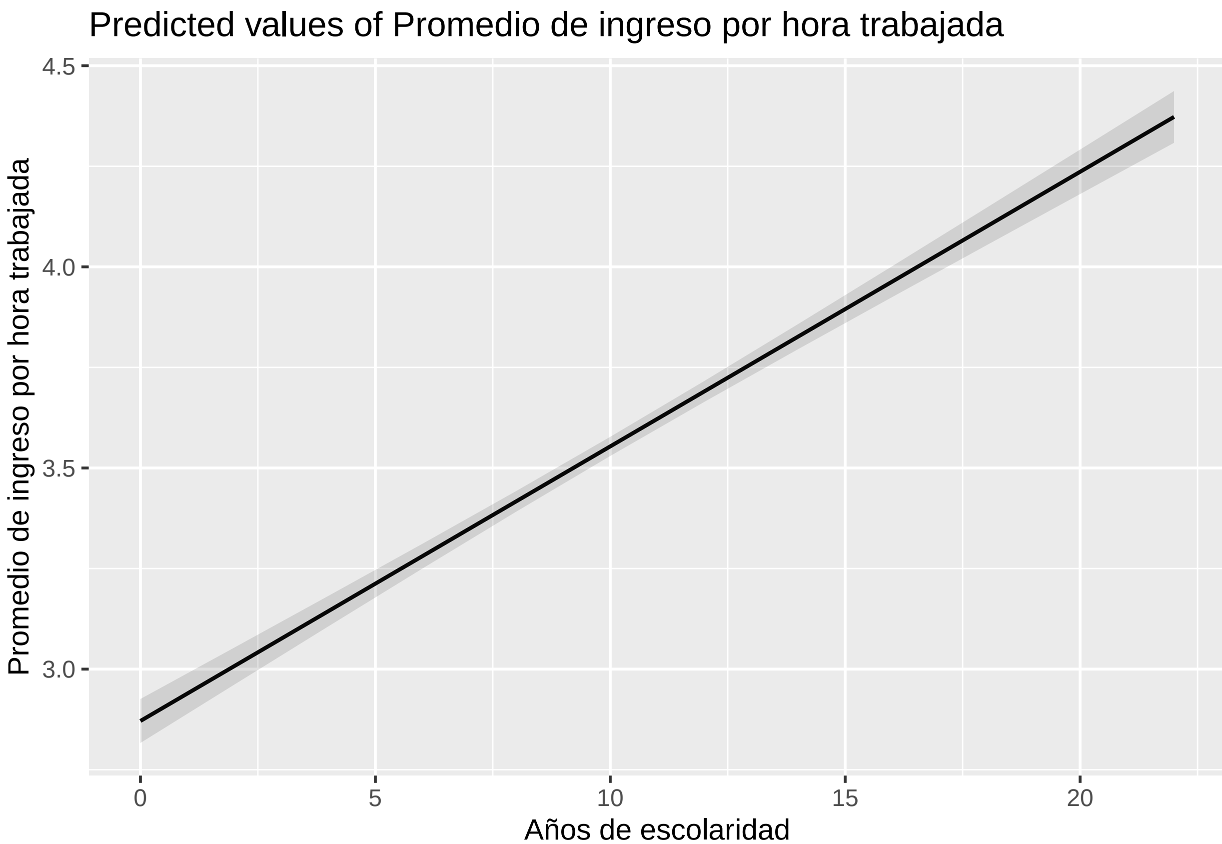


Post-estimación

Las predicciones

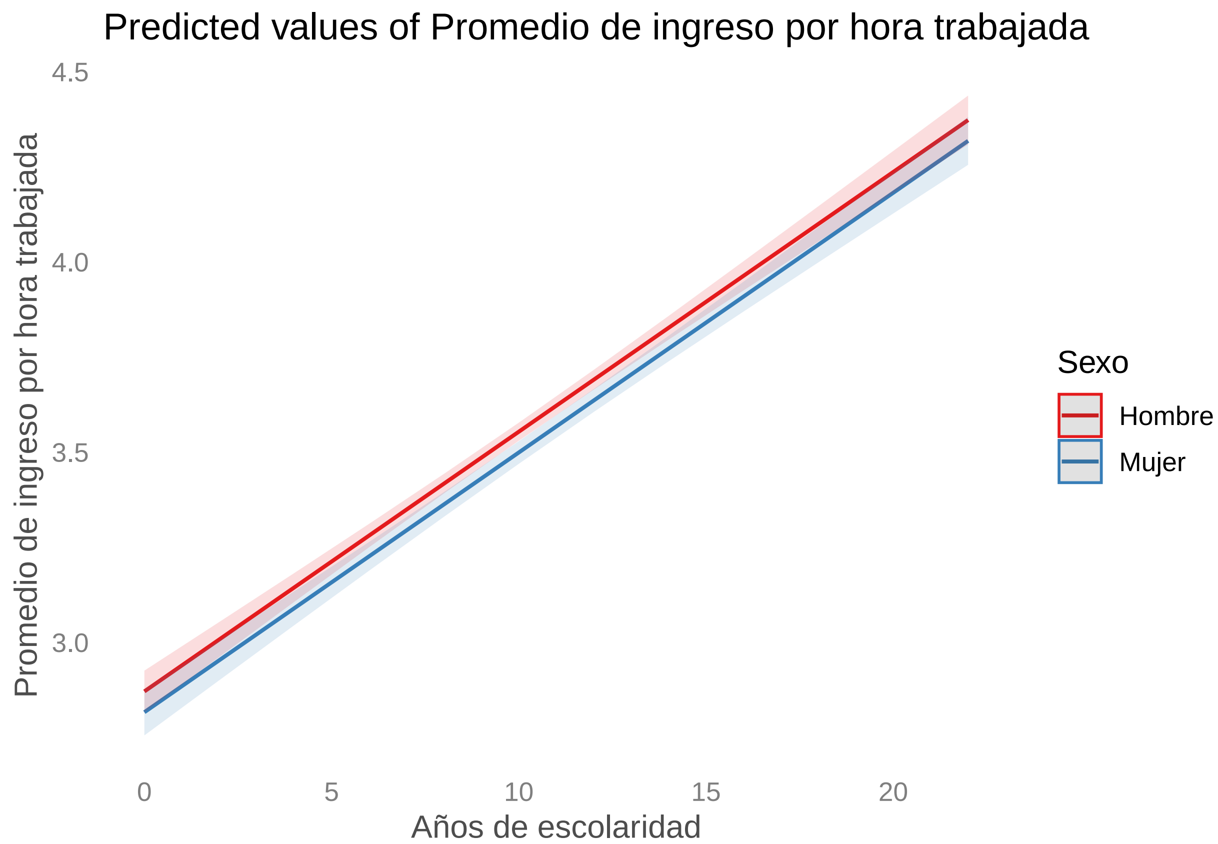
Unos de los usos más comunes de los modelos estadísticos es la predicción

```
sjPlot::plot_model(modelo2, type="pred", terms = "anios_esc")
```



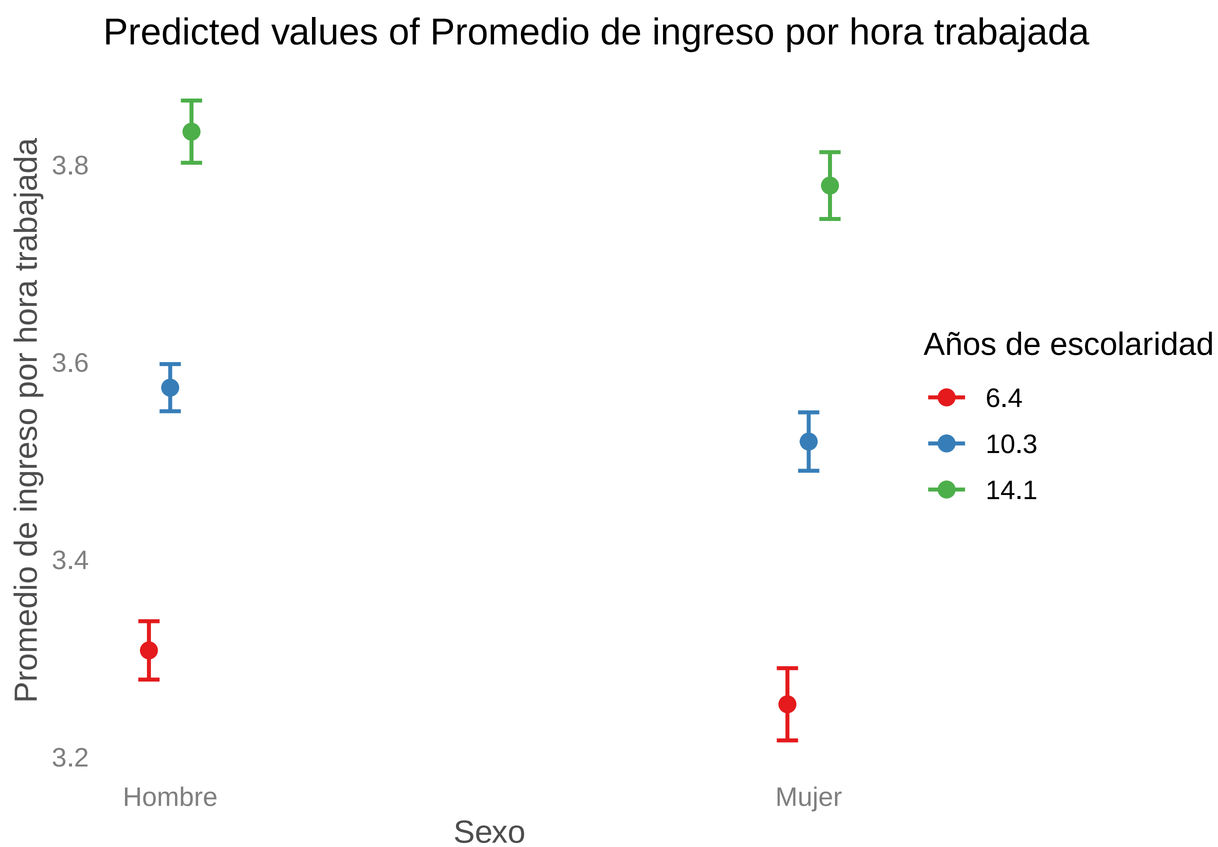
También podemos incluir la predecciones para los distintos valores de las variables

```
plot_model(modelo2, type="pred", terms = c("anios_esc","sex")) + theme_blank()
```



El orden de los términos importa:

```
plot_model(modelo2, type="pred", terms = c("sex","anios_esc")) + theme_blank()
```

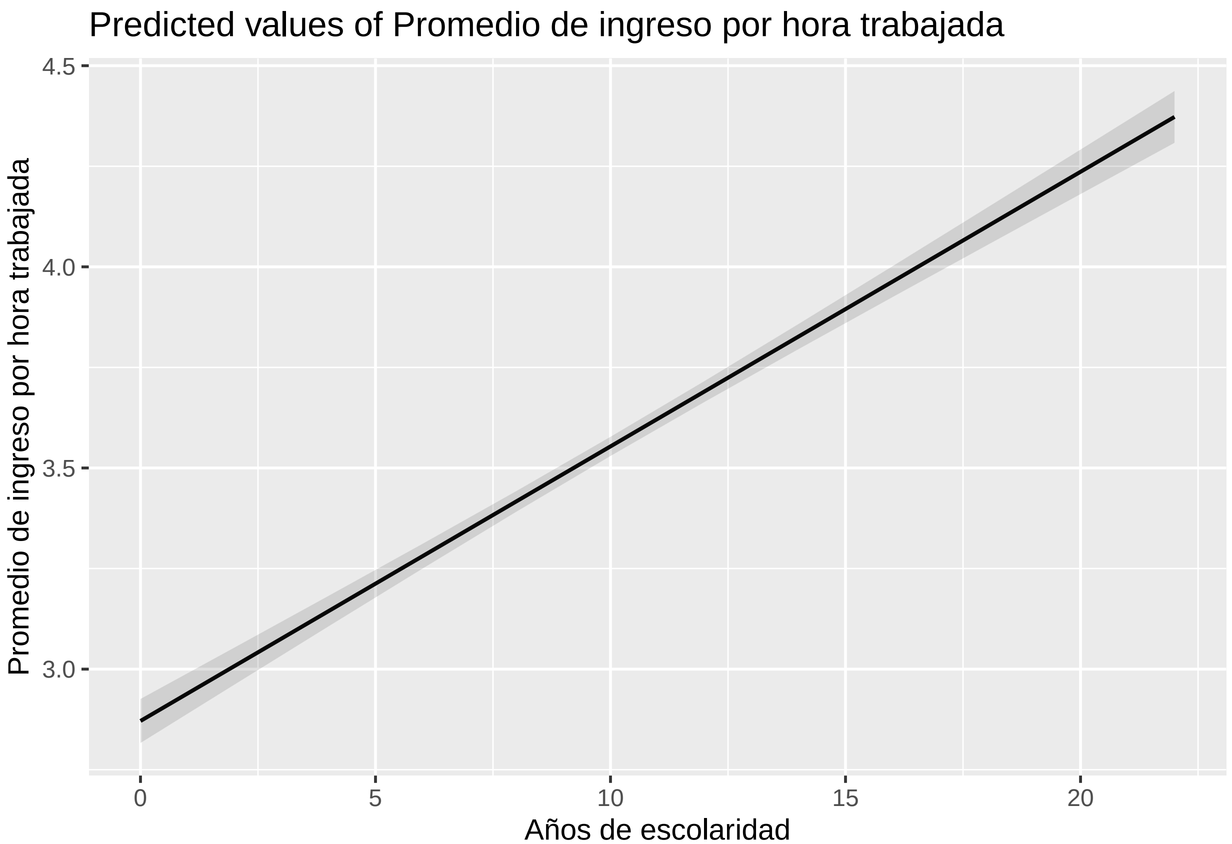


Efectos marginales

Con los efectos marginales, por otro lado medimos el efecto promedio, dejando el resto de variables constantes.

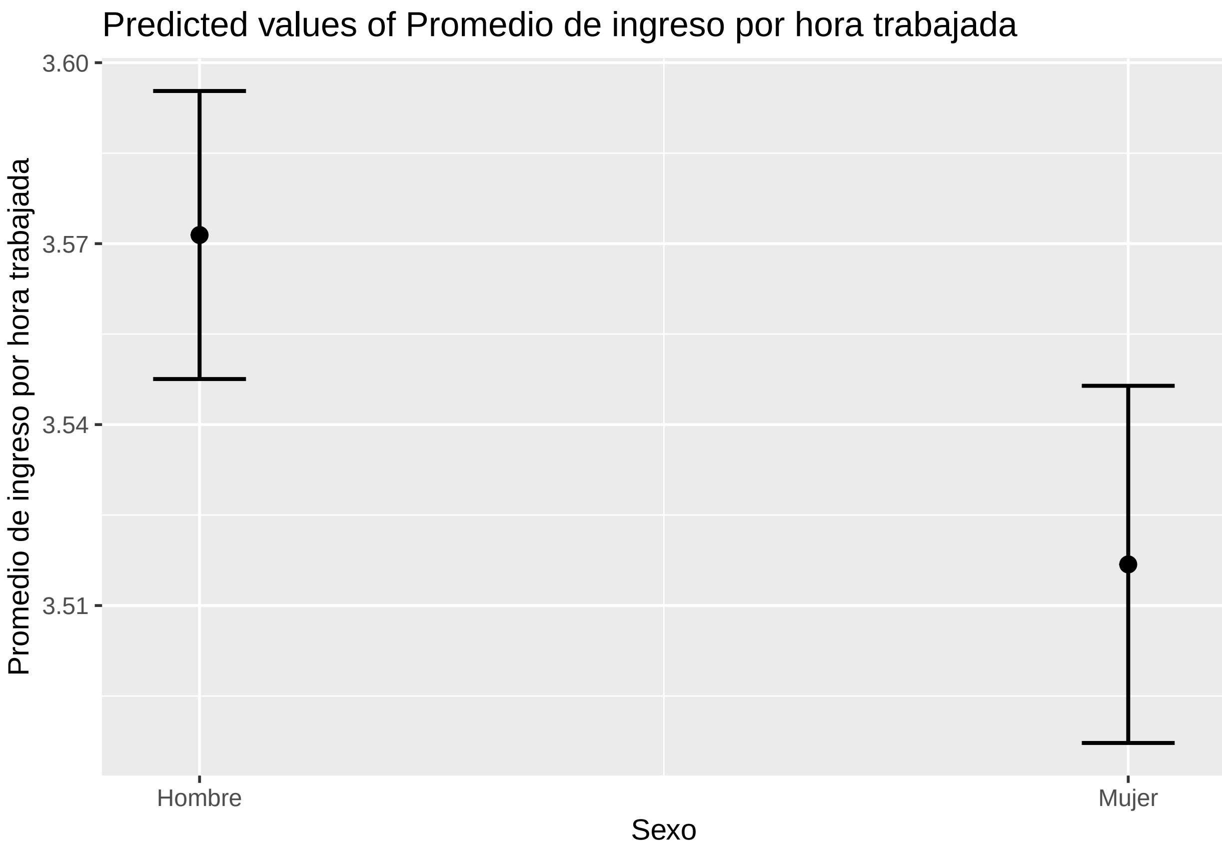
```
plot_model(modelo2, type="eff", terms = "anios_esc")
```

```
## Package `effects` is not available, but needed for `ggeffect()`. Either install package `effects`, o
```




```
plot_model(modelo2, type="eff", terms = "sex")
```

```
## Package `effects` is not available, but needed for `ggeffect()`. Either install package `effects`, o
```



¿Es el mismo gráfico que con "pred"? Veamos la ayuda

¿Y si queremos ver esta informaicón graficada?

```
eff<-plot_model(modelo2, type="eff", terms = "anios_esc")
```

```
## Package `effects` is not available, but needed for `ggeffect()`. Either install package `effects`, o
```

```
eff$data
```

```
eff<-plot_model(modelo2, type="pred", terms = "anios_esc")
eff$data
```

Extensiones del modelo de regresión

Introducción a las interacciones

Muchas veces las variables explicativas van a tener relación entre sí. Por ejemplo ¿Las horas tendrá que ver con el sexo y afectan no sólo en intercepto si no también la pendiente? Para ello podemos introducir una interacción

```
modelo_int1<-lm(log_ing_x_hrs ~ anios_esc * sex , data = ags_t321, na.action=na.exclude)
summary(modelo_int1)
```

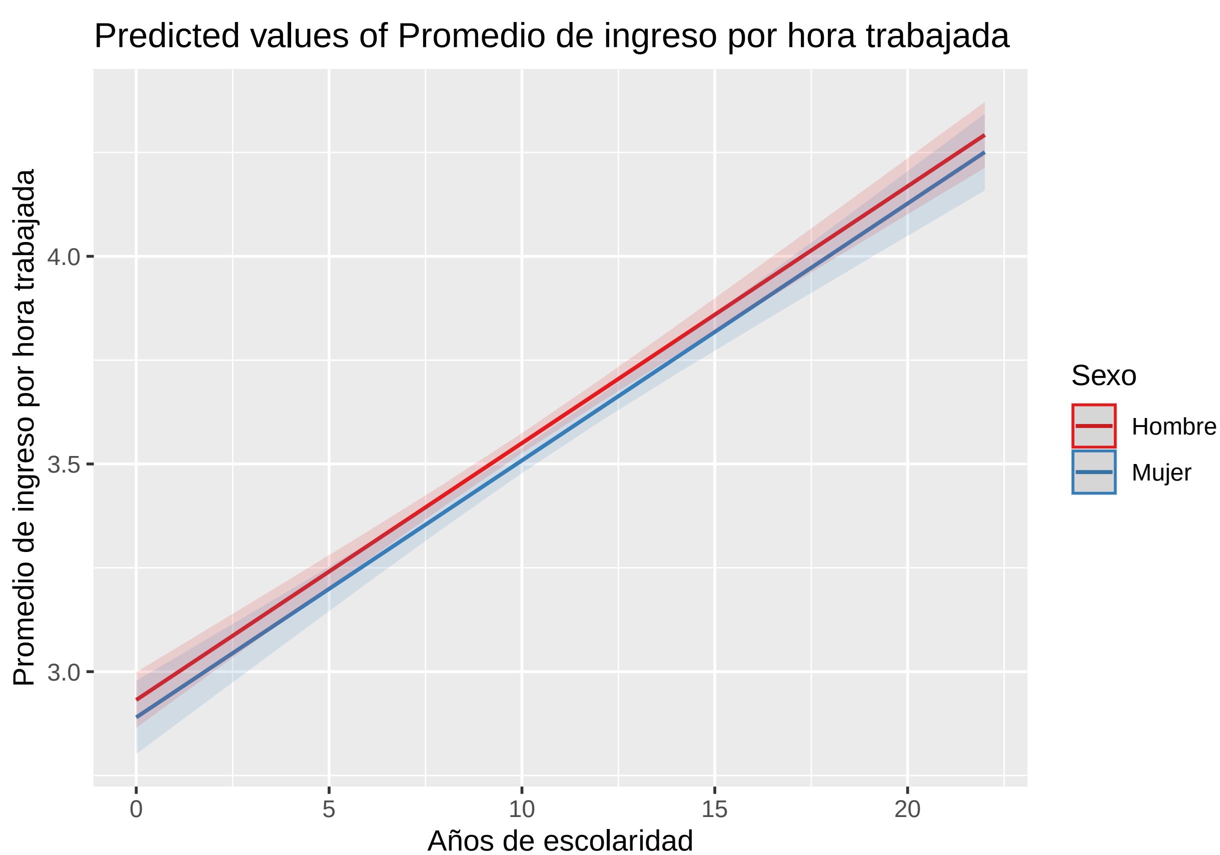
x	predicted	std.error	conf.low	conf.high	group	group_col
0	2.87	0.0279	2.82	2.93	1	1
2	3.01	0.0235	2.96	3.05	1	1
4	3.14	0.0193	3.11	3.18	1	1
6	3.28	0.0157	3.25	3.31	1	1
8	3.42	0.0131	3.39	3.44	1	1
10	3.55	0.0122	3.53	3.58	1	1
12	3.69	0.0132	3.66	3.72	1	1
14	3.83	0.0159	3.8	3.86	1	1
16	3.96	0.0196	3.92	4	1	1
18	4.1	0.0238	4.05	4.15	1	1
20	4.24	0.0282	4.18	4.29	1	1
22	4.37	0.0329	4.31	4.44	1	1

```
##
## Call:
## lm(formula = log_ing_x_hrs ~ anios_esc * sex, data = ags_t321,
##     na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.65146 -0.32715 -0.02537  0.29686  2.98066
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.974e+00  8.185e-02  36.333  < 2e-16 ***
## anios_esc     6.182e-02  7.520e-03   8.221 2.89e-16 ***
## sex          -4.192e-02  5.647e-02  -0.742    0.458
## anios_esc:sex  2.016e-05  5.080e-03   0.004    0.997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5436 on 3203 degrees of freedom
## Multiple R-squared:  0.162,  Adjusted R-squared:  0.1612
## F-statistic: 206.4 on 3 and 3203 DF,  p-value: < 2.2e-16
```

Esta interacción lo que asume es que las pendientes pueden moverse (aunque en este caso específico no lo hacen tanto porque no nos salió significativa)

x	predicted	std.error	conf.low	conf.high	group	group_col
0	2.87	0.0279	2.82	2.93	1	1
2	3.01	0.0235	2.96	3.05	1	1
4	3.14	0.0193	3.11	3.18	1	1
6	3.28	0.0157	3.25	3.31	1	1
8	3.42	0.0131	3.39	3.44	1	1
10	3.55	0.0122	3.53	3.58	1	1
12	3.69	0.0132	3.66	3.72	1	1
14	3.83	0.0159	3.8	3.86	1	1
16	3.96	0.0196	3.92	4	1	1
18	4.1	0.0238	4.05	4.15	1	1
20	4.24	0.0282	4.18	4.29	1	1
22	4.37	0.0329	4.31	4.44	1	1

```
plot_model(modelo_int1, type="int", terms = c("sex", "anios_esc"))
```



Efectos no lineales

Explicitando el logaritmo

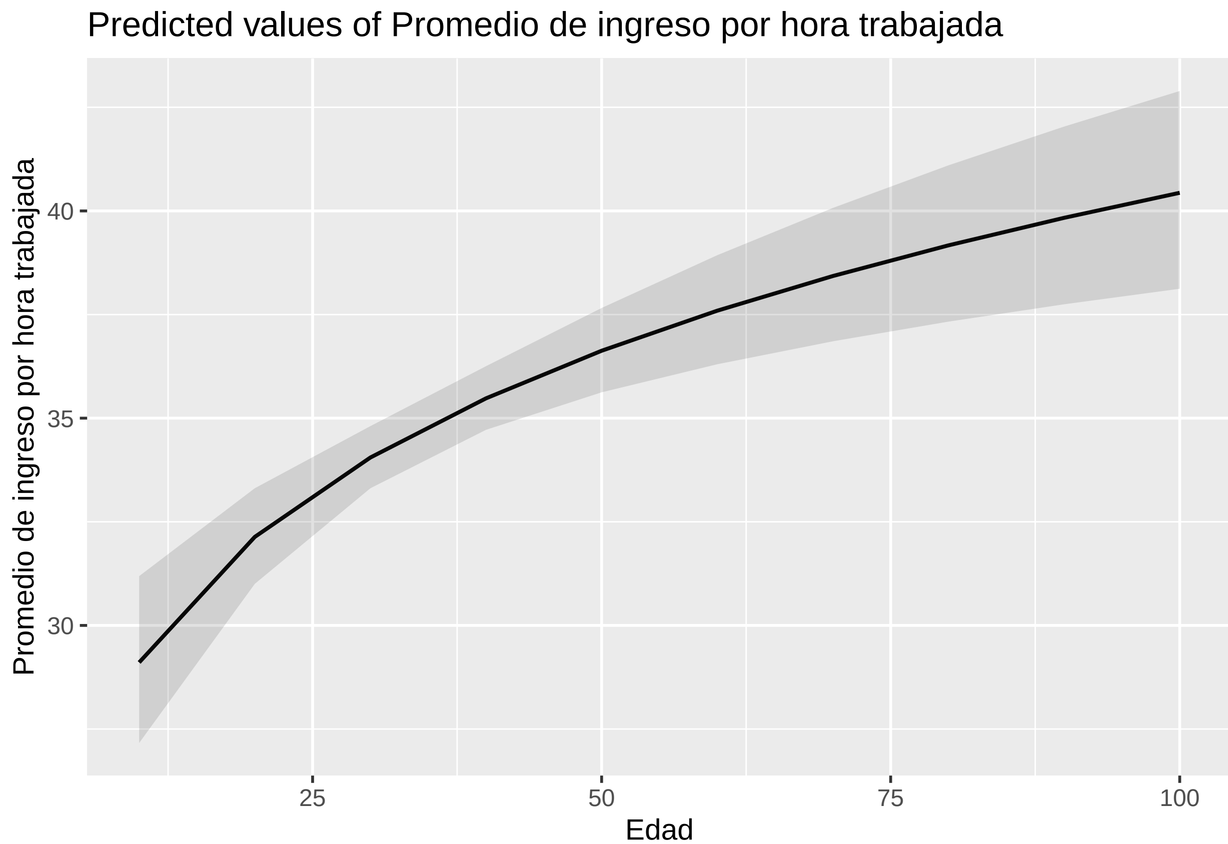
```
modelo_log<-ags_t321 %>%
  with(
    lm(log(ing_x_hrs) ~ log(eda) + sex))

summary(modelo_log)
```

```
##
## Call:
## lm(formula = log(ing_x_hrs) ~ log(eda) + sex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.78318 -0.35441 -0.07327  0.29257  3.02918
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.044232   0.099191  30.691  < 2e-16 ***
## log(eda)     0.142789   0.026880   5.312 1.16e-07 ***
## sex         -0.001465   0.021384  -0.069    0.945
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5911 on 3204 degrees of freedom
## Multiple R-squared:  0.008743,   Adjusted R-squared:  0.008125
## F-statistic: 14.13 on 2 and 3204 DF,  p-value: 7.765e-07
```

```
plot_model(modelo_log, type="pred", terms ="eda")
```

```
## Model has log-transformed response. Back-transforming predictions to original response scale. Standa
```



Efecto cuadrático (ojo con la sintaxis)

```
modelo_quadr<-lm(log_ing_x_hrs ~ anios_esc + I(anios_esc^2) + sex,
                 data=ags_t321)
summary(modelo_quadr)
```

```
##
## Call:
## lm(formula = log_ing_x_hrs ~ anios_esc + I(anios_esc^2) + sex,
##     data = ags_t321)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.57404 -0.30066 -0.01732  0.28228  3.03536
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.5268462  0.0565398  62.378  < 2e-16 ***
## anios_esc      -0.0614917  0.0099999  -6.149 8.75e-10 ***
## I(anios_esc^2)  0.0059201  0.0004656  12.714  < 2e-16 ***
## sex            -0.0418652  0.0192457  -2.175   0.0297 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5304 on 3203 degrees of freedom
```

```
## Multiple R-squared:  0.2023, Adjusted R-squared:  0.2015
## F-statistic: 270.7 on 3 and 3203 DF,  p-value: < 2.2e-16
```

Quizás con un gráfico de lo predicho tenemos más claro lo que hace ese término

```
plot_model(modelo_quadr, type="pred", terms = c("anios_esc"))
```