Laboratorio 6 – Variables dummy y factores en R

Table of contents

El propósito de este laboratorio en clase es practicar el uso de **variables dummy** (indicadoras) en R.

Primeros pasos

Abre un nuevo script de R y carga los paquetes

```
# Agrega los nombres de los integrantes del grupo AQUÍ (si corresponde)

# Instalar paquetes solo si faltan (estilo ECO/EPG)
pkgs <- c("tidyverse", "broom", "wooldridge", "modelsummary", "kableExtra", "magrittr", "for
to_install <- pkgs[!pkgs %in% rownames(installed.packages())]
if (length(to_install) > 0) {
  install.packages(to_install, repos = "http://cran.us.r-project.org")
}

library(tidyverse)
library(broom)
library(wooldridge)
```

```r
library(modelsummary)
library(kableExtra)
library(magrittr)   # para %<>%
library(forcats)    # para fct_recode()

# Receta ECO/EPG: forzar tablas estables (LaTeX clásico)
options(modelsummary_factory_default = "kableExtra")
```

El paquete `magrittr` agrega funciones/operadores útiles para escribir código más expresivo (por ejemplo, `%<>%`, que "reescribe" un objeto después de pasarlo por un pipeline).

Cargar los datos

Usaremos un conjunto de datos sobre **relaciones extramaritales**, llamado `affairs`:

```r
df <- as_tibble(affairs)
```

Revisa estadísticas descriptivas:

```r
datasummary_skim(df, histogram = FALSE)
```

Notarás que hay varias variables que solo toman valores 0/1: `male`, `kids`, `affair`, `hapavg`, `vryrel`, etc. También hay variables con algunos valores discretos: `relig`, `occup` y `ratemarr`.

Crear variables tipo factor

Convirtamos la variable numérica 0/1 `male` en un factor con niveles `"yes"` y `"no"`:

```r
df %<>% mutate(
  male = factor(male),
  male = fct_recode(male, yes = "1", no = "0")
)
```

El uso `df %<>% mutate(...)` utiliza el operador `%<>%`, que es un atajo para:

`df <- df %>% mutate(...)`

En otras palabras, `%<>%` aplica el pipeline y luego sobrescribe `df` con el resultado, ahorrando algo de sintaxis.

	Unique	Missing Pct.	Mean	SD	Min	Median	Max
id	601	0	\num{1059.7}	\num{914.9}	\num{4.0}	\num{1009.0}	\num{902
male	2	0	\num{0.5}	\num{0.5}	\num{0.0}	\num{0.0}	\num{1.0}
age	9	0	\num{32.5}	\num{9.3}	\num{17.5}	\num{32.0}	\num{57.0
yrsmarr	8	0	\num{8.2}	\num{5.6}	\num{0.1}	\num{7.0}	\num{15.0
kids	2	0	\num{0.7}	\num{0.5}	\num{0.0}	\num{1.0}	\num{1.0}
relig	5	0	\num{3.1}	\num{1.2}	\num{1.0}	\num{3.0}	\num{5.0}
educ	7	0	\num{16.2}	\num{2.4}	\num{9.0}	\num{16.0}	\num{20.0
occup	7	0	\num{4.2}	\num{1.8}	\num{1.0}	\num{5.0}	\num{7.0}
ratemarr	5	0	\num{3.9}	\num{1.1}	\num{1.0}	\num{4.0}	\num{5.0}
naffairs	6	0	\num{1.5}	\num{3.3}	\num{0.0}	\num{0.0}	\num{12.0
affair	2	0	\num{0.2}	\num{0.4}	\num{0.0}	\num{0.0}	\num{1.0}
vryhap	2	0	\num{0.4}	\num{0.5}	\num{0.0}	\num{0.0}	\num{1.0}
hapavg	2	0	\num{0.3}	\num{0.5}	\num{0.0}	\num{0.0}	\num{1.0}
avgmarr	2	0	\num{0.2}	\num{0.4}	\num{0.0}	\num{0.0}	\num{1.0}
unhap	2	0	\num{0.1}	\num{0.3}	\num{0.0}	\num{0.0}	\num{1.0}
vryrel	2	0	\num{0.1}	\num{0.3}	\num{0.0}	\num{0.0}	\num{1.0}
smerel	2	0	\num{0.3}	\num{0.5}	\num{0.0}	\num{0.0}	\num{1.0}
slghtrel	2	0	\num{0.2}	\num{0.4}	\num{0.0}	\num{0.0}	\num{1.0}
notrel	2	0	\num{0.3}	\num{0.4}	\num{0.0}	\num{0.0}	\num{1.0}

La primera parte de `mutate()` convierte 0/1 en categorías `"0"` y `"1"`. La segunda parte les asigna etiquetas más descriptivas (`"yes"` y `"no"`).

Repitamos esto para otras variables: `ratemarr`, `relig`, `kids` y `affair`:

```
df %<>%
  mutate(
    ratemarr = factor(ratemarr),
    ratemarr = fct_recode(
      ratemarr,
      very_happy    = "5",
      happy         = "4",
      average       = "3",
      unhappy       = "2",
      very_unhappy  = "1"
    )
  ) %>%
  mutate(
    relig = factor(relig),
    relig = fct_recode(
      relig,
      very_relig        = "5",
      relig             = "4",
      average           = "3",
      not_relig         = "2",
      not_at_all_relig  = "1"
    )
  ) %>%
  mutate(
    kids = factor(kids),
    kids = fct_recode(kids, yes = "1", no = "0")
  ) %>%
  mutate(
    affair = factor(affair),
    affair = fct_recode(affair, yes = "1", no = "0")
  )
```

Vuelve a ejecutar:

```
datasummary_skim(df, histogram = FALSE)
```

Verás que las variables tipo factor pueden "desaparecer" del output por defecto de `datasummary_skim()`. Para que aparezcan como categóricas, usa el tipo `categorical`.

	Unique	Missing Pct.	Mean	SD	Min	Median
id	601	0	\num{1059.7}	\num{914.9}	\num{4.0}	\num{1009.0}
age	9	0	\num{32.5}	\num{9.3}	\num{17.5}	\num{32.0}
yrsmarr	8	0	\num{8.2}	\num{5.6}	\num{0.1}	\num{7.0}
educ	7	0	\num{16.2}	\num{2.4}	\num{9.0}	\num{16.0}
occup	7	0	\num{4.2}	\num{1.8}	\num{1.0}	\num{5.0}
naffairs	6	0	\num{1.5}	\num{3.3}	\num{0.0}	\num{0.0}
vryhap	2	0	\num{0.4}	\num{0.5}	\num{0.0}	\num{0.0}
hapavg	2	0	\num{0.3}	\num{0.5}	\num{0.0}	\num{0.0}
avgmarr	2	0	\num{0.2}	\num{0.4}	\num{0.0}	\num{0.0}
unhap	2	0	\num{0.1}	\num{0.3}	\num{0.0}	\num{0.0}
vryrel	2	0	\num{0.1}	\num{0.3}	\num{0.0}	\num{0.0}
smerel	2	0	\num{0.3}	\num{0.5}	\num{0.0}	\num{0.0}
slghtrel	2	0	\num{0.2}	\num{0.4}	\num{0.0}	\num{0.0}
notrel	2	0	\num{0.3}	\num{0.4}	\num{0.0}	\num{0.0}
		N	\%			
male	no	315	\num{52.4}			
	yes	286	\num{47.6}			
kids	no	171	\num{28.5}			
	yes	430	\num{71.5}			
relig	not_at_all_relig	48	\num{8.0}			
	not_relig	164	\num{27.3}			
	average	129	\num{21.5}			
	relig	190	\num{31.6}			
	very_relig	70	\num{11.6}			
ratemarr	very_unhappy	16	\num{2.7}			
	unhappy	66	\num{11.0}			
	average	93	\num{15.5}			
	happy	194	\num{32.3}			
	very_happy	232	\num{38.6}			
affair	no	451	\num{75.0}			
	yes	150	\num{25.0}			

		N	%
male	no	315	52.4
	yes	286	47.6
kids	no	171	28.5
	yes	430	71.5
relig	not_at_all_relig	48	8.0
	not_relig	164	27.3
	average	129	21.5
	relig	190	31.6
	very_relig	70	11.6
ratemarr	very_unhappy	16	2.7
	unhappy	66	11.0
	average	93	15.5
	happy	194	32.3
	very_happy	232	38.6
affair	no	451	75.0
	yes	150	25.0

Estadísticas descriptivas de factores

Puedes ver frecuencias con `datasummary_skim()` o con `table()`:

```
datasummary_skim(df, type = "categorical", histogram = FALSE)
```

```
table(df$relig)
```

```
not_at_all_relig        not_relig          average            relig
              48              164              129              190
      very_relig
              70
```

```
table(df$ratemarr, df$kids)
```

```
              no yes
  very_unhappy  3  13
  unhappy       8  58
  average      24  69
  happy        40 154
  very_happy   96 136
```

También puedes usar `prop.table()` para obtener proporciones por fila (`margin=1`) o por columna (`margin=2`):

```
table(df$ratemarr) %>% prop.table()
```

```
very_unhappy      unhappy      average        happy   very_happy
   0.0266223    0.1098170    0.1547421    0.3227953    0.3860233
```

```
table(df$ratemarr, df$kids) %>% prop.table(margin = 1)
```

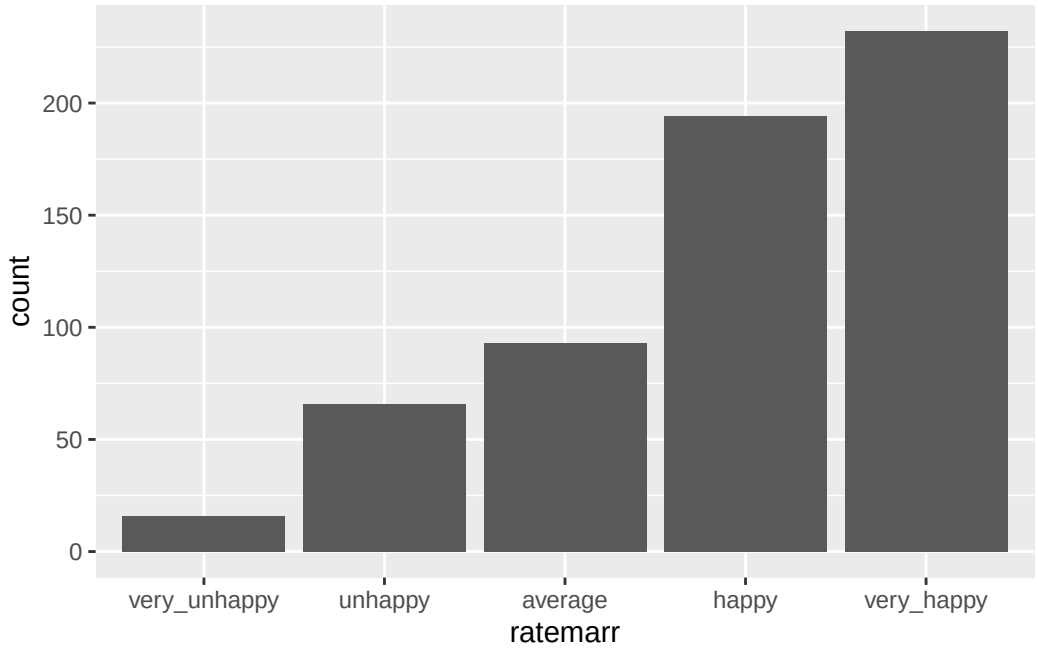
```
                    no       yes
  very_unhappy 0.1875000 0.8125000
  unhappy      0.1212121 0.8787879
  average      0.2580645 0.7419355
  happy        0.2061856 0.7938144
  very_happy   0.4137931 0.5862069
```

```
table(df$ratemarr, df$kids) %>% prop.table(margin = 2)
```

```
                     no        yes
  very_unhappy 0.01754386 0.03023256
  unhappy      0.04678363 0.13488372
  average      0.14035088 0.16046512
  happy        0.23391813 0.35813953
  very_happy   0.56140351 0.31627907
```

También puedes graficar un histograma (barras) para una variable categórica:

```r
ggplot(df, aes(x = ratemarr)) + geom_bar()
```



Esto ayuda a visualizar qué fracción de los datos cae en cada categoría.

Regresión múltiple con variables factor

Corramos una regresión con `naffairs` como variable dependiente y `male`, `yrsmarr`, `kids` y `ratemarr` como covariables:

```r
est1 <- lm(naffairs ~ male + yrsmarr + kids + ratemarr, data = df)
```

Mostramos la tabla del modelo (estable para PDF):

```r
modelsummary(est1, output = "kableExtra") |>
  kable_styling(full_width = FALSE, position = "center", latex_options = "hold_position")
```

Interpreta el coeficiente asociado a `ratemarrvery_happy`.

	(1)
(Intercept)	2.934
	(0.845)
maleyes	0.083
	(0.257)
yrsmarr	0.086
	(0.028)
kidsyes	−0.212
	(0.349)
ratemarrunhappy	0.277
	(0.877)
ratemarraverage	−2.136
	(0.854)
ratemarrhappy	−2.275
	(0.821)
ratemarrvery_happy	−2.683
	(0.822)
Num.Obs.	601
R2	0.112
R2 Adj.	0.102
AIC	3085.7
BIC	3125.3
Log.Lik.	−1533.873
F	10.696
RMSE	3.11

Modelo de Probabilidad Lineal (LPM)

Corramos el mismo modelo, pero ahora usando `affair` como variable dependiente. ¿Qué pasa si ejecutas el siguiente código?

```
est2_try <- lm(affair ~ male + yrsmarr + kids + ratemarr, data = df)
tidy(est2_try)
```

```
# A tibble: 8 x 5
  term               estimate std.error statistic p.value
  <chr>                 <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)         1.39           NA        NA      NA
2 maleyes             0.0393         NA        NA      NA
3 yrsmarr             0.00415        NA        NA      NA
```

```
4 kidsyes              0.0513         NA        NA      NA
5 ratemarrunhappy      0.00138        NA        NA      NA
6 ratemarraverage     -0.197          NA        NA      NA
7 ratemarrhappy       -0.245          NA        NA      NA
8 ratemarrvery_happy  -0.321          NA        NA      NA
```

R no "quiere" estimar este LPM porque `affair` es un factor (categórico) y el `lm()` espera una variable numérica continua en el lado izquierdo. Para correr el LPM, convierte `affair` a numérica usando `as.numeric(affair)` como variable dependiente. (Ojo: `as.numeric()` sobre factor devuelve 1/2 según el nivel; aquí lo usamos para replicar el ejercicio. Si quieres 0/1 exacto, lo hacemos con una transformación explícita.)

```
est2 <- lm(as.numeric(affair) ~ male + yrsmarr + kids + ratemarr, data = df)
```

Tabla del LPM:

```
modelsummary(est2, output = "kableExtra") |>
  kable_styling(full_width = FALSE, position = "center", latex_options = "hold_position")
```

Interpreta los coeficientes de `ratemarraverage` y `kidsyes`.

Términos de interacción

Finalmente, corramos un modelo más flexible donde permitimos que el efecto de `kids` sea diferente para hombres y mujeres. En `lm()` esto se escribe así:

```
est3 <- lm(as.numeric(affair) ~ male * kids + yrsmarr + ratemarr, data = df)
tidy(est3)
```

```
# A tibble: 9 x 5
  term                estimate std.error statistic  p.value
  <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)         1.39       0.116    12.0     7.09e-30
2 maleyes             0.0549     0.0652    0.842   4.00e- 1
3 kidsyes             0.0616     0.0594    1.04    3.00e- 1
4 yrsmarr             0.00407    0.00382   1.06    2.88e- 1
5 ratemarrunhappy     0.000797   0.117     0.00680 9.95e- 1
6 ratemarraverage    -0.197      0.114    -1.73    8.45e- 2
7 ratemarrhappy      -0.244      0.110    -2.22    2.66e- 2
8 ratemarrvery_happy -0.321      0.110    -2.91    3.71e- 3
9 maleyes:kidsyes    -0.0216     0.0770   -0.281   7.79e- 1
```

	(1)
(Intercept)	1.394
	(0.113)
maleyes	0.039
	(0.034)
yrsmarr	0.004
	(0.004)
kidsyes	0.051
	(0.047)
ratemarrunhappy	0.001
	(0.117)
ratemarraverage	−0.197
	(0.114)
ratemarrhappy	−0.245
	(0.110)
ratemarrvery_happy	−0.321
	(0.110)
Num.Obs.	601
R2	0.080
R2 Adj.	0.069
AIC	667.0
BIC	706.5
Log.Lik.	−324.477
F	7.331
RMSE	0.42

Mostramos la tabla del modelo con interacción:

```
modelsummary(est3, output = "kableExtra") |>
  kable_styling(full_width = FALSE, position = "center", latex_options = "hold_position")
```

El coeficiente del término de interacción se etiqueta `maleyes:kidsyes`.
¿Tienen los padres una tasa diferencial de relaciones extramaritales comparado con las madres, según este modelo?

	(1)
(Intercept)	1.387
	(0.116)
maleyes	0.055
	(0.065)
kidsyes	0.062
	(0.059)
yrsmarr	0.004
	(0.004)
ratemarrunhappy	0.001
	(0.117)
ratemarraverage	−0.197
	(0.114)
ratemarrhappy	−0.244
	(0.110)
ratemarrvery_happy	−0.321
	(0.110)
maleyes × kidsyes	−0.022
	(0.077)
Num.Obs.	601
R2	0.080
R2 Adj.	0.067
AIC	668.9
BIC	712.9
Log.Lik.	−324.437
F	6.415
RMSE	0.42