Tema 06 - Modelización: métodos estadísticos

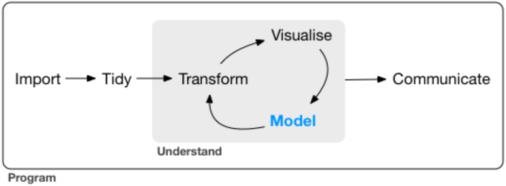
Pedro Albarrán

Dpto. de Fundamentos del Análisis Económico. Universidad de Alicante

Contenidos I

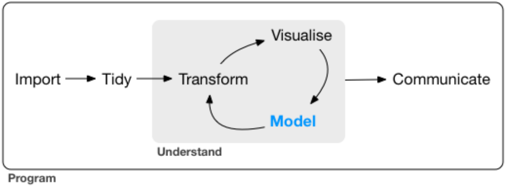
Métodos Estadísticos

- Los métodos estadísticos (**modelización**) permiten
  - Encontrar patrones
  - Interpretar datos



Métodos Estadísticos

- Los métodos estadísticos (**modelización**) permiten
  - Encontrar patrones
  - Interpretar datos



- Los datos (casos observados) son una **muestra** de una *población* mayor (casos potenciales)
- Las ventas salen de una población teórica $\chi^2_{(10)}$, con media poblacional 10:

```
data <- tibble(x = rchisq(n=1e6, df=10))
ggplot(data, aes(x)) + geom_vline(xintercept = 10, color = "red") +
  geom_density()
```

- ¿Cómo de fiable es un *estadístico* (ej., la media) calculado en una *muestra*?

Variabilidad muestral

- Simulemos la distribución de la media en muchas muestras de $n = 25$

```
set.seed(101)
nsim <- 100
N <- 25
ybar <- numeric(nsim)
```

```
for (j in 1:nsim) {
  muestra <- round(rchisq(n=N, df=10))
  ybar[j] <- mean(muestra)
}
summary(ybar)
```

Variabilidad muestral

- Simulemos la distribución de la media en muchas muestras de $n = 25$

```
set.seed(101)
nsim <- 100
N <- 25
ybar <- numeric(nsim)
```

```
for (j in 1:nsim) {
  muestra <- round(rchisq(n=N, df=10))
  ybar[j] <- mean(muestra)
}
summary(ybar)
```

- **Distribución muestral** es la distribución del estadístico en **todas** las muestras potenciales de tamaño n

```
as_tibble(ybar) %>% ggplot(aes(x=value)) + geom_density() # solo 10(
```

- ¿Es posible *cuantificar la incertidumbre* con UNA MUESTRA?

Procedimiento *Bootstrap*

- **Idea**: pensar en la ÚNICA muestra como si fuera la población

1. Tomar muchas *remuestras* (muestras de Bootstrap) con reemplazamiento
   - p.e., para (1,2,3) incluye los casos (1,1,2), (1,1,3), (2,2,1), etc.
2. En cada remuestra, se puede calcular cualquier estadístico

- Este procedimiento permite generar variación (de remuestras) a partir de una única muestra
- La **distribución muestral bootstrap** NO es la distribución muestral, pero aproxima sus aspectos principales sin supuestos (normalidad, TCL)
- Puede aplicarse a **cualquier estadístico** (media, varianzas, regresión, etc.)

Procedimiento *Bootstrap* (cont.)

```
library(rsample)
set.seed(101)
UNAmuestra <- tibble(x=rchisq(n=25, df=10))

nboot <- 10
remuestras <- bootstraps(UNAmuestra, times = nboot)

distrib <- list()
for (i in 1:nboot) {
  remuestrai <- remuestras$splits[[i]] %>% as_tibble()
  mediai  <- mean(remuestrai$x)
  sdi     <- sd(remuestrai$x)
  distrib[[i]] <- list(medias = mediai, sds =sdi )
}

distribF <- distrib %>% bind_rows()
```

```
distribF %>% ggplot(aes(x = medias)) +
```

Modelización Condicional. Causalidad

- La población NO suele ser *homogenea*: más visitas ciertos días, horas, etc.
- Un modelo de regresión permite incluir todas las **variables explicativas** de la variación de la variable dependiente
- **"Correlación no implica causalidad"**, salvo en ensayos científicos aleatorios cuidadosamente controlados
  - en otros campos como marketing digital o analítica de web se denominan pruebas A/B (ej. dos versiones de una misma web)
- En general (datos observacionales), nos preocupa que otros factores que puedan ser los verdaderos determinantes de la relación observada

"Confounding factor": Descuentos y ventas

```
datos <- read_csv("data/discount.csv")
```

- El porcentaje medio de descuentos afecta negativamente a las ventas

```
datos %>% ggplot(aes(x=discount, y=sales)) + geom_point() + geom_smo
lm(data = datos, sales ~ discount) %>% summary()
```

- Pero si tenemos en cuenta la renta...

```
datos %>% mutate(renta_baja = income < 7500) %>%
  ggplot(aes(x=discount, y=sales)) + geom_point() + geom_smooth(meth
  facet_wrap(~renta_baja)
lm(data = datos, sales ~ discount + income) %>% summary()
```

Regresión Lineal

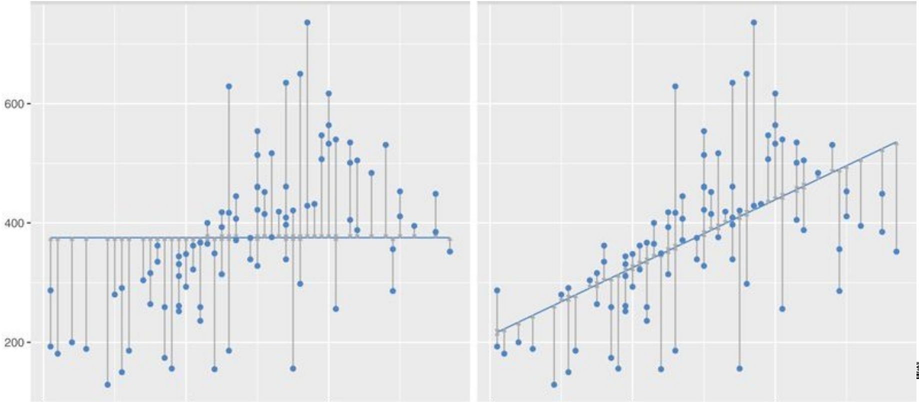
- La regresión lineal predice una respuesta cuantitativa Y como a partir de k regresores $X = X_1, X_2, \dots, X_k$
- Supuesto: relación lineal entre X e Y

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \varepsilon$$

- Los coeficientes o parámetros del modelo representan
  - β_0 (constante): valor esperado de Y cuando $X_1 = X_2 = \cdots = X_k = 0$
  - β_j (pendiente de la línea): cambio medio en Y por un incremento de una unidad en X_j (para $j = 1, ..., k$), *ceteris paribus*
- Objetivo: estimar los coeficientes desconocidos a partir de una muestra

Regresión Lineal: Estimación

- El error de estimación o **residuo** es $\widehat{e}_i = y_i - \widehat{y}_i$, donde la predicción a partir del modelo estimado es $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 X_1 + \cdots + \widehat{\beta}_k X_p$
- Los coeficientes estimados son los que minimizan la Suma Cuadrática de Residuos: la suma total de distancias entre los datos observados y predichos



Regresión Lineal: Precisión de las estimaciones

- El **error estándar** $se(\widehat{\beta}_j)$ mmide la precisión del coeficiente estimado
- Se usan para construir intervalos de confianza y estadísticos para contrastar hipótesis sobre los parámetros, p.e., significatividad
  - individual: $H_0 : \beta_1 = 0$ con un estadístico $t = \frac{\widehat{\beta}_1 - 0}{se(\widehat{\beta}_1)}$
  - conjunta: $H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0$ con un estadístico F
- Medidas de la precisión del modelo: MSE o $R^2 = 1 - \frac{SCR}{SCT} = \frac{SCE}{SCT}$
- La predicción $\hat{y}$ también está sujeta a incertidumbre por la estimación: se puede calcular su error estándar e intervalos de confianzas

```
res <- lm(data = datos, sales ~ discount + income)
cbind(datos$sales, res$fitted.values) %>% head()
```

Regresión Lineal: superando la linealidad

- Se pueden incluir transformaciones no lineales de las variables del modelo
- La interpretación del cambio de un regresor sobre Y es diferente

```
library(ISLR)
data(Carseats)
lm(data=Carseats %>% filter(Sales != 0), log(Sales) ~ poly(Advertisi
```

- Otra forma de permitir no linealidades es discretizando variables continuas: permite efectos "escalón" diferentes para distintos valores

```
lm(data=Carseats , Sales ~ cut_width(Advertising, 10)) %>% summary()
```

- Se incluyen indicadores binarios para cada clase excepto uno
  - la constante recoge el valor medio de Y para ese grupo de referencia
  - cada coeficiente recoge el efecto adicional de su grupo sobre Y

Regresión Lineal: superando la linealidad (cont.)

- También podemos incluir interacciones entre variables: el efecto de un regresor dependerá de otro regresor

```
lm(data=Carseats , Sales ~ Advertising*Income)
```

- Principio jerárquico: al incluir una interacción *siempre* deben incluirse los factores principales (NO sólo `Advertising:Income`)
- Cuando interactuamos un regresor continuo y uno binario, permitimos que la pendiente del primero sea diferente para cada grupo

```
lm(data=Carseats , Sales ~ (Income + Advertising)*Urban)
```

- La interacción de dos variables binarias tiene una interpretación similar

```
lm(data=Carseats , Sales ~ ShelveLoc*Urban)
```

“Problemas” del Modelo de Regresión Lineal

- *No linealidad*: incluir transformaciones no lineales
- *Correlación de los errores*: afecta a los errores estándar, no la estimación
  - usar errores estándar robustos o modelizar la dinámica
- *Heterocedasticidad*: ídem, usar errores estándar robustos
  - los gráficos de los residuos frente a un regresor o valores predichos: ¿heterocedasticidad o no linealidad?
- *Outliers* en la variable de respuesta o en los regresores
- *Colinearidad*: indica que no es posible separar el efecto de cada regresor: eliminar alguno o recombinarlos
- *No normalidad*: TCL, Bootstrap,...
- El único supuesto realmente importante es $E[\varepsilon|X] = 0$

Regresión Logística

- La regresión lineal puede usarse respuestas binarias (no más de dos categorías), aunque genera predicciones fuera del rango $[0, 1]$
- Solución: aplicar al índice lineal una transformación $F(z) \in [0, 1]$
- La función logística: $\Lambda(z) = \frac{e^z}{1+e^z}$
- De manera que $\Pr(Y = 1|X) = p(x) = \Lambda(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)$

Regresión Logística (cont.)

- Los coeficientes NO se interpretan como cambios en la probabilidad ante cambios unitarios en un regresor (efecto marginal sobre la probabilidad)
- PERO su signo (y significatividad) son los mismos que los del efecto marginal
- Como NO tiene sentido minimizar la SCR, el objetivo es maximizar la probabilidad (verosimilitud) de observar los unos y ceros en los datos
- La regresión logística pertenece a la familia de modelos lineales generalizados (GLM, en inglés)
- Se pueden incluir como variables explicativas tanto variables cuantitativas como cualitativas, e incurrir en sesgo por omisión de variables

```
glm(data = Default, default ~ student, family = "binomial" ) %>% sum
glm(data = Default, default ~ student + balance, family = "binomial'
```

Regresión Logística: Predicciones

- El objeto de R de `glm()` contiene valores predichos, que son probabilidades de $Y = 1$

```
logit <- glm(data = Default, default ~ balance*student, family = "bi
cbind(Default$default, logit$fitted)
```

- También se puede predecir usando una muestra distinta de la usada para estimar o con valores concretos de los regresores

```
logit <- glm(data = Default, default ~ balance, family = "binomial"
predict(logit, newdata = tibble(balance=c(0,100)), type="response")
```

Regresión logística con más de dos clases

- La regresión logística se puede generalizar a situaciones con múltiples clases (modelos multinomiales) con un índice lineal para cada clase

$$\Pr(Y = c|X) = \frac{e^{\beta_{0c}+\beta_{1c}X_1+\cdots+\beta_{kc}X_k}}{\sum_{l=1}^{C} e^{\beta_{0l}+\beta_{1l}X_1+\cdots+\beta_{kl}X_k}}$$

- La librería `glmnet()` permite la estimación de estos modelos

```
library(glmnet)
x <- model.matrix(Species ~ Sepal.Length + Sepal.Width, data = iris)
mod.glmnet <- glmnet(x = x, y = iris$Species, family = "multinomial"
                     lambda = 0, type.multinomial = "grouped")
coef(mod.glmnet)
predict(mod.glmnet, newx=x, type = "response")  # probabilidad de ca
predict(mod.glmnet, newx=x, type = "class")     # clase
```