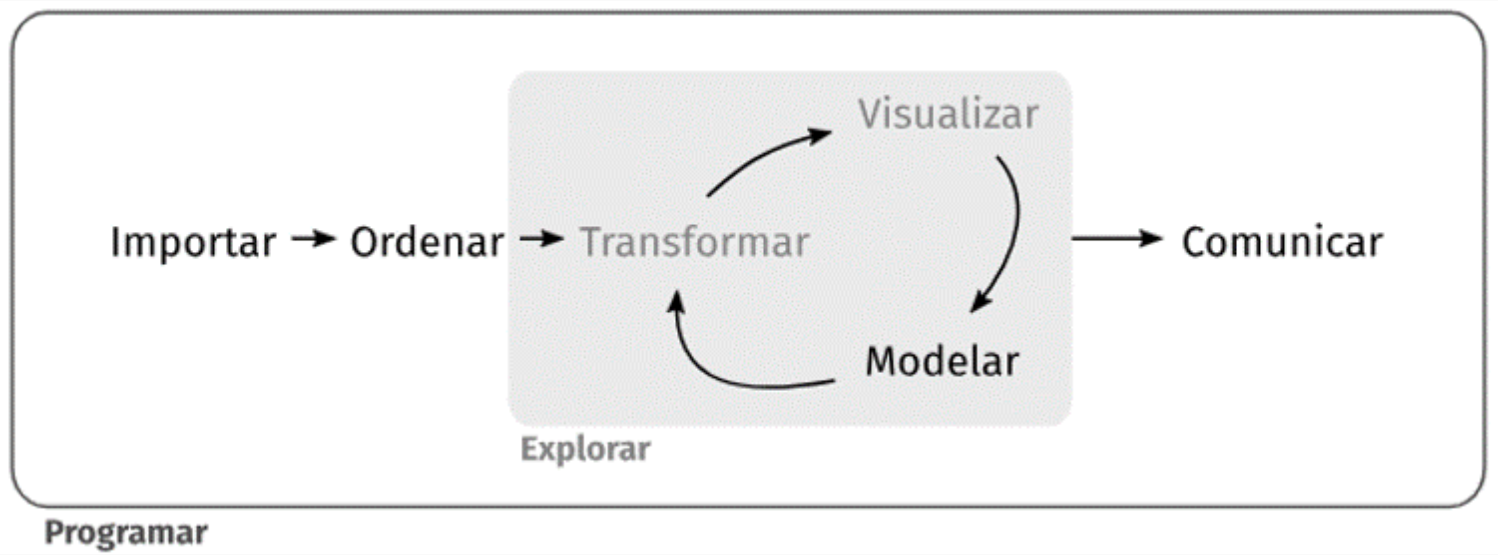
Ciencia de Datos para Políticas Públicas

Clase 05 - Regresión y Clasificación

Pablo Aguirre Hormann
02/09/2020

¿Qué veremos hoy?

- Regresión
- Clasificación



¿Por qué?

Objetivo: representar la relación entre una variable dependiente Y y una o varias variables explicativas/independientes $X_1, X_2, \ldots, X_k$.

$$Y = f(X) + \epsilon$$

- Si Y es una variable *continua*: **regresión**
- Si Y es una variable *categórica*: **clasificación**

Inferencia vs Predicción

Inferencia:

- Aprender y concluir algo sobre como se relacionan variables
- Evitar sesgo
- *Dentro de muestra*
- $\hat{f}$ / $\hat{\beta}$

Predicción:

- Que la predicción esté lo más cerca posible del valor real
- Evitar sobreajuste al entrenar modelos
- *Fuera de muestra*
- $\hat{Y}$

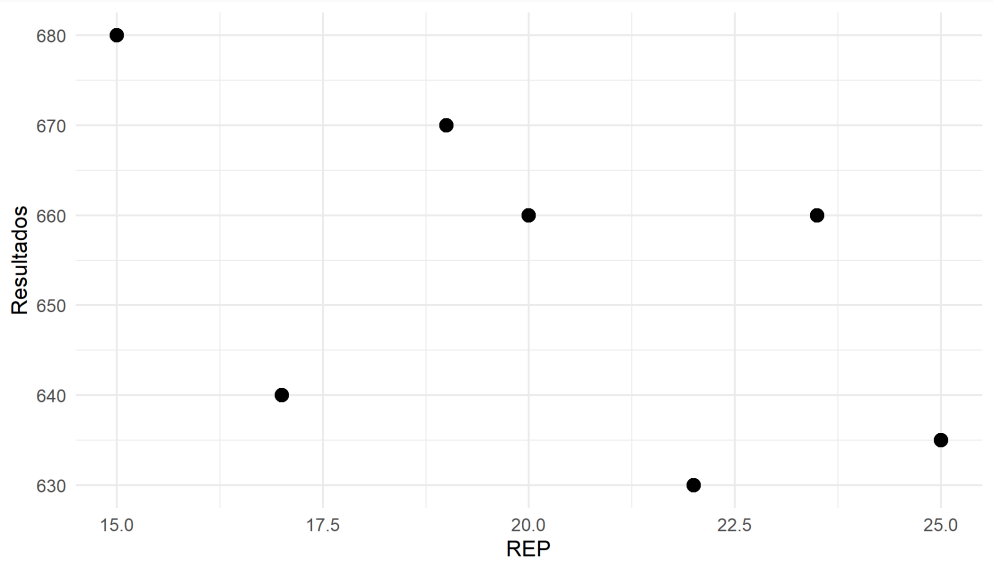
Algunos algorítmos pueden servir para ambos objetivos pero con diferencias en la implementación (*ej. Regresión lineal para inferencia o para predicción*).

Por ahora no hablaremos de predicción

Regresión Lineal

Estudiantes/Profesor vs Resultados

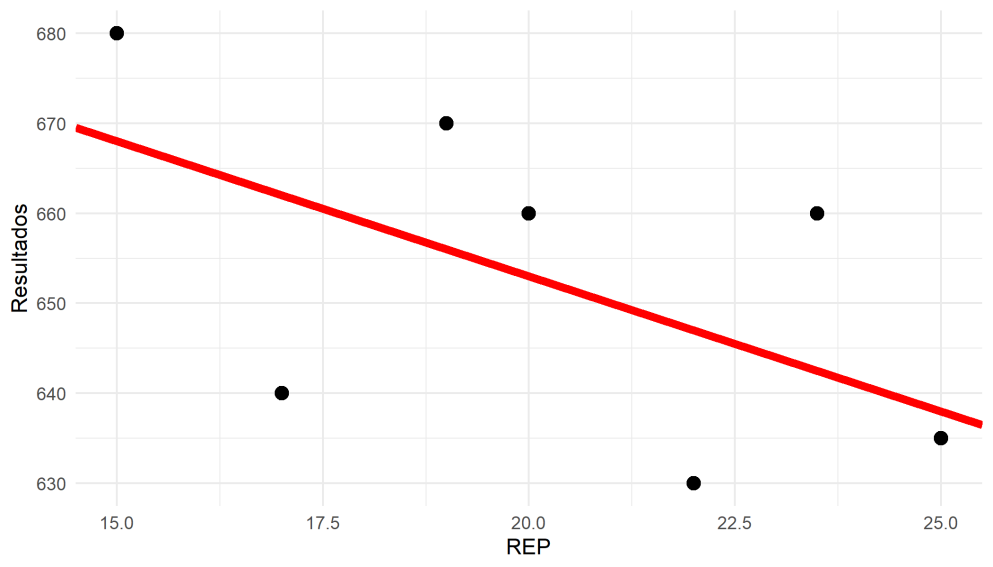
```r
# crear vectores de datos
REP ← c(15, 17, 19, 20, 22, 23.5, 25)
Resultados ← c(680, 640, 670, 660, 630, 660, 635)

# juntar ambos vectores en un data frame
datos_colegio ← data.frame(REP, Resultados)

ggplot(datos_colegio, aes(REP, Resultados)) +
  geom_point(size = 3) +
  theme_minimal()
```



Una linea que describe esta relación (?)

$$Resultados_i = 713 - 3 * REP_i$$

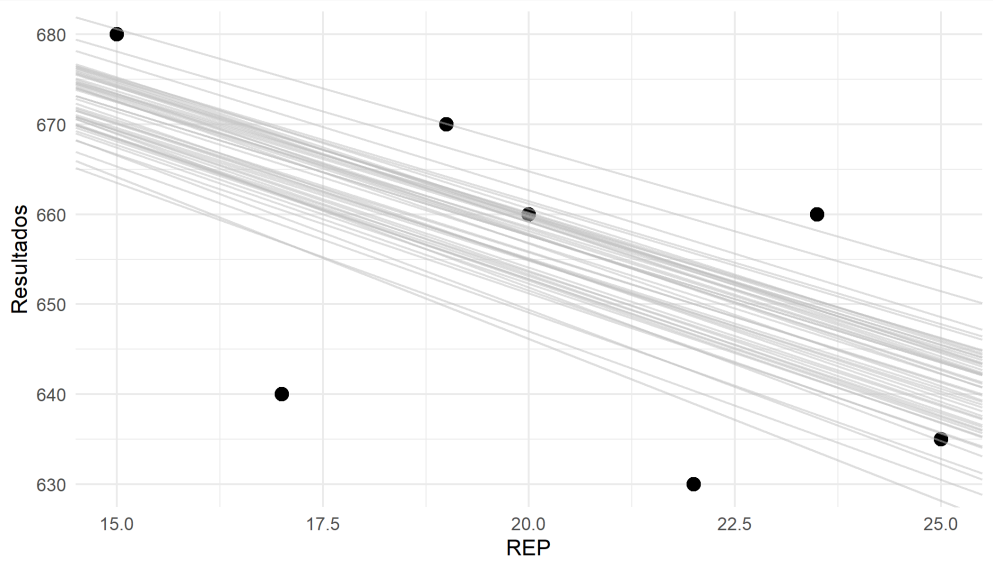
```
ggplot(datos_colegio, aes(REP, Resultados)) +
  geom_point(size = 3) +
  geom_abline(aes(intercept = 713, slope = -3), size = 2, col = "red") +
  theme_minimal()
```



Ahora muchas lineas

$$Resultados_i = ¿? - ¿? * REP_i$$

```
curvas ← data.frame(int = sample(713:720, 50, replace = TRUE),
                    pend = -3 + rnorm(50, mean = 0, sd = 0.2))

ggplot(datos_colegio, aes(REP, Resultados)) +
  geom_point(size = 3) +
  geom_abline(aes(intercept = int, slope = pend), data = curvas, alpha = 0.5, col = "g
  theme_minimal()
```



Mínimos Cuadrados Ordinarios

- Dada una verdad:

$$Y_i = \beta_0 + \beta_1 * X_i + u_i$$

- Realizamos una estimación:

$$y_i = b_0 + b_1 * X_i$$

$$Datos \rightarrow Cálculos \rightarrow Estimación \xrightarrow{si\ todo\ sale\ bien} Verdad$$

$$X, Y \rightarrow (X'X)^{-1}X'Y \rightarrow \hat{\beta} \xrightarrow{si\ todo\ sale\ bien} \beta$$

$\hat{\beta} = b$

Mínimos Cuadrados Ordinarios (cont.)

Regresión lineal simple

- Modelo a estimar dada la existencia de ciertos datos

$$\hat{y}_i = b_0 + b_1 * X_i$$

- ¿Cómo se estiman los parámetros?
  - **objetivo**: *minimizar la suma del cuadrado de los residuales*

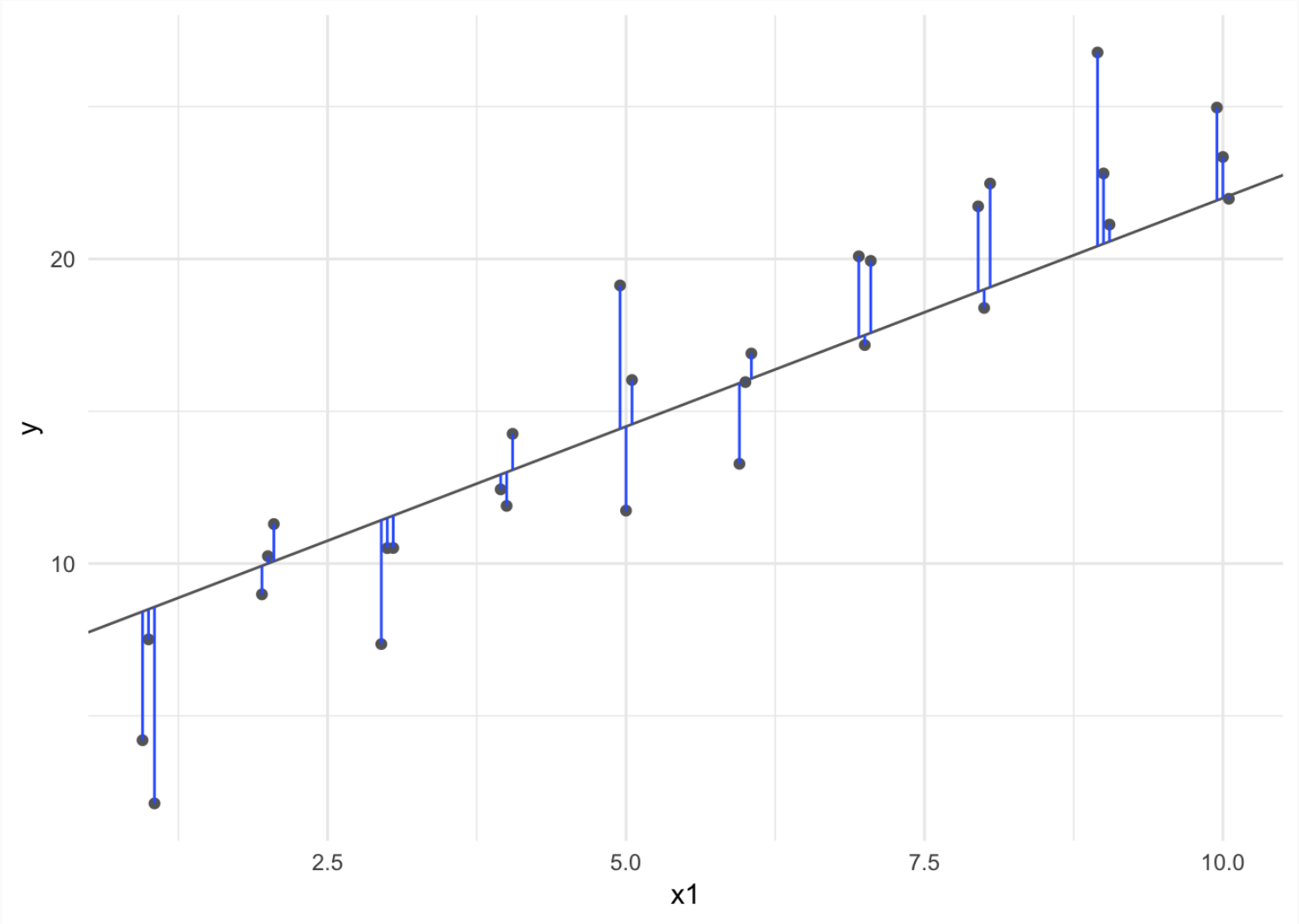
$$\min_{b_o,b_1} \sum_i^n \hat{u_i}^2 = \min_{b_o,b_1} \sum_i^n (Y_i - \hat{y}_i)^2 = \min_{b_o,b_1} \sum_i^n (Y_i - b_0 + b_1 * X_i)^2$$

- Parámetros estimados

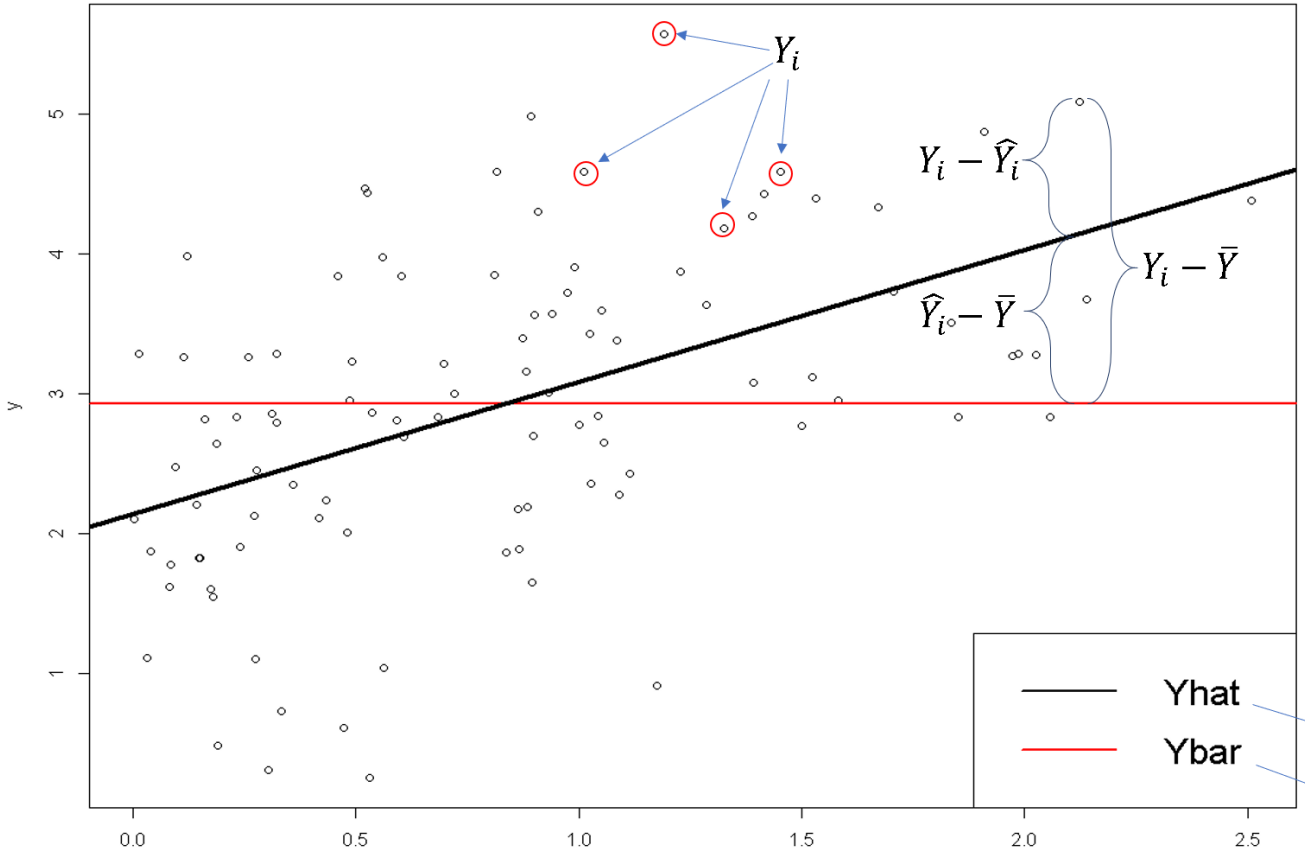
$$b_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

$$b_0 = \bar{Y} - b_1 \bar{X}$$

Representación visual



Representación visual (cont)



$$\sum (Y_i - \bar{Y})^2 = \sum (\hat{Y}_i - \bar{Y})^2 + \sum (Y_i - \hat{Y}_i)^2$$

Sum de cuadrados totales

$$\sum (Y_i - \bar{Y})^2$$

Suma de cuadrados de la regresión

$$\sum (\hat{Y}_i - \bar{Y})^2$$

Suma de cuadrados de los residuales

$$\sum (Y_i - \hat{Y}_i)^2$$

Valores Ajustados(E[Y|X])

Promedio de Y (E[Y])

R^2 - Coeficiente de determinación

- Suma de cuadrados de la regresión (SCR):

$$\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$$

- Suma de cuadrados de los residuales (SCE):

$$\sum_{i=1}^{n}{\hat{u}_i}^2 = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

- Suma de cuadrados totales (SCT):

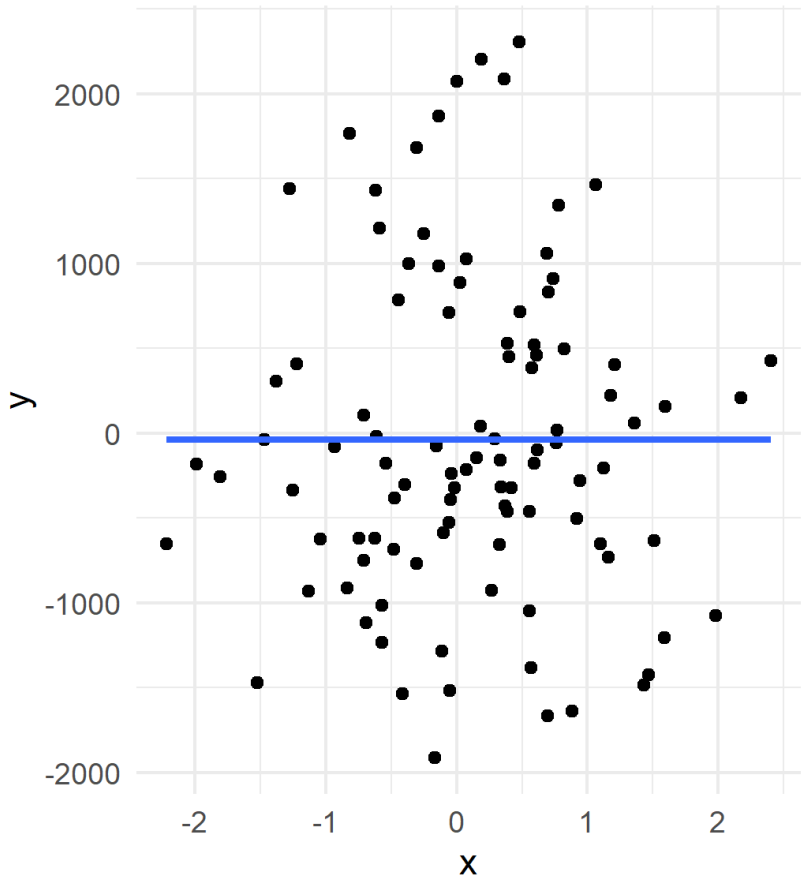
$$SCR + SCE = \sum(Y_i - \bar{Y})^2$$

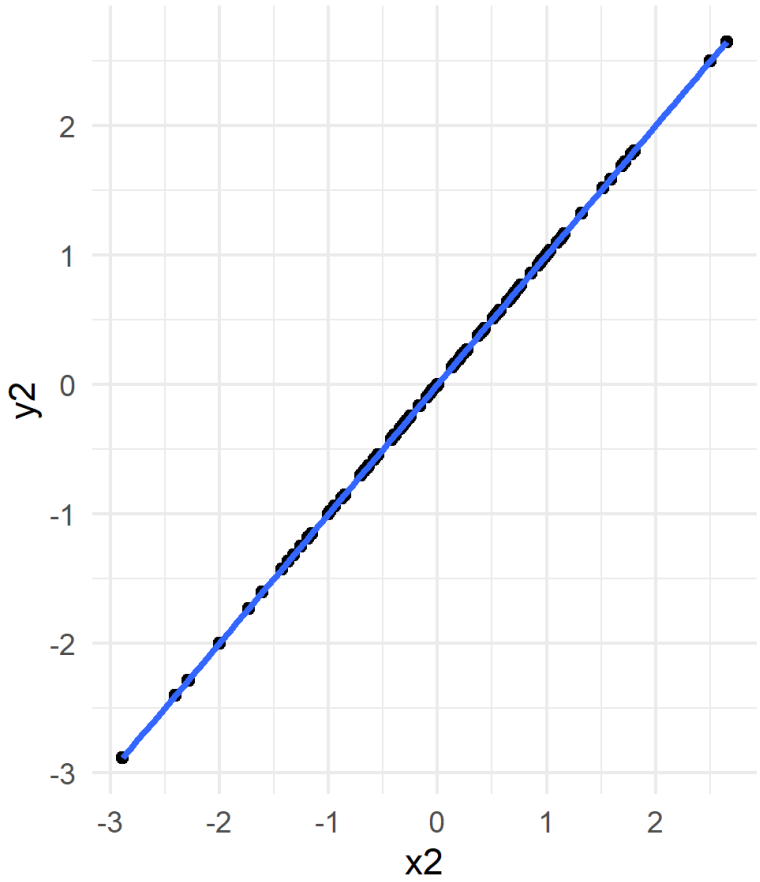
Teniendo estos valores, el coeficiente de determinación corresponde a:

$$R^2 = \frac{SCR}{SCT} = 1 - \frac{SCE}{SCT}$$

Noten que $R^2 \in [0, 1]$, con 0 correspondiente a un nulo ajuste y 1 a un ajuste perfecto (todos los puntos sobre la curva estimada)

¿Cómo se ve esto?





Ahora con "datos reales"

(Disponibles en el paquete **AER**)

Datos California (USA)

```
library(tidyverse)
library(AER)
data("CASchools")
str(CASchools)
```

```
## 'data.frame':    420 obs. of  14 variables:
##  $ district   : chr  "75119" "61499" "61549" "61457" ...
##  $ school     : chr  "Sunol Glen Unified" "Manzanita Elementary" "Thermalito Union Eleme
##  $ county     : Factor w/ 45 levels "Alameda","Butte",..: 1 2 2 2 2 6 29 11 6 25 ...
##  $ grades     : Factor w/ 2 levels "KK-06","KK-08": 2 2 2 2 2 2 2 2 2 1 ...
##  $ students   : num  195 240 1550 243 1335 ...
##  $ teachers   : num  10.9 11.1 82.9 14 71.5 ...
##  $ calworks   : num  0.51 15.42 55.03 36.48 33.11 ...
##  $ lunch      : num  2.04 47.92 76.32 77.05 78.43 ...
##  $ computer   : num  67 101 169 85 171 25 28 66 35 0 ...
##  $ expenditure: num  6385 5099 5502 7102 5236 ...
##  $ income     : num  22.69 9.82 8.98 8.98 9.08 ...
##  $ english    : num  0 4.58 30 0 13.86 ...
##  $ read       : num  692 660 636 652 642 ...
##  $ math       : num  690 662 651 644 640 ...
```

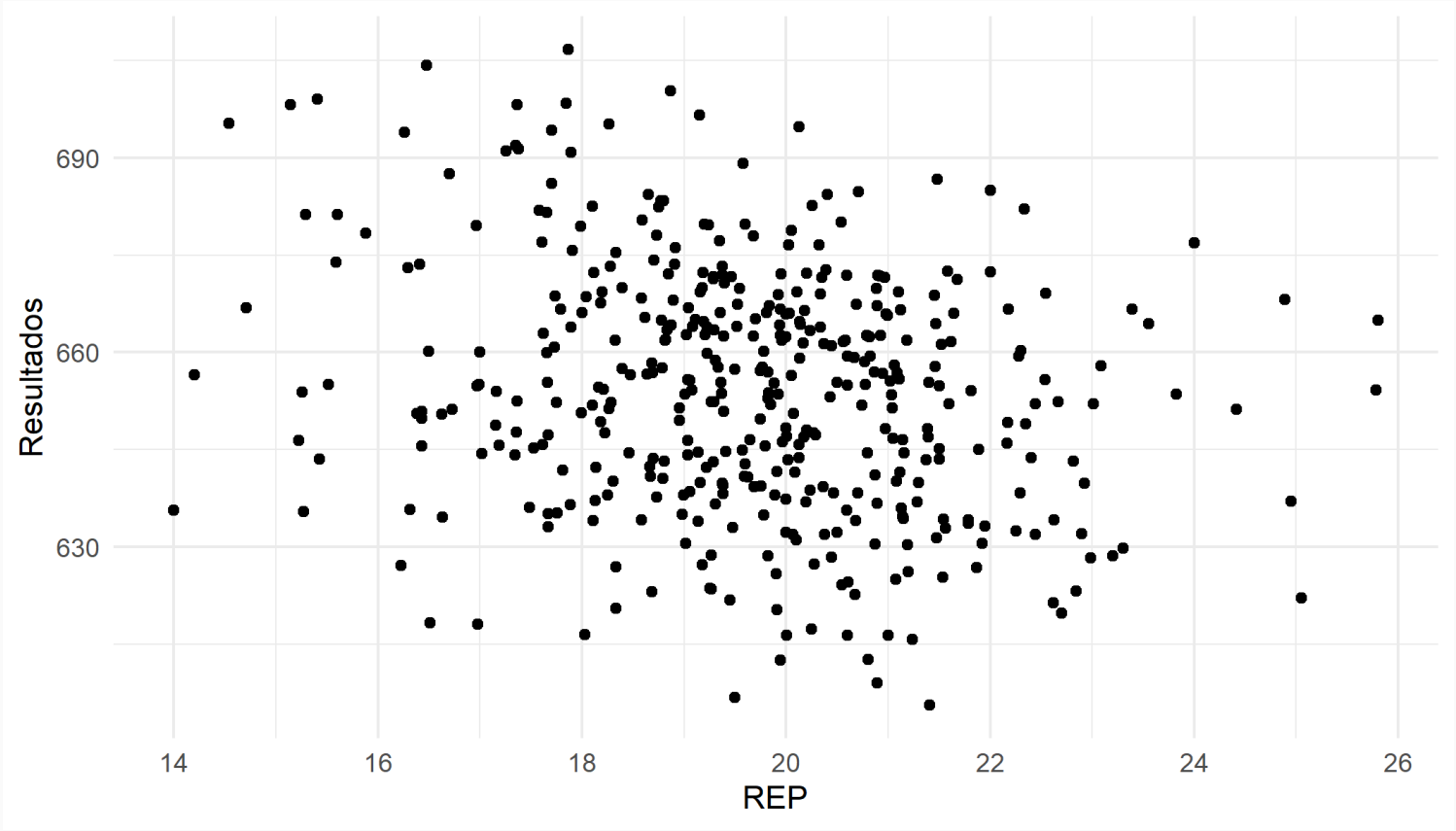
Preparar datos

```
(datos_reg ← CASchools %>%
  rename(ingresos = income) %>%
  transmute(district, school,
            Resultados = (read + math)/2,
            REP = students/teachers,
            ingresos,
            grupo_ingresos = as.factor(ifelse(ingresos ⩾ median(.$ingresos), 1, 0))))
```

```
##     district                              school Resultados      REP
## 1      75119                  Sunol Glen Unified     690.80 17.88991
## 2      61499                Manzanita Elementary     661.20 21.52466
## 3      61549            Thermalito Union Elementary     643.60 18.69723
## 4      61457      Golden Feather Union Elementary     647.70 17.35714
## 5      61523             Palermo Union Elementary     640.85 18.67133
## 6      62042              Burrel Union Elementary     605.55 21.40625
## 7      68536                Holt Union Elementary     606.75 19.50000
## 8      63834                 Vineland Elementary     609.00 20.89412
## 9      62331             Orange Center Elementary     612.50 19.94737
## 10     67306          Del Paso Heights Elementary     612.65 20.80556
## 11     65722            Le Grand Union Elementary     615.75 21.23810
## 12     62174               West Fresno Elementary     616.30 21.00000
## 13     71795               Allensworth Elementary     616.30 20.60000
```

Relación entre variables

```
datos_reg %>%
  ggplot(aes(REP, Resultados)) +
  geom_point() +
  theme_minimal()
```

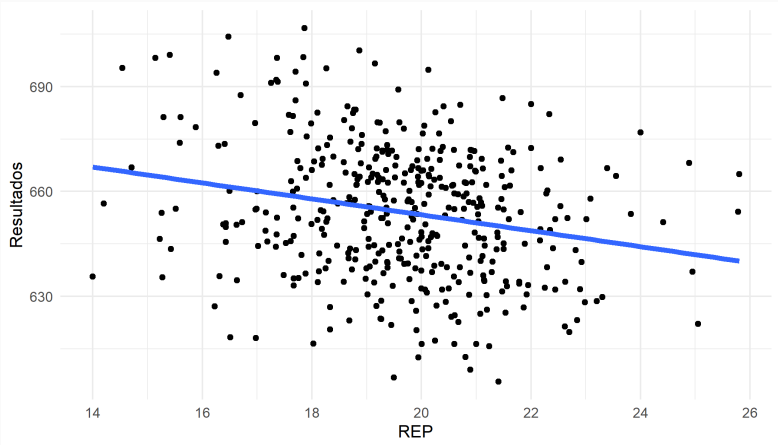


Graficar la curva de regresión

La regresión lineal simple a estimar sería $Resultados_i = b_0 + b_1 * REP_i$

- `ggplot` nos permite graficar esta curva de forma simple con `geom_smooth`
  - `method = lm`: curva representando *linear model*
  - `se = FALSE`: sin intervalo de confianza

```
datos_reg %>%
  ggplot(aes(REP, Resultados)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE,
  theme_minimal()
```



¿Cuáles son los coeficientes? (b_0 , b_1)

Estimar coeficientes "a mano"

$$b_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}; \;\; b_0 = \bar{Y} - b_1\bar{X}$$

```
beta1 ← sum((datos_reg$REP - mean(datos_reg$REP)) * (datos_reg$Resultado - mean(datos
  sum((datos_reg$REP - mean(datos_reg$REP))^2)
beta0 ← mean(datos_reg$Resultados) - (beta1*mean(datos_reg$REP))

round(c(beta0, beta1), 4)
```

```
## [1] 698.9329  -2.2798
```

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\sum(Y_i - \bar{Y})^2}$$

```
SCR ← sum(((beta0 + (beta1*datos_reg$REP)) - mean(datos_reg$Resultados))^2)
SCT ← sum((datos_reg$Resultados - mean(datos_reg$Resultados))^2)
R2 ← SCR/SCT
round(R2, 5)
```

```
## [1] 0.05124
```

Por suerte R lo hace más simple

```
modelo1 ← lm(Resultados ~ REP, data = datos_reg)
summary(modelo1)
```

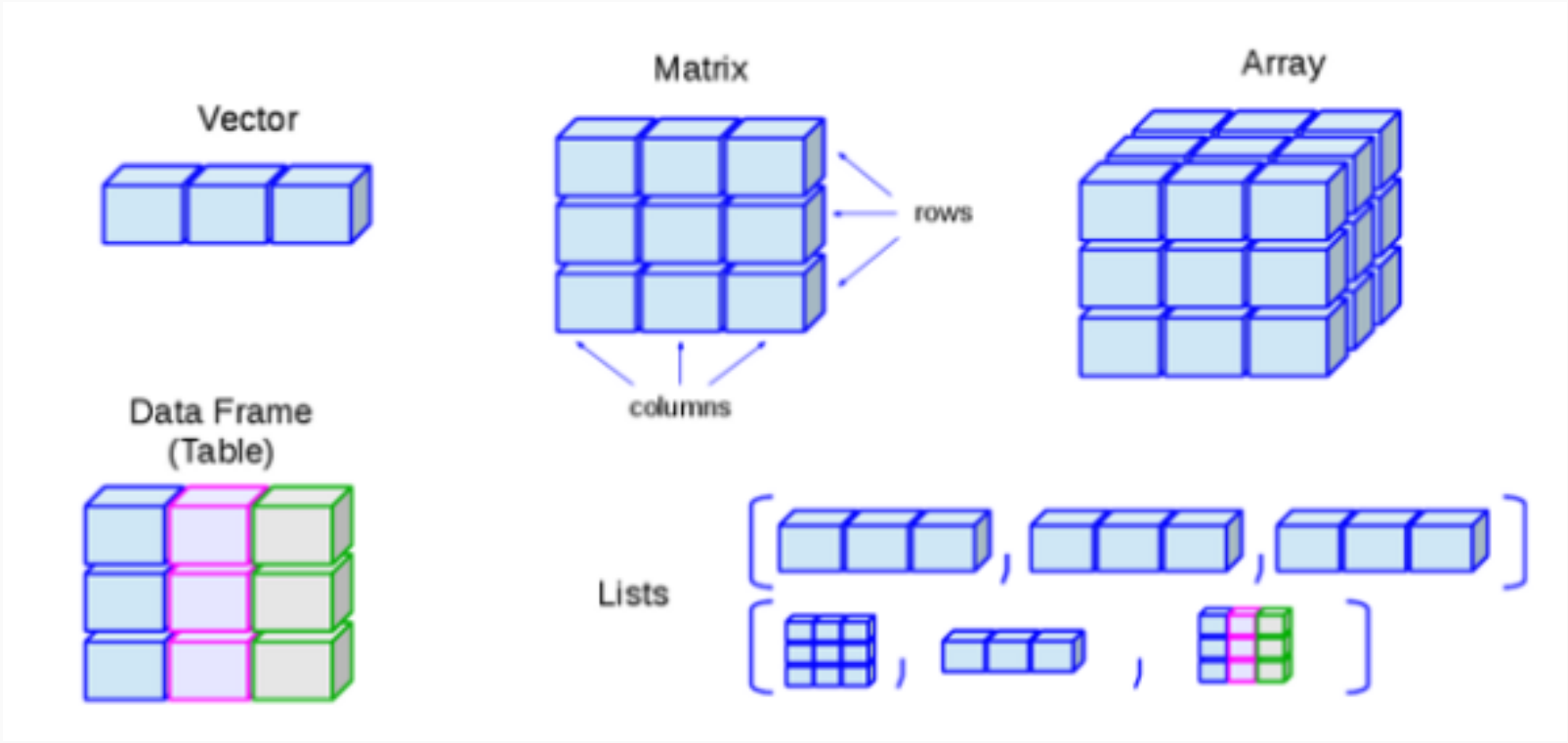
```
##
## Call:
## lm(formula = Resultados ~ REP, data = datos_reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.727 -14.251   0.483  12.822  48.540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 698.9329     9.4675  73.825  < 2e-16 ***
## REP          -2.2798     0.4798  -4.751 2.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.58 on 418 degrees of freedom
## Multiple R-squared:  0.05124,    Adjusted R-squared:  0.04897
## F-statistic: 22.58 on 1 and 418 DF,  p-value: 2.783e-06
```

¿Qué es esto?

```
str(modelo1)
```

```
## List of 12
##  $ coefficients : Named num [1:2] 698.93 -2.28
##   ..- attr(*, "names")= chr [1:2] "(Intercept)" "REP"
##  $ residuals    : Named num [1:420] 32.7 11.3 -12.7 -11.7 -15.5 ...
##   ..- attr(*, "names")= chr [1:420] "1" "2" "3" "4" ...
##  $ effects      : Named num [1:420] -13406.2 88.3 -14 -12.6 -16.8 ...
##   ..- attr(*, "names")= chr [1:420] "(Intercept)" "REP" "" "" ...
##  $ rank         : int 2
##  $ fitted.values: Named num [1:420] 658 650 656 659 656 ...
##   ..- attr(*, "names")= chr [1:420] "1" "2" "3" "4" ...
##  $ assign       : int [1:2] 0 1
##  $ qr           :List of 5
##   ..$ qr   : num [1:420, 1:2] -20.4939 0.0488 0.0488 0.0488 0.0488 ...
##   ..  ..- attr(*, "dimnames")=List of 2
##   ..  ..  ..$ : chr [1:420] "1" "2" "3" "4" ...
##   ..  ..  ..$ : chr [1:2] "(Intercept)" "REP"
##   ..  ..- attr(*, "assign")= int [1:2] 0 1
##   ..$ qraux: num [1:2] 1.05 1.05
##   ..$ pivot: int [1:2] 1 2
##   ..$ tol  : num 1e-07
##   ..$ rank : int 2
```

Tipos de objetos



Paquete broom

```
library(broom)
tidy(modelo1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   699.       9.47      73.8  6.57e-242
## 2 REP            -2.28     0.480     -4.75 2.78e-  6
```

```
glance(modelo1)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1    0.0512        0.0490  18.6      22.6 2.78e-6     1 -1822. 3650. 3663.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Paquete `broom` (cont.)

```
augment(modelo1)
```

```
## # A tibble: 420 x 8
##    Resultados   REP .fitted .resid .std.resid    .hat .sigma  .cooksd
##         <dbl> <dbl>   <dbl>  <dbl>      <dbl>   <dbl>  <dbl>    <dbl>
##  1       691.  17.9    658.   32.7       1.76  0.00442   18.5 0.00689
##  2       661.  21.5    650.   11.3       0.612 0.00475   18.6 0.000893
##  3       644.  18.7    656.  -12.7      -0.685 0.00297   18.6 0.000700
##  4       648.  17.4    659.  -11.7      -0.629 0.00586   18.6 0.00117
##  5       641.  18.7    656.  -15.5      -0.836 0.00301   18.6 0.00105
##  6       606.  21.4    650.  -44.6      -2.40  0.00446   18.5 0.0130
##  7       607.  19.5    654.  -47.7      -2.57  0.00239   18.5 0.00794
##  8       609   20.9    651.  -42.3      -2.28  0.00343   18.5 0.00895
##  9       612.  19.9    653.  -41.0      -2.21  0.00244   18.5 0.00597
## 10       613.  20.8    652.  -38.9      -2.09  0.00329   18.5 0.00723
## #  ...  with 410 more rows
```

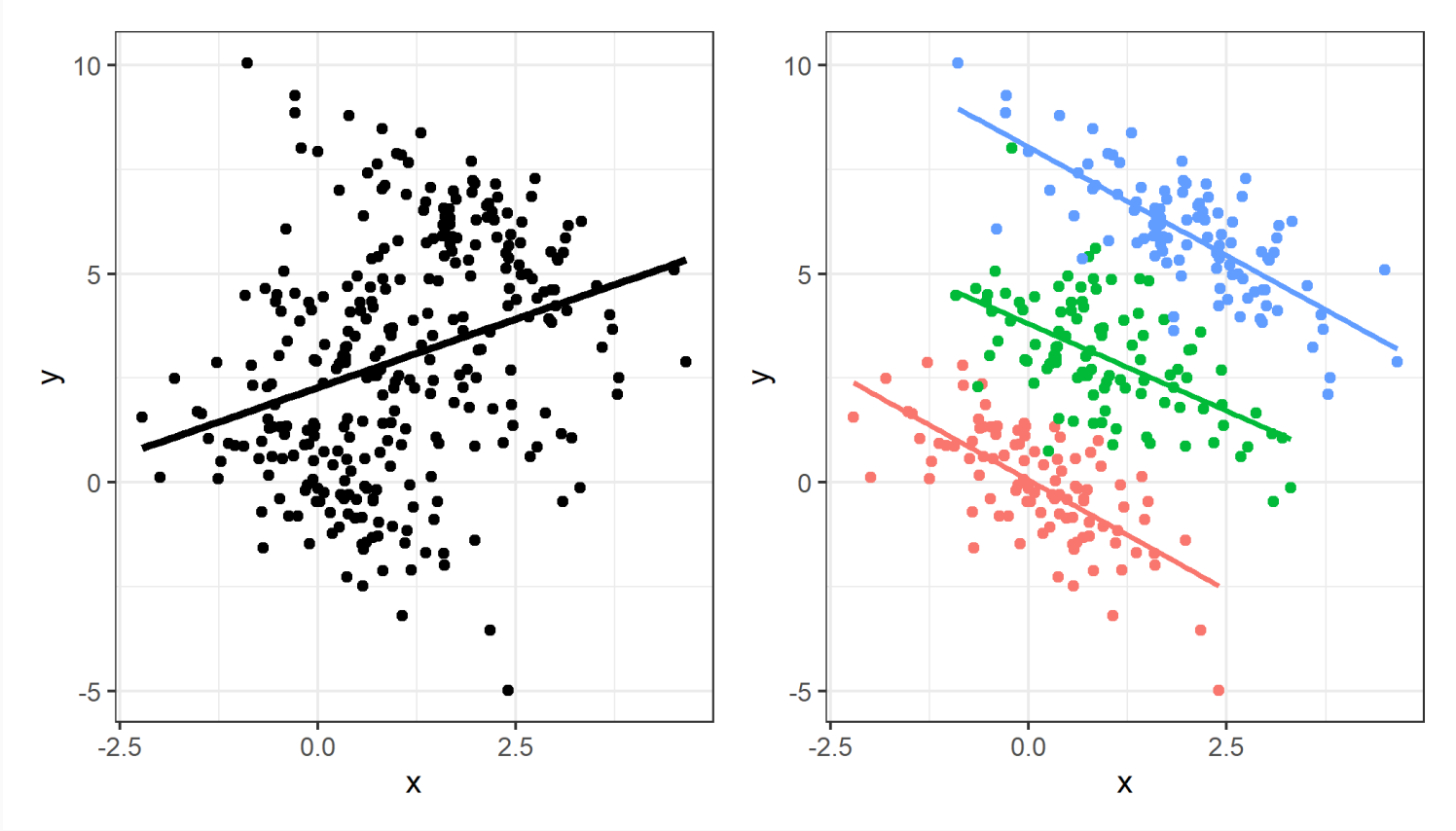
Regresión Lineal Múltiple

- ¿Es factible que solo `REP` influya en `Resultados`?

$$Resultados_i = b_0 + b_1 REP + b_2 A + b_3 B + \ldots + b_n Z$$

- ¿Qué pasa si no se incluyen otras variables relacionadas?

Paradoja de Simpson



- Sesgo de variable omitida

Regresión Lieal Múltiploe (cont)

¿Cómo se estiman los parámetros?

En forma matricial:

$$Y = X\beta + \epsilon$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Lo que debemos hacer es estimar el vector de parámetros $\hat{\beta}$

$$\hat{\beta} = (X'X)^{-1}X'Y$$

Terminando finalmente con $\hat{Y} = X\hat{\beta}$

Por suerte R lo hace más simple

$$Resultados_i = b_0 + b_1 * REP_i + b_2 * ingresos_i$$

```
modelo2 ← lm(Resultados ~ REP + ingresos, data = datos_reg)
summary(modelo2)
```

```
##
## Call:
## lm(formula = Resultados ~ REP + ingresos, data = datos_reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.608  -9.052   0.707   9.259  31.898
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 638.72916    7.44908  85.746   <2e-16 ***
## REP          -0.64874    0.35440  -1.831   0.0679 .
## ingresos      1.83911    0.09279  19.821   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.35 on 417 degrees of freedom
## Multiple R-squared:  0.5115,    Adjusted R-squared:  0.5091
```

Comparemos

Comparemos los valores de R^2 para `modelo1` (simple) y `modelo2` (múltiple)

```
glance(modelo1)$r.squared
```

```
## [1] 0.05124009
```

```
glance(modelo2)$r.squared
```

```
## [1] 0.511483
```

- `modelo2` ajusta mejor
- Pero **OJO** con el R^2: aumentará siempre que sumemos variables

R^2 ajustado

$$R^2_{adj} = 1 - \left(\frac{SCE}{SCT} \frac{n-1}{n-k-1} \right) = 1 - \left(\frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{\sum (Y_i - \bar{Y})^2} \frac{n-1}{n-k-1} \right)$$

n es el número de observaciones y k es el número de variables independientes.

- Si la nueva variable no "aporta nueva información", R^2_{adj} no aumenta
- Debido a lo anterior, R^2_{adj} suele ser mejor para la comparación entre modelos
  - R^2_{adj} no es la única métrica de comparación

```
glance(modelo1)$adj.r.squared
```

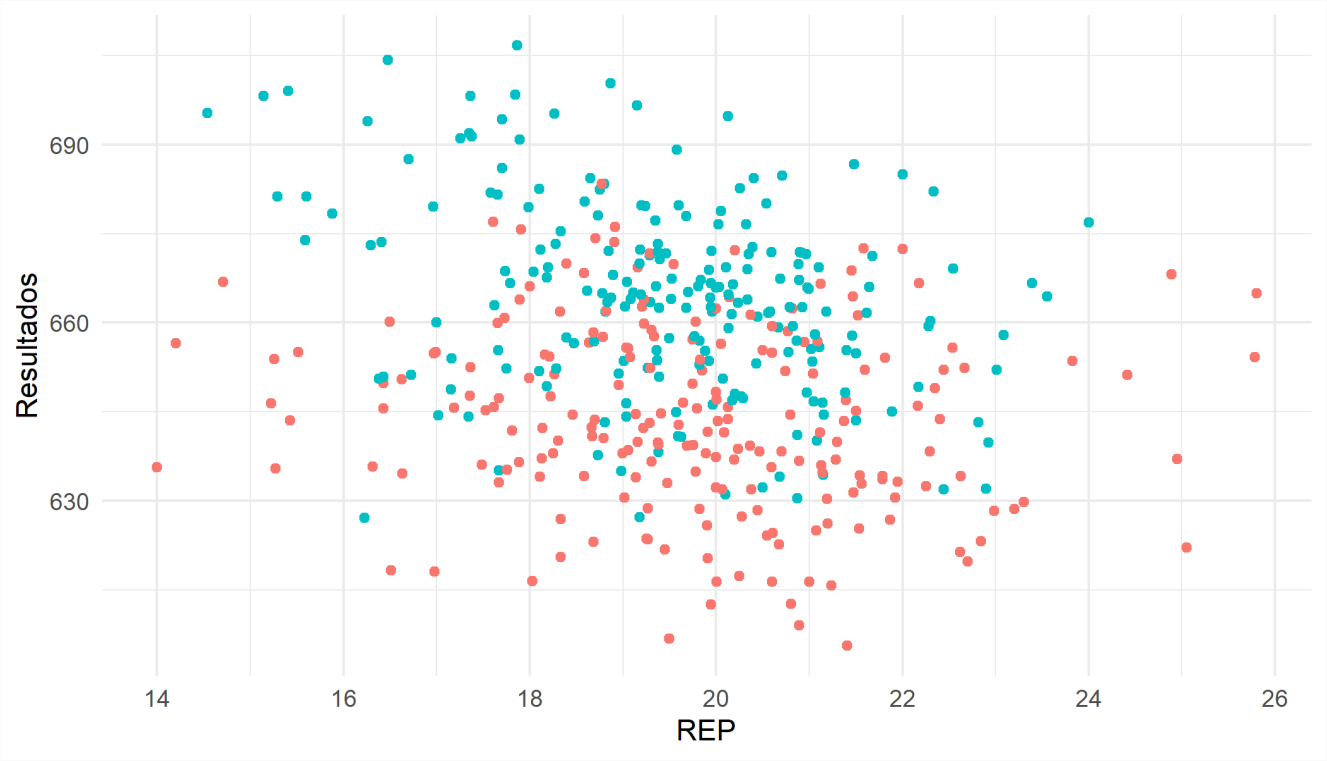
```
## [1] 0.04897033
```

```
glance(modelo2)$adj.r.squared
```

```
## [1] 0.50914
```

Modelos con interacciones

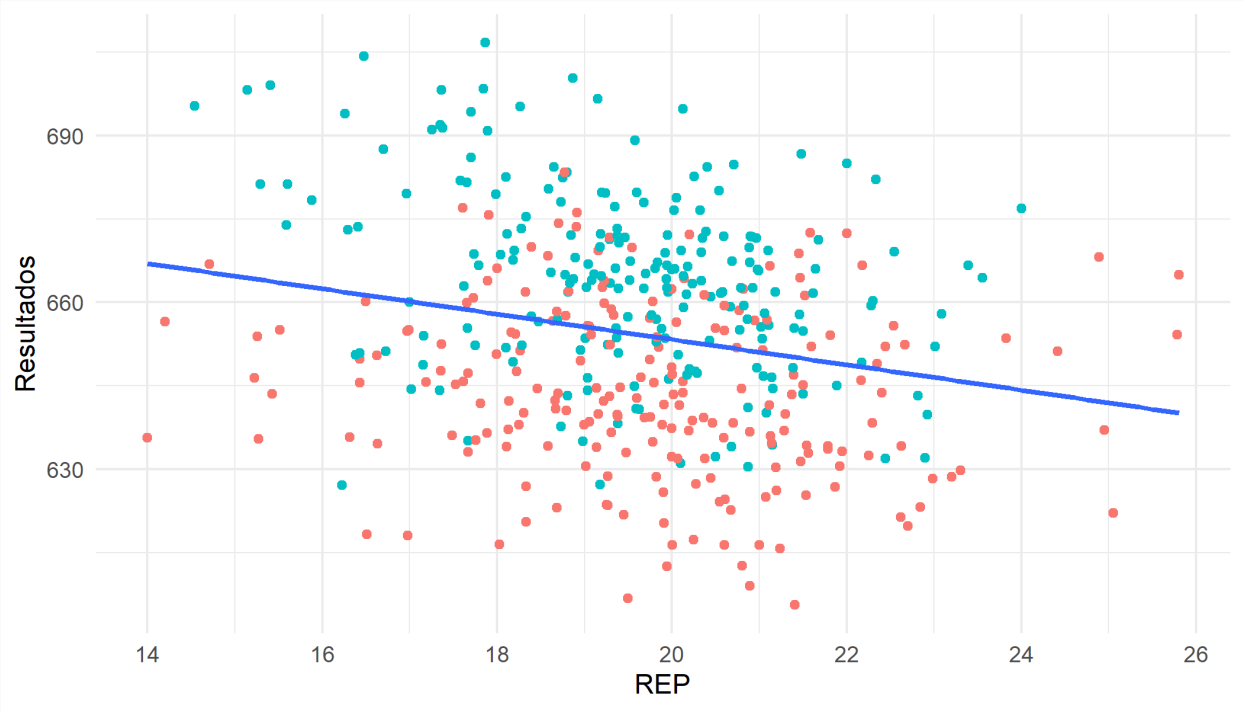
```
(base ← datos_reg %>%
  ggplot(aes(x = REP, y = Resultados)) +
  geom_point(aes(col = grupo_ingresos)) +
  theme_minimal() +
  theme(legend.position = "none"))
```



Modelos con interacciones (cont.)

$$Resultados_i = b_0 + b_1 REP_i$$

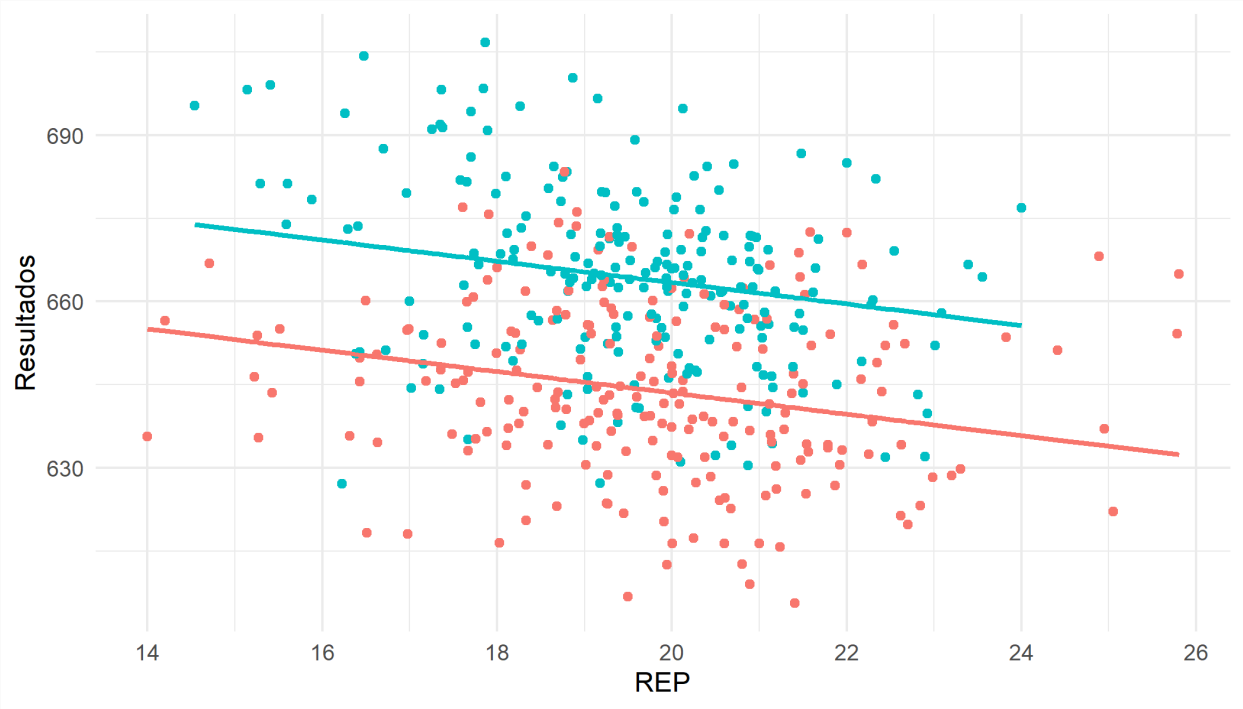
```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   699.       9.47      73.8  6.57e-242
## 2 REP            -2.28     0.480     -4.75 2.78e-  6
```



Modelos con interacciones (cont.)

$$Resultados_i = b_0 + b_1 REP_i + b_2 1_{med\_ing}$$

```
## # A tibble: 3 x 5
##   term             estimate std.error statistic   p.value
##   <chr>               <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)       682.       8.11      84.1  1.34e-263
## 2 REP                -1.92     0.407     -4.73 3.05e-  6
## 3 grupo_ingresos1    19.9      1.54      12.9  1.88e- 32
```



Modelos con interacciones (cont.)

$$Resultados_i = b_0 + b_1 REP_i + b_2 1_{med\_ing} + b_3 (REP_i * 1_{med\_ing})$$

```
## # A tibble: 4 x 5
##   term                estimate std.error statistic   p.value
##   <chr>                  <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)           664.      10.5       62.9  1.08e-214
## 2 REP                    -1.01     0.531     -1.90 5.83e-  2
## 3 grupo_ingresos1        62.5     16.1        3.88 1.21e-  4
## 4 REP:grupo_ingresos1    -2.17     0.818     -2.66 8.16e-  3
```

¿Con qué modelo nos quedamos?

- Depende...
- ¿ R^2 ?
- *¿p-value?*
- ¿Estadístico F?
- Queremos estimaciones insesgadas
  - "foco en $\hat{\beta}$"
  - Supuestos de Gauss-Markov

$$Datos \rightarrow Cálculos \rightarrow Estimación \xrightarrow{si\ todo\ sale\ bien} Verdad$$

$$X, Y \rightarrow (X'X)^{-1}X'Y \rightarrow \hat{\beta} \xrightarrow{si\ todo\ sale\ bien} \beta$$

Idea a retener

- Flexibilidad de un modelo
- Más variables $\longrightarrow$ Más Flexibilidad
- Interacciones $\longrightarrow$ Más Flexibilidad
- ¿Bueno o Malo?
- Sesgo vs Varianza (próxima clase)

Regresión Logística / Clasificación

Variable dependiente binaria

- Hasta ahora consideramos una variable dependiente Y continua (Resultados de prueba)
- Pero también podemos tener casos en que Y es una variable categórica/binaria (1 o 0)
  - Otorgamiento de crédito/subsidio
  - Ocurrencia de algún evento/episodio
  - Ingreso a la Universidad
  - ...
- Esto conlleva algunos desafíos extra a los que hemos visto hasta ahora

Datos de créditos hipotecarios

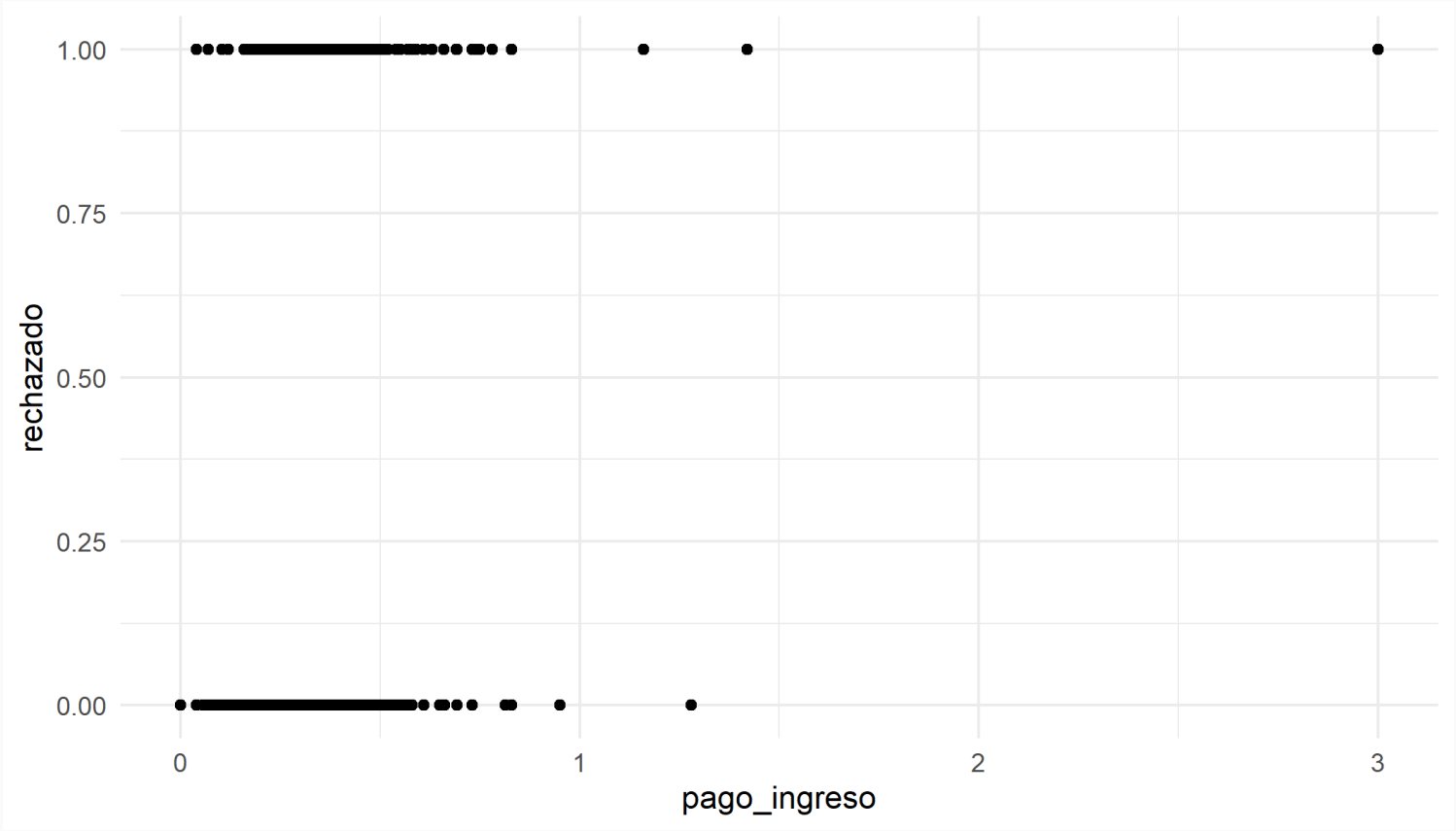
- Un crédito hipotecario puede ser aprobado o rechazado
- Uno de los principales criterios de evaluación es el ratio entre el dividendo y el sueldo

```
data(HMDA)
datos_logit ← HMDA %>%
  select(rechazado = deny, pago_ingreso = pirat) %>%
  mutate(rechazado = as.numeric(rechazado)-1)
summary(datos_logit)
```

```
##    rechazado       pago_ingreso
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.2800
##  Median :0.0000   Median :0.3300
##  Mean   :0.1197   Mean   :0.3308
##  3rd Qu.:0.0000   3rd Qu.:0.3700
##  Max.   :1.0000   Max.   :3.0000
```

¿Cómo se ve esto?

```
datos_logit %>%
  ggplot(aes(x = pago_ingreso, y = rechazado)) +
  geom_point() +
  theme_minimal()
```

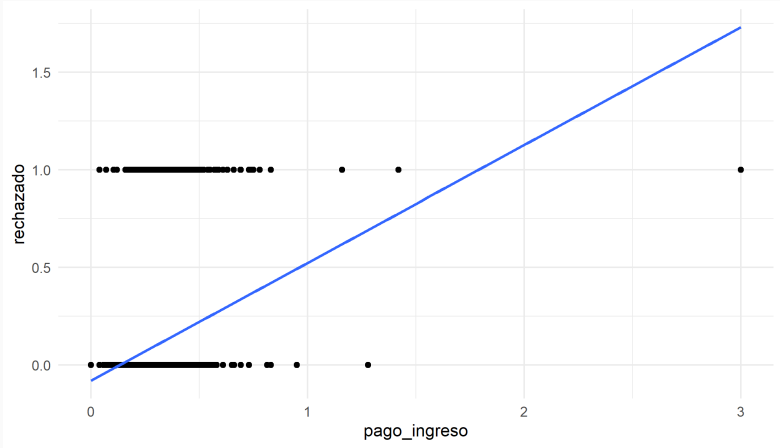


Modelo de probabilidad lineal

¿Qué ocurre si modelamos esto al igual que una regresión con Y continua?

$$P(Rechazado = 1|pago\_ingreso) = b_0 + b_1 * pago\_ingresos$$

```
datos_logit %>%
  ggplot(aes(x = pago_ingreso, y = recha
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
  theme_minimal()
```



- El modelo permite valores menores a 0 y superiores a 1. ¿Cómo interpretamos eso?
- Debemos buscar una forma de limitar los valores de Y
  - $P(Rechazado = 1|pago\_ingreso) = F(b_0 + b_1 * pago\_ingresos)$

Modelo logit

- El modelo logit (o logístico) nos permite limitar los valores de Y entre 0 y 1 usando como función auxiliar $F = \frac{exp(z)}{1+exp(z)}$ con $z = b_0 + b_1 * pago\_ingresos$.

$$P(Rechazado = 1|pago\_ingreso) = \frac{e^{(b_0+b_1*pago\_ingresos)}}{1 + e^{(b_0+b_1*pago\_ingresos)}}$$

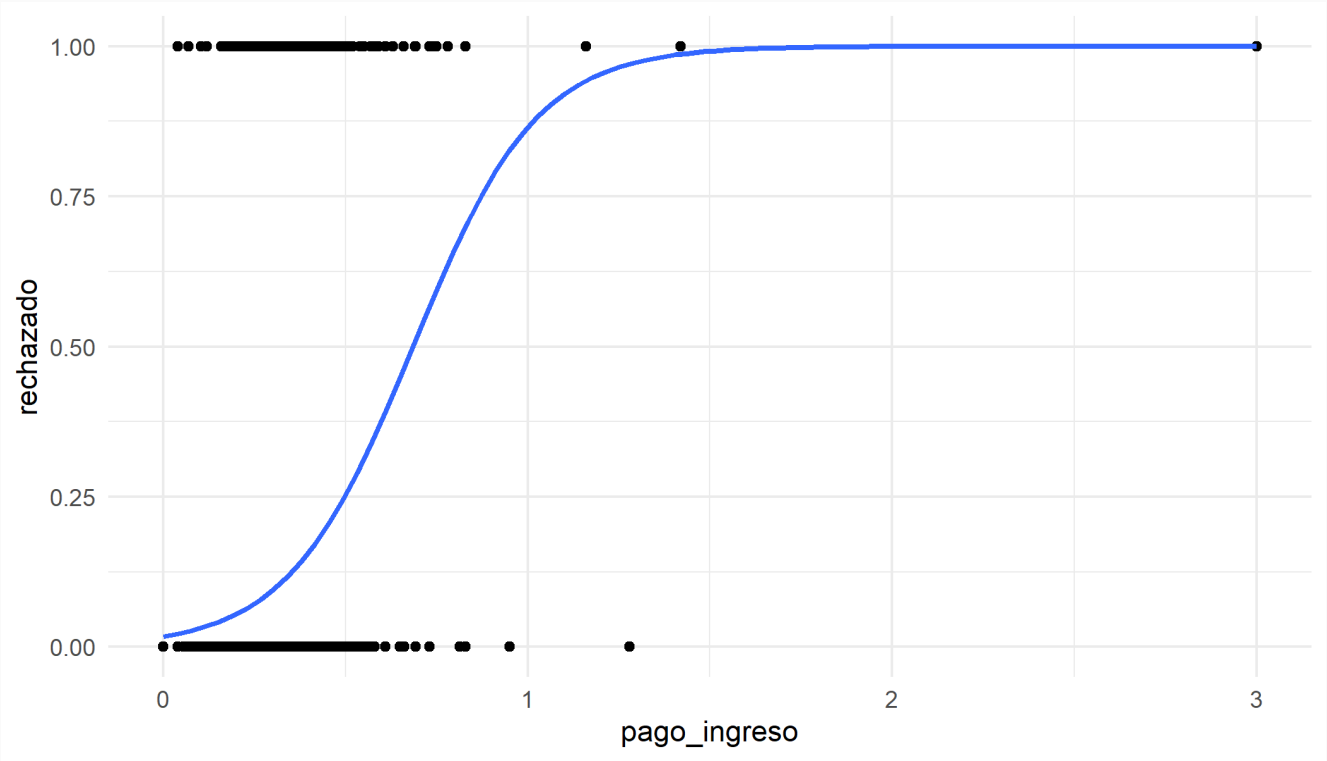
El proceso de estimación es algo distinto a lo que vimos para MCO. En este caso se hace por algo llamado máxima verosimilitud (no entraremos en detalles).

Pero en R...

```
modelo_logit ← glm(rechazado ~ pago_ingreso, family = "binomial", data = datos_logit)
```

¿Cómo se ve esto?

```
datos_logit %>%
  ggplot(aes(x = pago_ingreso, y = rechazado)) +
  geom_point() +
  geom_smooth(method = "glm", se = FALSE,
              method.args = list(family = "binomial")) +
  theme_minimal()
```



Interpretar el resultado

```
tidy(modelo_logit)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)     -4.03     0.269    -15.0  7.43e-51
## 2 pago_ingreso     5.88     0.734      8.02 1.05e-15
```

$$P(Rechazado = 1|pago\_ingreso) = \frac{e^{(-4.03+5.88*pago\_ingresos)}}{1 + e^{(-4.03+5.88*pago\_ingresos)}}$$

El efecto del ratio entre dividendo e ingresos en la probabilidad de que el crédito sea rechazado depende del "lugar de la curva" donde estemos (no es lineal).

```
predict(modelo_logit,
        newdata = data.frame("pago_ingreso" = c(0.1, 0.3, 0.5, 0.7, 1, 2)),
        type = "response") %>% round(4)
```

```
##      1      2      3      4      5      6
## 0.0311 0.0942 0.2523 0.5227 0.8648 0.9996
```

Interpretar el resultado (cont)

```
tidy(modelo_logit)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)     -4.03     0.269    -15.0  7.43e-51
## 2 pago_ingreso     5.88     0.734      8.02 1.05e-15
```

¿Y cómo se interpreta $\hat{\beta}$?

$$log\left(\frac{P(Rechazado = 1)}{P(Rechazado = 0)}\right) = -4.03 + 5.88 * pago\_ingresos$$

¿Cómo evaluamos este modelo?

Pseudo- R^2

Logit es un ejemplo de modelos de regresión no lineal y es importante destacar que en estos casos una métrica como el R^2 no tiene sentido ya que sus supuestos son que las relaciones son lineales.

Una alternativa es utilizar una métrica conocida como *pseudo-* R^2:

$$pseudo - R^2 = 1 - \frac{ln(f_{full}^{max})}{ln(f_{nulo}^{max})} = 1 - \frac{devianza}{devianza\ nula}$$

```
glance(modelo_logit)
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1         1744.    2379  -830. 1664. 1676.    1660.        2378  2380
```

```
1 - (glance(modelo_logit)$deviance/glance(modelo_logit)$null.deviance)
```

```
## [1] 0.04815042
```

¿Cómo evaluamos este modelo? (cont)

Por otro lado, ahora tenemos una probabilidad de que el crédito sea rechazado para cada observación

```
augment(modelo_logit, type.predict = "response")
```

```
## # A tibble: 2,380 x 8
##    rechazado pago_ingreso .fitted .resid .std.resid     .hat .sigma   .cooksd
##        <dbl>        <dbl>   <dbl>  <dbl>      <dbl>    <dbl>  <dbl>     <dbl>
##  1         0        0.221  0.0613 -0.356     -0.356 0.000800  0.836 0.0000262
##  2         0        0.265  0.0781 -0.403     -0.403 0.000617  0.836 0.0000262
##  3         0        0.372  0.137  -0.543     -0.543 0.000513  0.836 0.0000408
##  4         0        0.32   0.105  -0.470     -0.471 0.000456  0.836 0.0000267
##  5         0        0.36   0.129  -0.526     -0.526 0.000472  0.836 0.0000349
##  6         0        0.24   0.0681 -0.376     -0.376 0.000720  0.836 0.0000263
##  7         0        0.35   0.123  -0.511     -0.511 0.000452  0.836 0.0000316
##  8         0        0.28   0.0847 -0.421     -0.421 0.000561  0.836 0.0000260
##  9         1        0.31   0.0994  2.15       2.15  0.000474  0.835 0.00215
## 10         0        0.18   0.0488 -0.316     -0.317 0.000963  0.836 0.0000248
## # ... with 2,370 more rows
```

¿Qué criterio usamos para decidir si se clasifica como rechazado o no?

Generalmente se considera **0.5** como punto de corte

¿Cómo evaluamos este modelo? (cont)

```
estimacion_logit ← augment(modelo_logit, type.predict = "response") %>%
  transmute(rechazado = as.factor(rechazado),
            .fitted,
            clasificacion = as.factor(ifelse(.fitted ⩾ 0.5, 1, 0)))
```

```
library(tidymodels)
(matriz_confusion ← conf_mat(estimacion_logit, rechazado, clasificacion)$table)
```

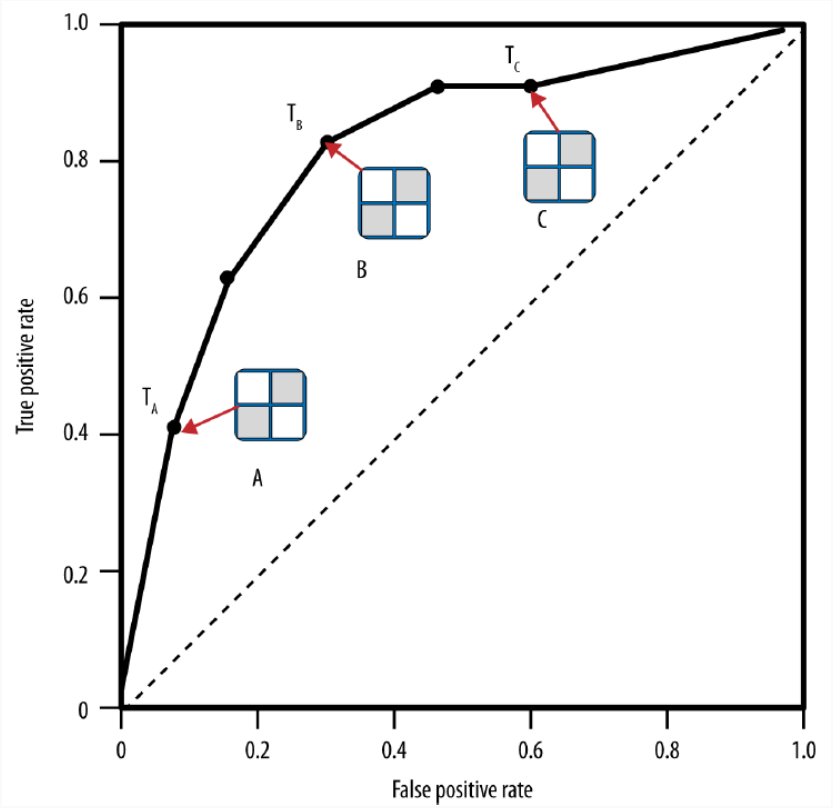
```
##           Truth
## Prediction    0    1
##          0 2094  281
##          1    1    4
```

```
VP ← matriz_confusion[2,2]
FP ← matriz_confusion[2,1]
VN ← matriz_confusion[1,1]
FN ← matriz_confusion[1,2]

(tasa_VP ← VP/(VP+FN))
## [1] 0.01403509
(tasa_FP ← FP/(FP+VN))
## [1] 0.000477327
```

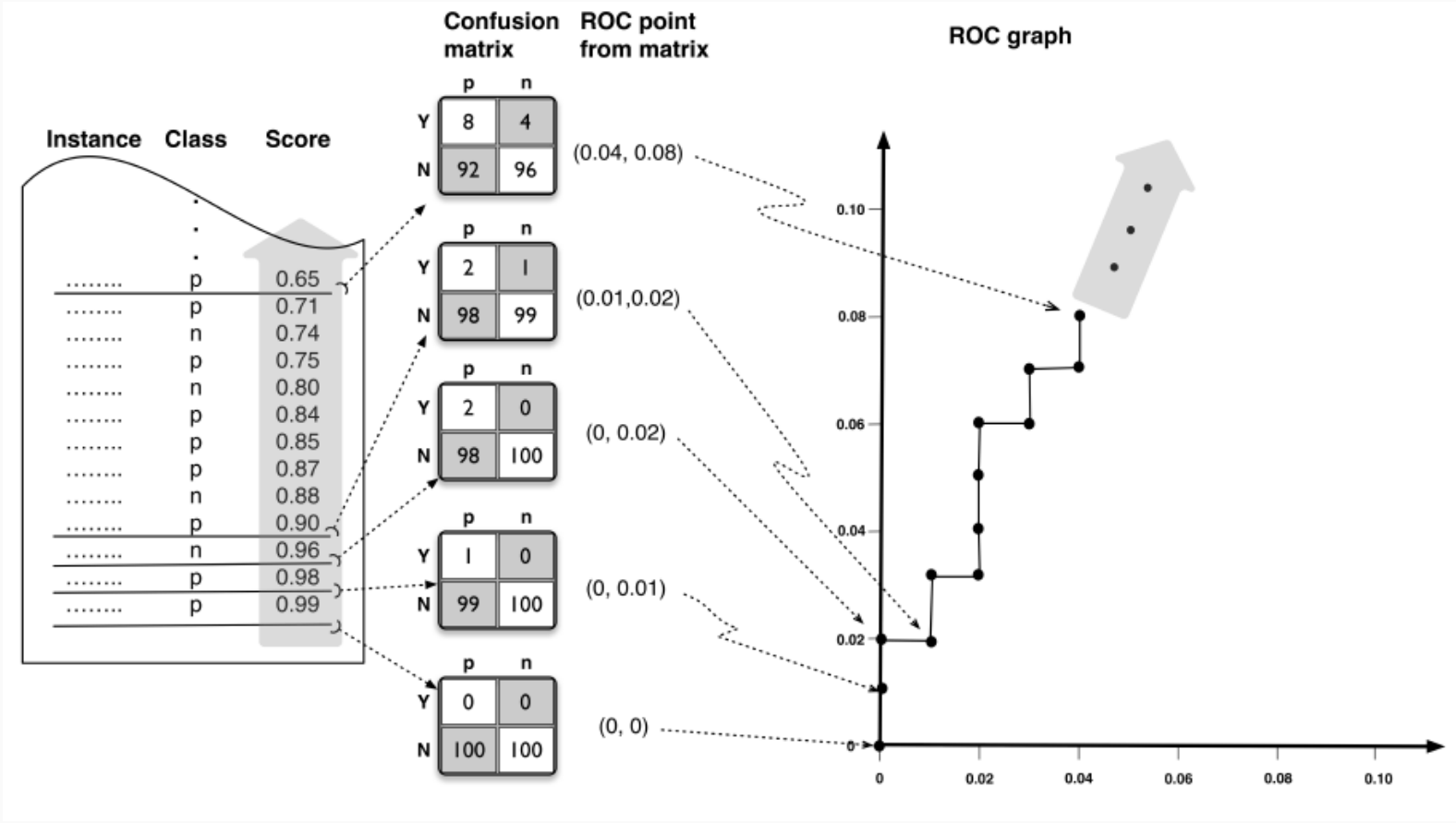
Curva ROC

Pero hasta ahora solo se considera el puntaje de corte **0.5**.

La curva ROC (Receiver Operating Characteristic) permite mostrar todo el espacio de posibilidades dependiendo de distintos puntos de cortes y mostrando el *trade-off* entre *beneficios* (verdaderos positivos) y *costos* (falsos positivos)

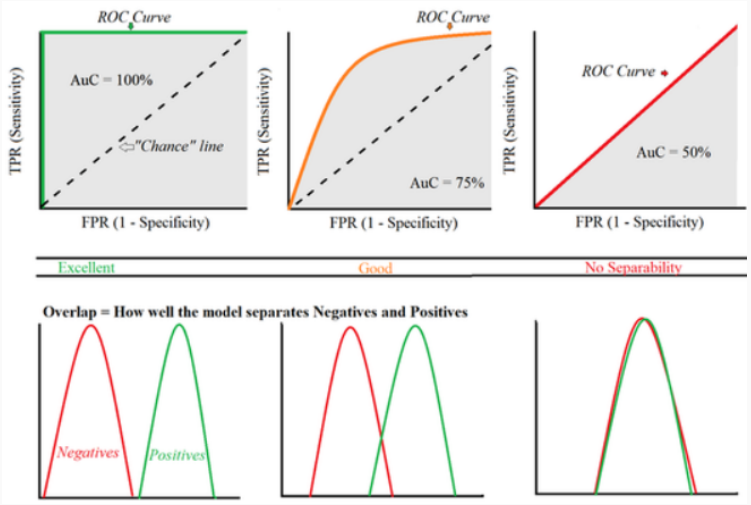


Curva ROC (cont)



Área bajo la curva (AUC)

Una métrica que nos permite resumir parte de toda la información que la curva ROC entrega es el área bajo esta misma o AUC.



En nuestro ejemplo:

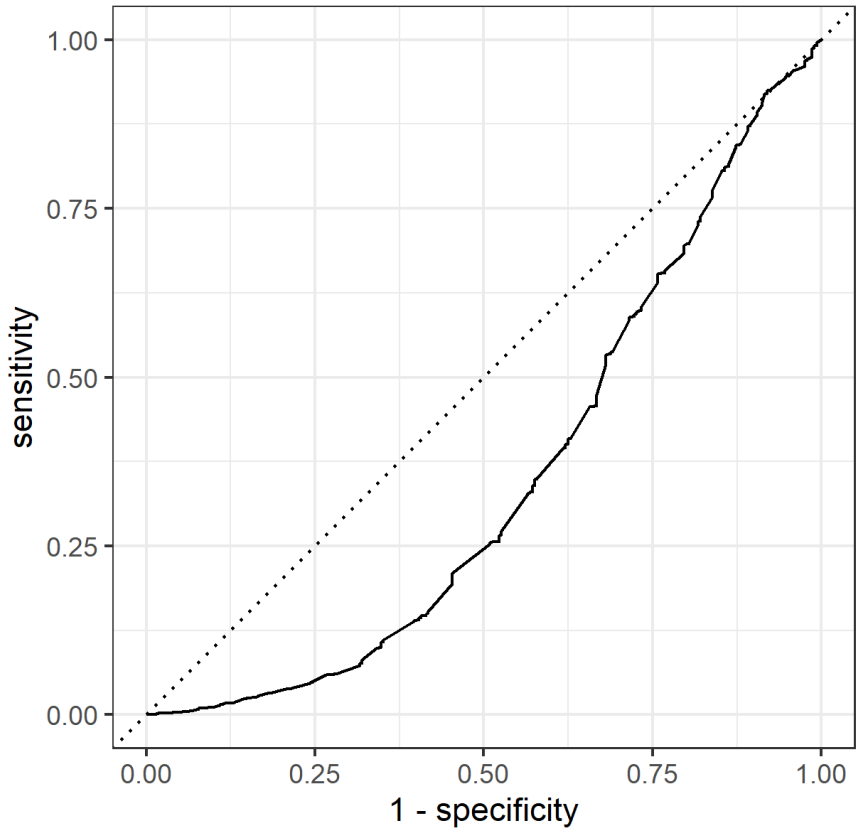
```
roc_auc(estimacion_logit, rechazado, .fitted)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.351
```

Muy mal clasificador

Nuestro modelo es peor que "tirar una moneda" para asignar la clasificación

```
roc_curve(estimacion_logit, rechazado, .fitted) %>%
  autoplot()
```



Explorar el potencial de R al modelar

- **DemoMuchasRegresiones.R**

Siguiente clase

- Predicción
- Sesgo vs varianza
- Validación cruzada (*cross-validation*)

Tarea 2 para el sábado