ITBA
Instituto Tecnológicc
de Buenos Aires

25/OCTUBRE

REGULARIZACIÓN

:D

OLS

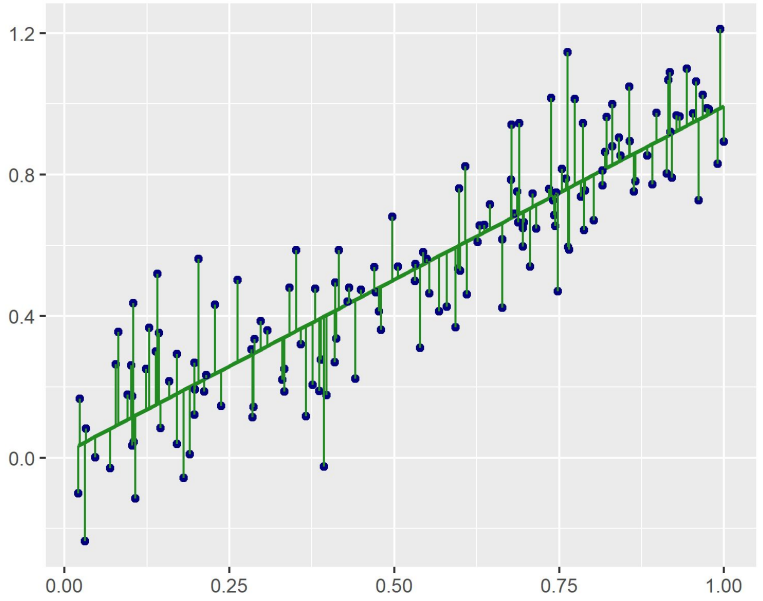
$$y = f(x) + \varepsilon \qquad y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

$$\min \boxed{SCR = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

Función de pérdida

Ordinary Least Squares (OLS / MCO)

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \ldots + \hat{\beta}_p x_{pi}$$



OLS: interpretabilidad

Simulación: 2 variables relevantes y el resto son ruido

$$y = 1 + 2x_1 + 2x_2 + \epsilon \qquad n = 960 \quad p = 802$$

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	x2	28.9	14.8	1.95	0.0533
2	x1	-10.5	16.2	-0.650	0.517
3	(Intercept)	8.14	11.2	0.729	0.467
4	xc1_32	1.28	0.429	2.97	0.00340
5	xc2_291	-1.16	0.379	-3.05	0.00268
6	xc1_211	-1.12	0.401	-2.80	0.00576
7	xc2_157	-1.11	0.384	-2.90	0.00423
8	xc1_349	1.06	0.409	2.59	0.0104
9	xc2_109	1.01	0.407	2.48	0.0140
10	xc2_396	-1.01	0.427	-2.36	0.0196

... with 793 more rows

No hay *variable selection* 😕

⇒ el modelo puede ser innecesariamente complejo si hay **muchas variables potencialmente irrelevantes**

OLS: capacidad predictiva

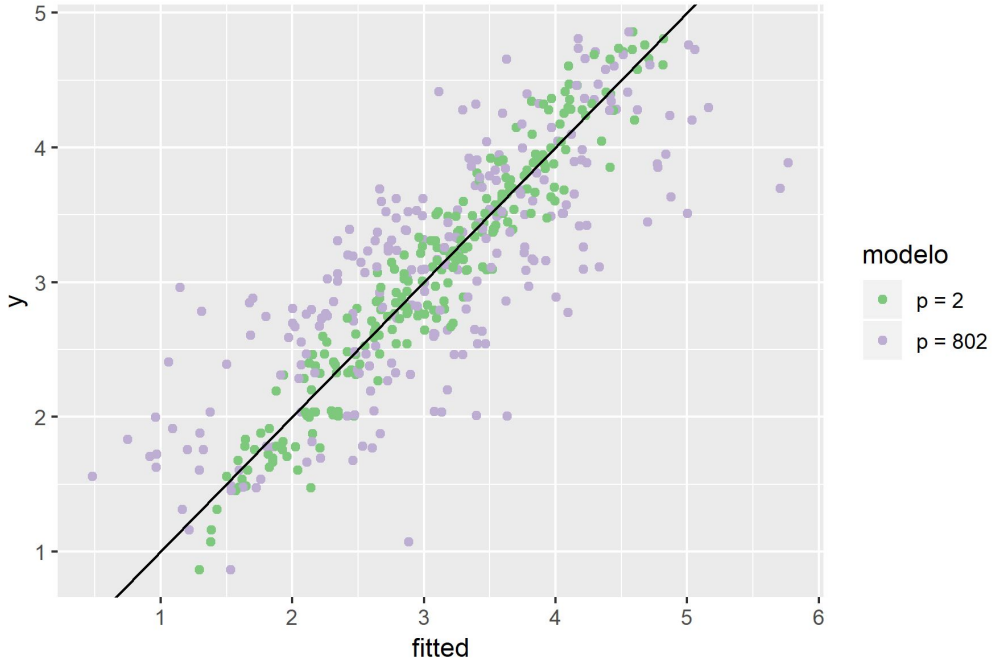
$$y = 1 + 2x_1 + 2x_2 + \epsilon$$

$$n = 960 \quad p = 802$$

p/n ≅ 0.83

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	x2	28.9	14.8	1.95	0.0533
2	x1	-10.5	16.2	-0.650	0.517
3	(Intercept)	8.14	11.2	0.729	0.467
4	xc1_32	1.28	0.429	2.97	0.00340
5	xc2_291	-1.16	0.379	-3.05	0.00268
6	xc1_211	-1.12	0.401	-2.80	0.00576
7	xc2_157	-1.11	0.384	-2.90	0.00423
8	xc1_349	1.06	0.409	2.59	0.0104
9	xc2_109	1.01	0.407	2.48	0.0140
10	xc2_396	-1.01	0.427	-2.36	0.0196

... with 793 more rows



$$RMSE_{test}^{p=2} = 3.56$$

$$RMSE_{test}^{p=802} = 10.10$$

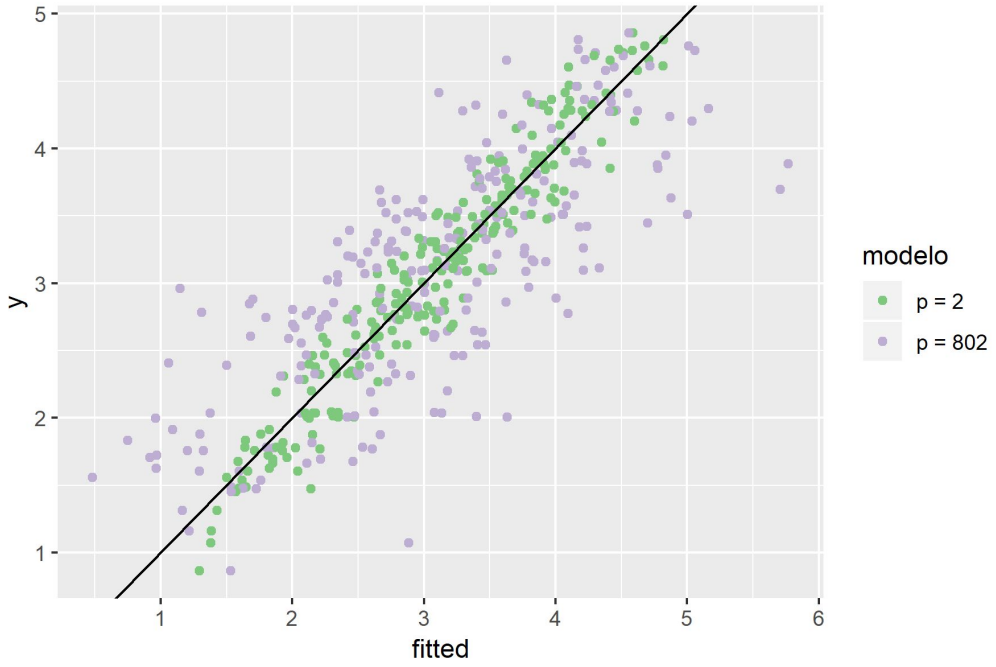
OLS: capacidad predictiva

$$y = 1 + 2x_1 + 2x_2 + \epsilon$$

$$n = 960 \quad p = 802$$

p/n ≅ 0.83

Se **deteriora el rendimiento** cuando ***p/n*** **alto** 😕



$$RMSE_{test}^{p=2} = 3.56$$

$$RMSE_{test}^{p=802} = 10.10$$

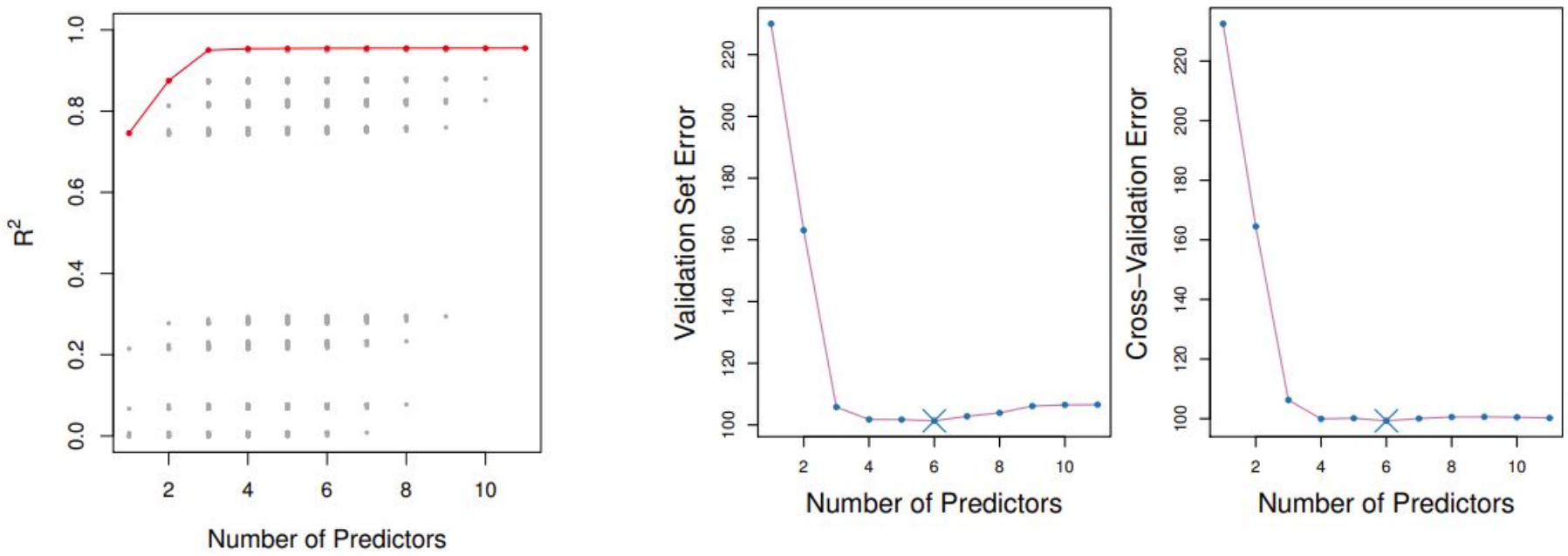
Subset selection

- **Best subset selection**

  (Probamos todos los posibles modelos)

  ⟶ Para cada p elegimos el que mejor ajusta (R^2)

  ⟶ Elegimos el modelo X de mejor ajuste en validation (o CV) entre los p restantes



Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Subset selection

Restringiendo la búsqueda:

- **Forward stepwise selection**
  → Comenzamos con el modelo nulo
  → Incluir cada covariable por separado y elegir la que mejora el ajuste
  → Repetir el paso anterior hasta incluir las p variables (terminamos con p modelos)
  → Elegimos el modelo de mejor ajuste en validation (o CV)

- **Backward stepwise selection**
  → Comenzamos con el modelo saturado
  → Sacar cada covariable por separado y eliminar la que mejora el ajuste
  → Repetir el paso anterior hasta eliminar todas las variables (terminamos con p modelos)
  → Elegimos el modelo de mejor ajuste en validation (o CV)

Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Regularización

$$\min \boxed{SCR + P}$$

Cambiamos la **función de pérdida**

P: término de penalización

RIDGE

$$\min \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j{}^2$$

$$||x||_2 = \sqrt{\sum_{j=1}^{p} x_j{}^2}$$

norma ℓ-2

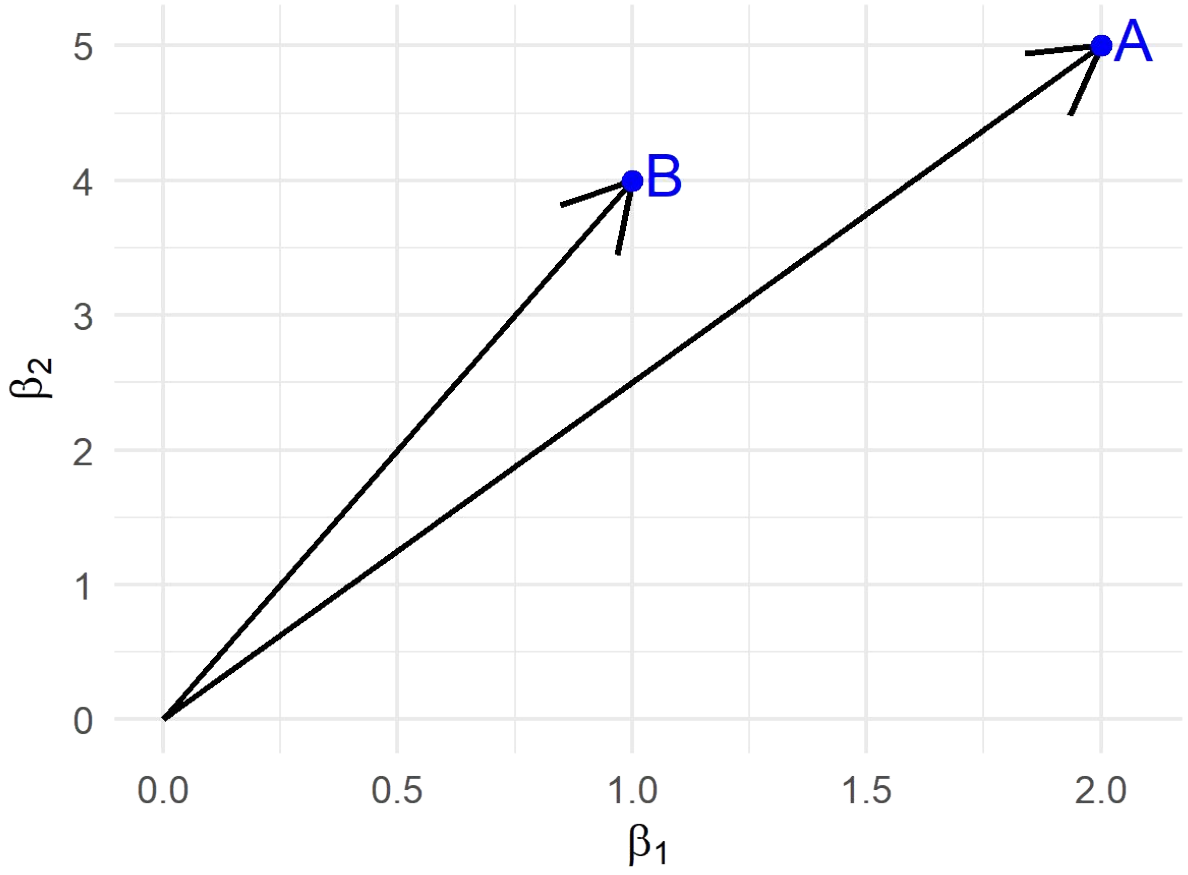
LASSO

$$\min \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$||x||_1 = \sum_{j=1}^{p} |x_j|$$

norma ℓ-1

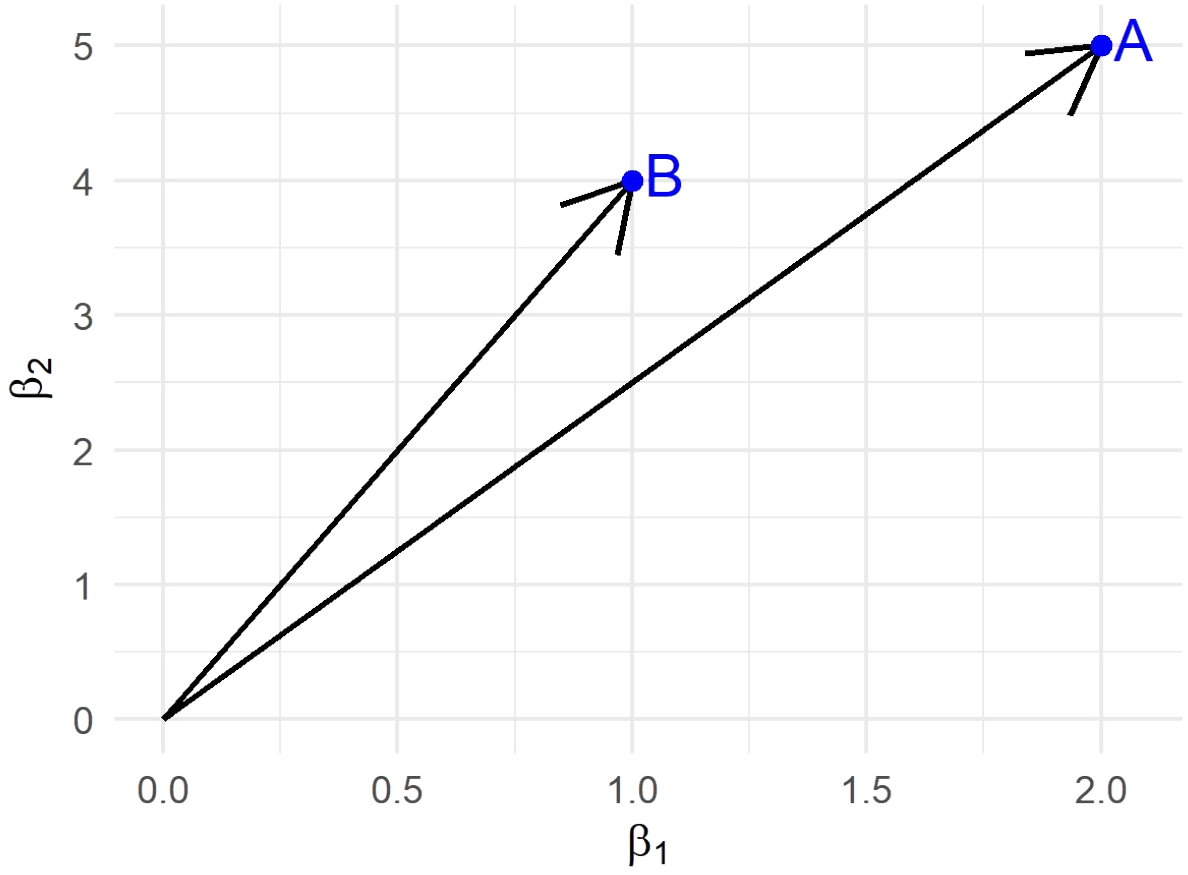
Regularización



$$\beta^A = (2, 5)$$
$$\beta_B = (1, 4)$$

¿Qué vector tiene mayor norma ℓ-1?
¿Y ℓ-2?

Regularización



$$\beta^A = (2, 5)$$
$$\beta_B = (1, 4)$$

¿Qué vector tiene mayor norma norma ℓ-1? ¿Y ℓ-2?

⇒ *La penalización tiende a restringir el tamaño de los coeficientes* ***(shrinkage)***

$$\min \; SCR + P$$

$$\min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \; + \lambda \sum_{j=1}^{p} |\beta_j|$$

Regularización

Con λ podemos controlar el peso de P

⇒ Es decir, λ es un …

$$\min \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

(¡hay que normalizar las variables!)

¿Qué pasa con los β_hat del modelo cuando …

λ = 0

$\lambda \rightarrow \infty$

?

Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Regularización

Con λ podemos controlar el peso de P

⇒ Es decir, λ es un hiperparámetro

$$\min \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

(¡hay que normalizar las variables!)

¿Qué pasa con los β_hat del modelo cuando …

λ = 0 ⇒ OLS

λ→∞ ⇒ β_hat → 0

?

Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Regularización

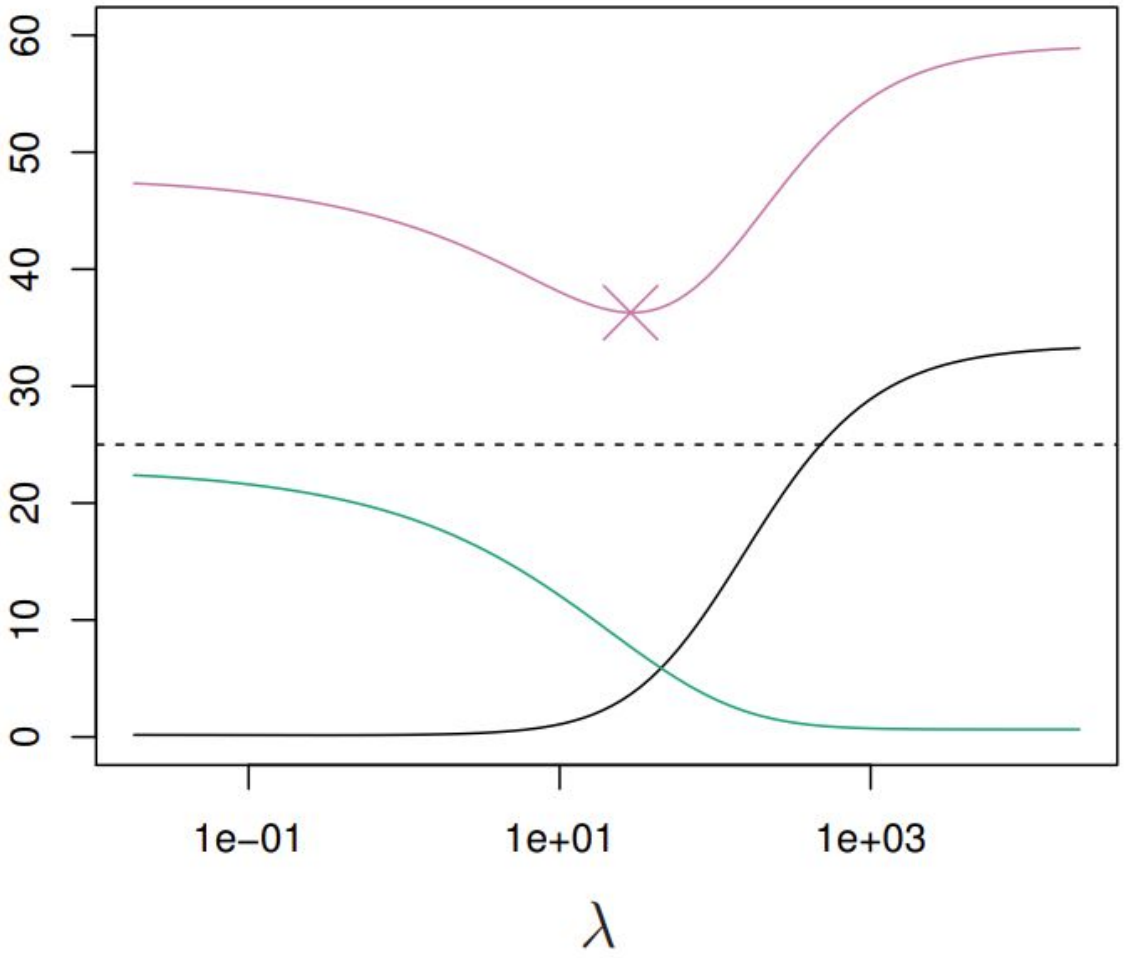
Con λ podemos controlar el peso de P

⇒ Es decir, λ es un hiperparámetro

$$\min \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

(¡hay que normalizar las variables!)

Y esto funciona! 😲💥💣😱😲😮🇸🇪



¿Qué son la línea negra y la verde?

Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Regularización

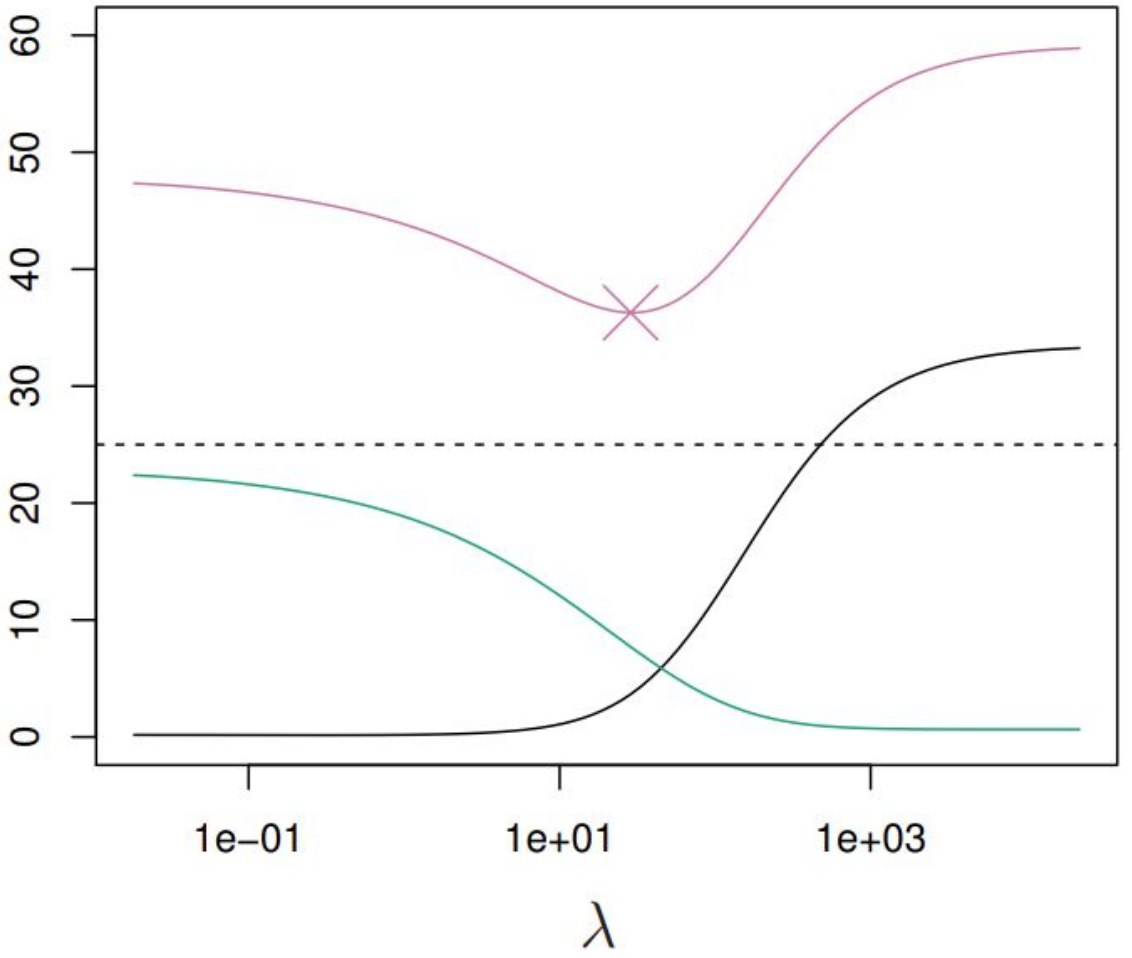
Y esto funciona! 😲💥💣😱😲😮🇸🇪

Con λ podemos controlar el peso de P

⇒ Es decir, λ es un hiperparámetro

$$\min \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

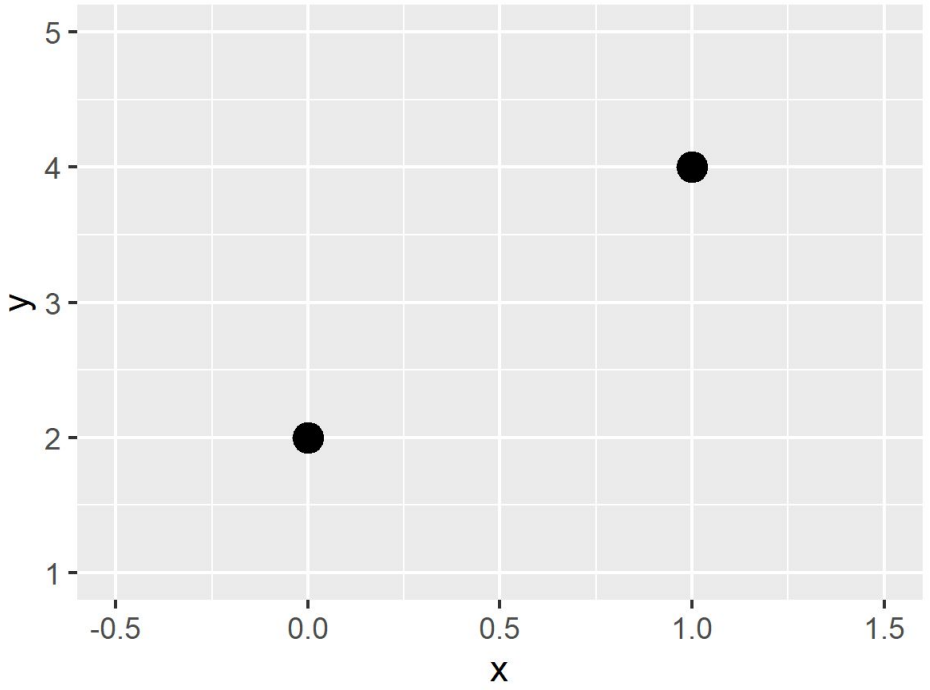
(¡hay que normalizar las variables!)



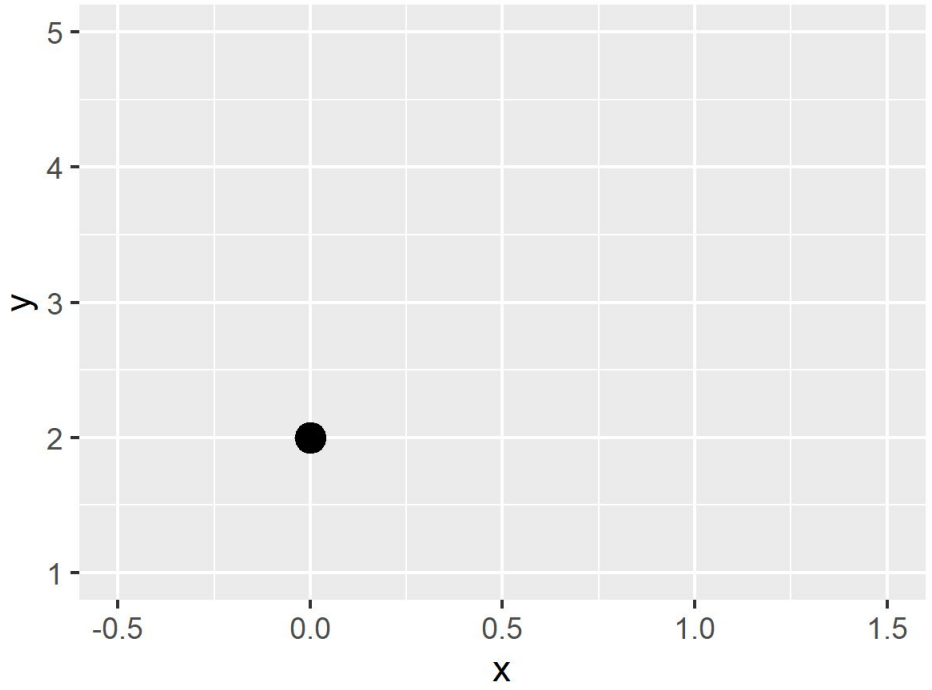
Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

OLS en *dimensionalidad alta*

p+1=2 n=2

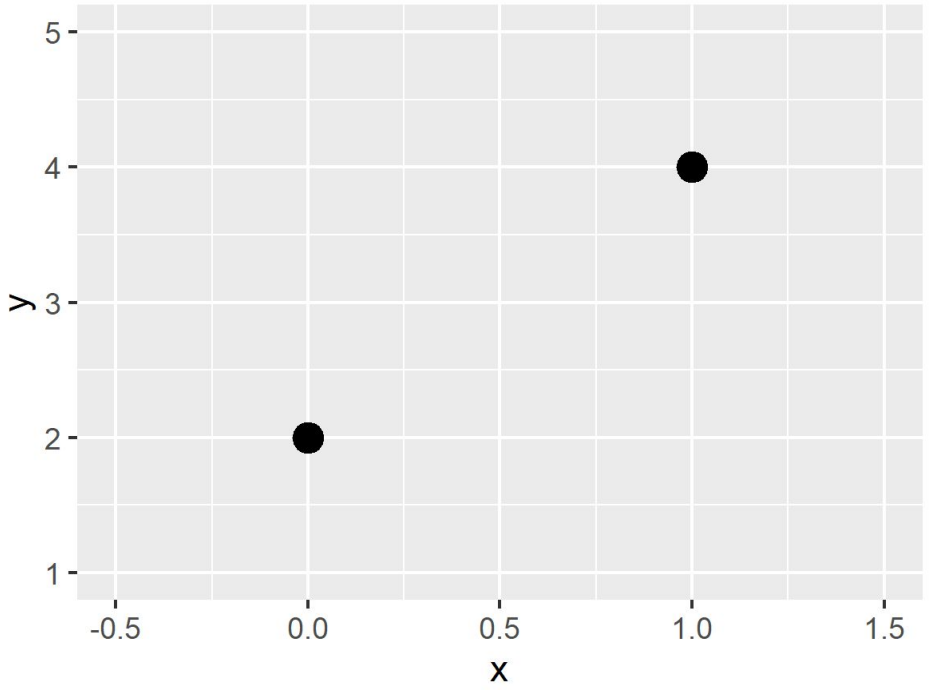


p+1=2 n=1

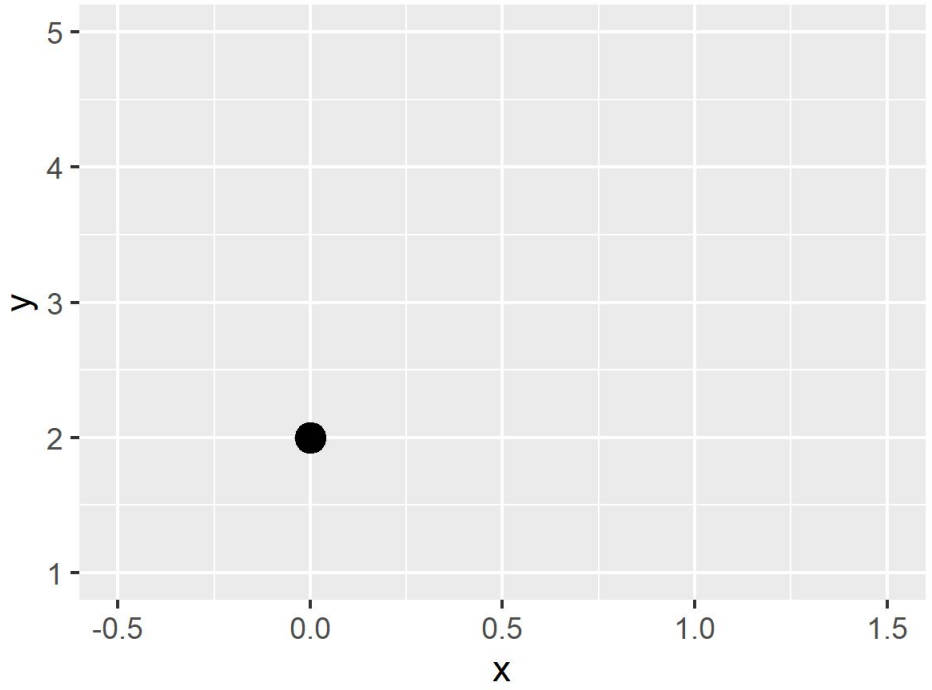


OLS en *dimensionalidad alta*

p+1=2 n=2



p+1=2 n=1



Cuando p/n es alto:

OLS puede tener mucha **varianza** (depende mucho de los puntos de entrenamiento)

⇒ ajuste bueno en train y pobre en test

Regularización

RIDGE

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s$$

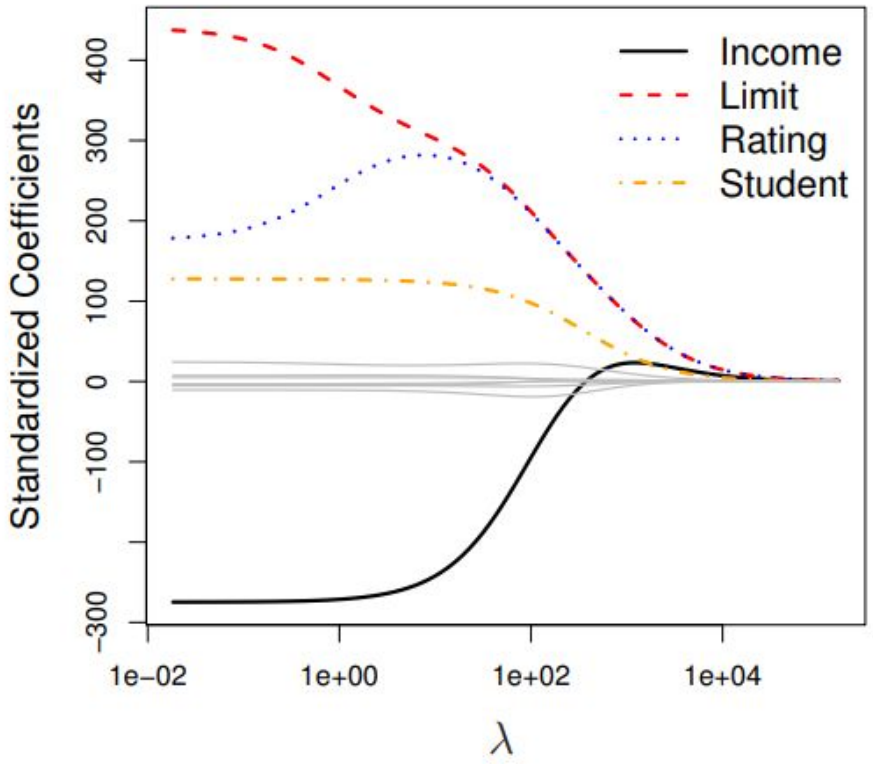
LASSO

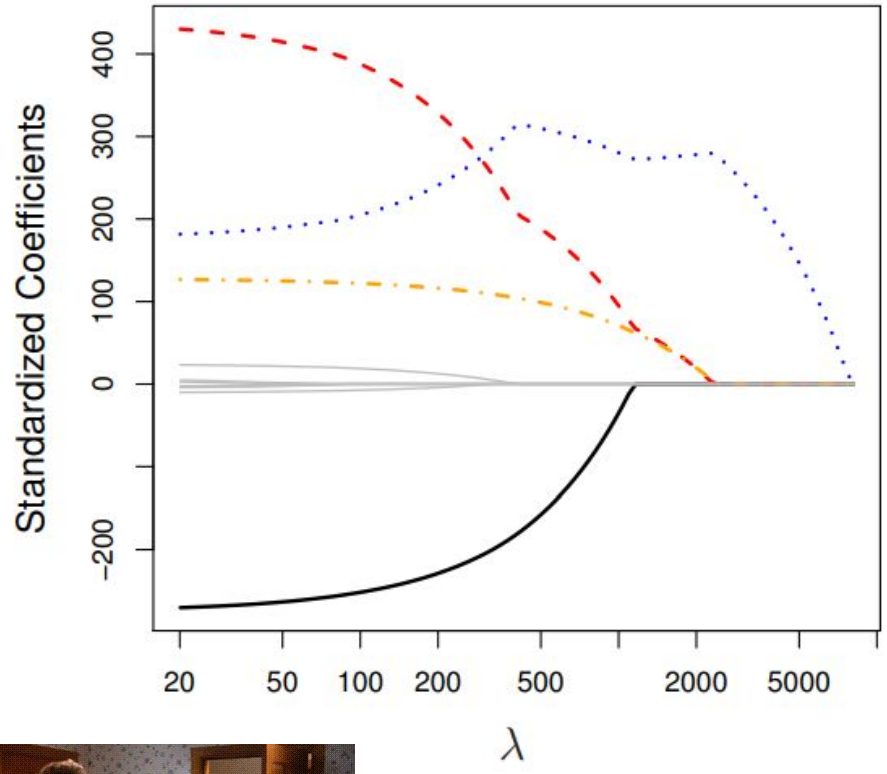
$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

Equivale a una optimización con restricción (Los coeficientes se pueden mover dentro de un presupuesto)

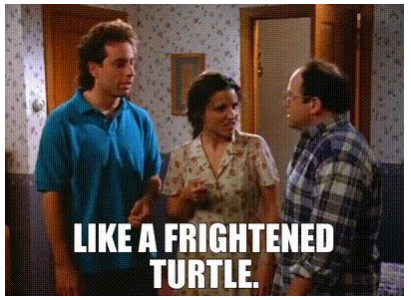
Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Ridge vs Lasso



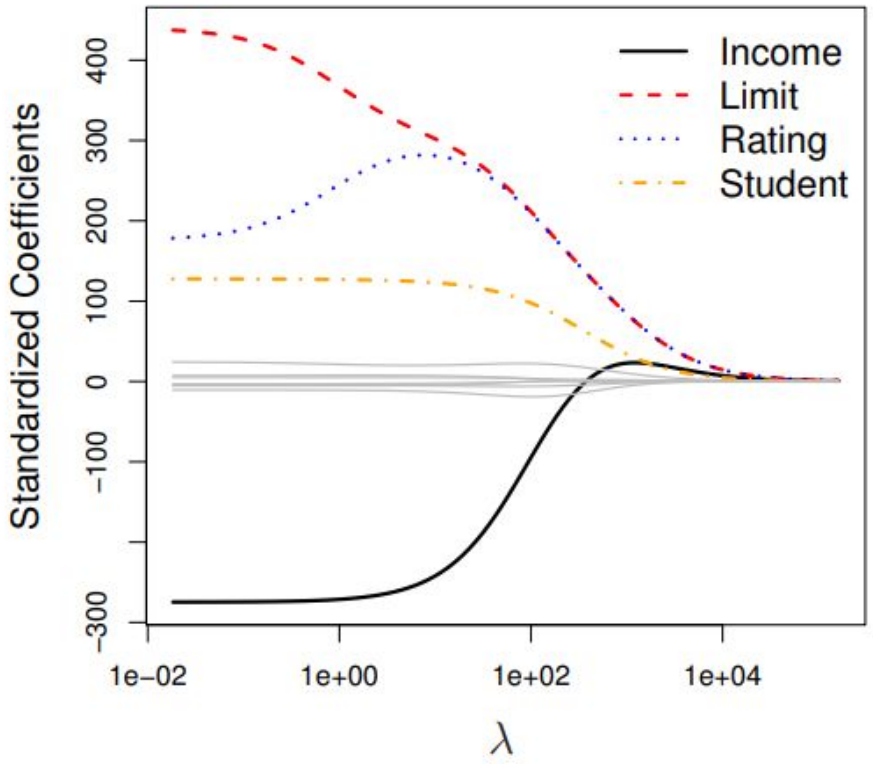


shrinkage!

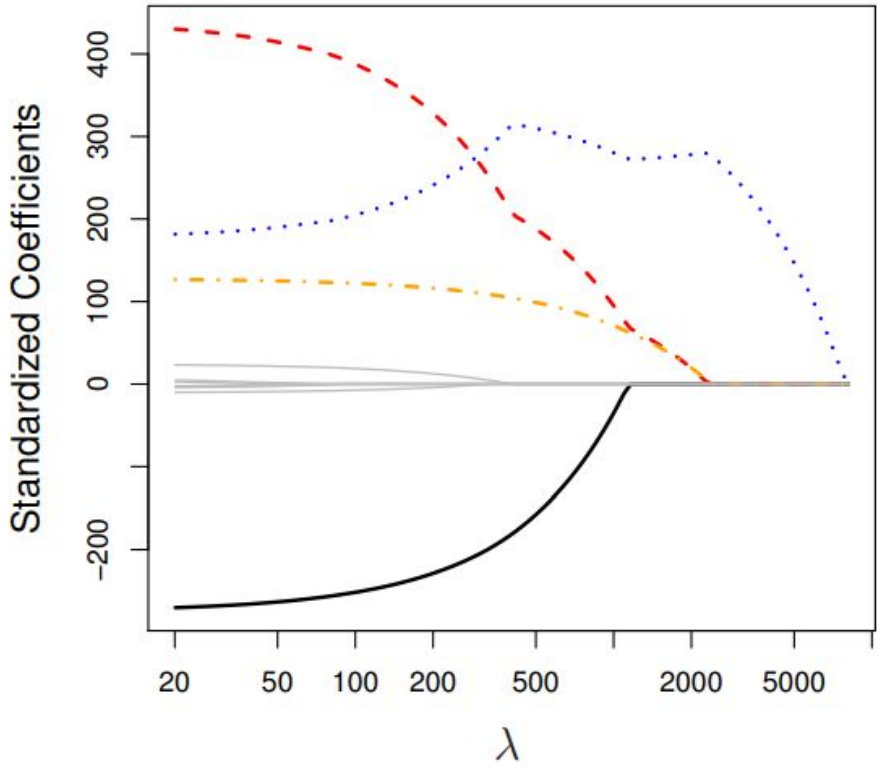


Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Ridge vs Lasso



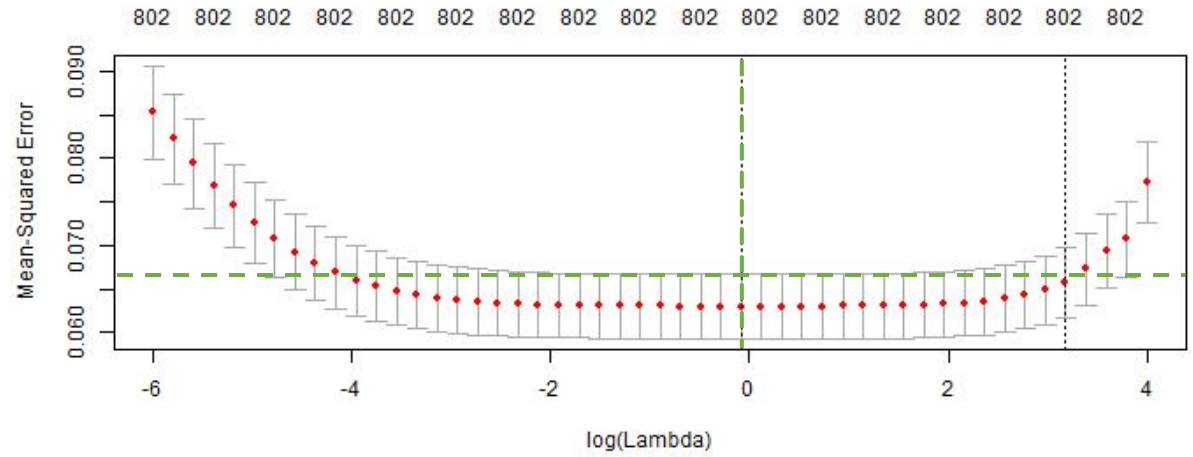
No hay coeficientes = 0

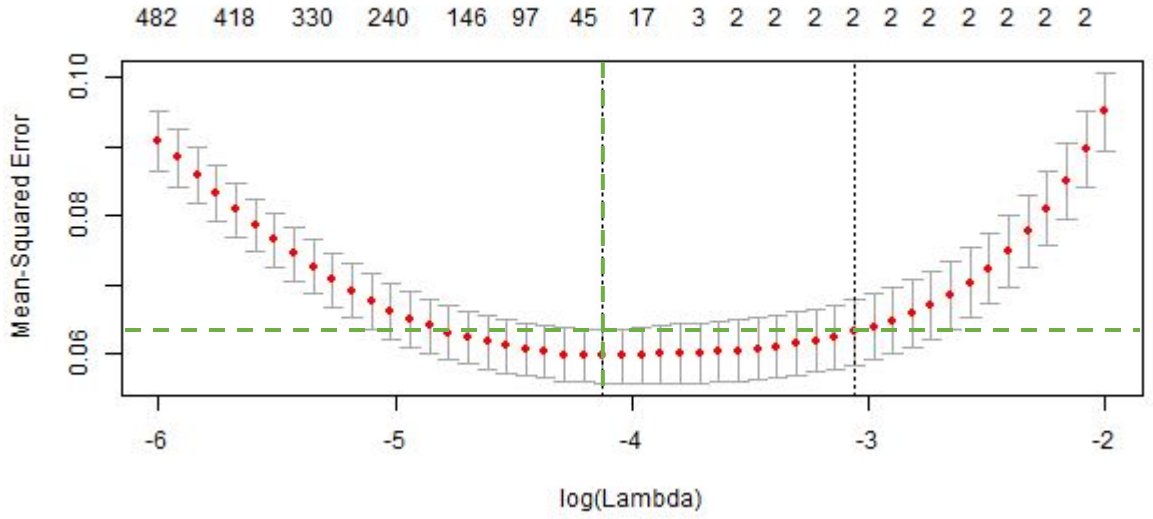


Hay coeficientes = 0

Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

Selección de modelos



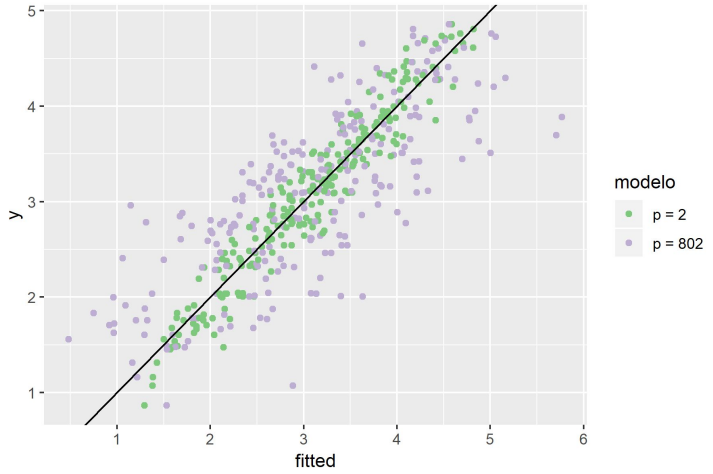


Con validation set
o Cross Validation

Podemos usar
one-standard-error rule (1se):
elegimos el modelo *más simple*
cuyo error esté dentro de un desvío
del mínimo

Ridge

$$y = 1 + 2x_1 + 2x_2 + \epsilon \qquad n = 960 \quad p = 802$$

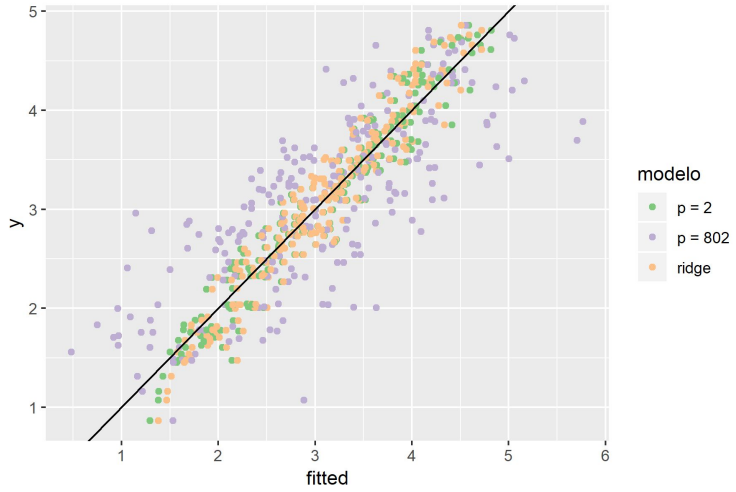


term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1 x2	28.9	14.8	1.95	0.0533
2 x1	-10.5	16.2	-0.650	0.517
3 (Intercept)	8.14	11.2	0.729	0.467
4 xc1_32	1.28	0.429	2.97	0.00340
5 xc2_291	-1.16	0.379	-3.05	0.00268
6 xc1_211	-1.12	0.401	-2.80	0.00576
7 xc2_157	-1.11	0.384	-2.90	0.00423
8 xc1_349	1.06	0.409	2.59	0.0104
9 xc2_109	1.01	0.407	2.48	0.0140
10 xc2_396	-1.01	0.427	-2.36	0.0196

... with 793 more rows

$$RMSE_{test}^{p=2} = 3.56$$

$$RMSE_{test}^{p=802} = 10.10$$



term	step	estimate	lambda	dev.ratio
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1 (Intercept)	1	-0.777	24.1	0.911
2 x1	1	0.00455	24.1	0.911
3 x2	1	0.00450	24.1	0.911
4 xc1_112	1	0.00337	24.1	0.911
5 xc1_110	1	0.00336	24.1	0.911
6 xc2_112	1	0.00335	24.1	0.911
7 xc1_125	1	0.00335	24.1	0.911
8 xc1_100	1	0.00335	24.1	0.911
9 xc2_111	1	0.00334	24.1	0.911
10 xc1_121	1	0.00334	24.1	0.911

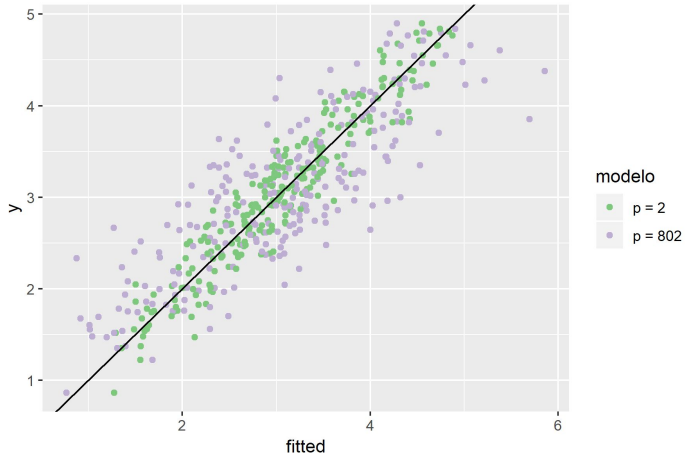
... with 793 more rows

$$\lambda_{1se} = 24.13$$

$$RMSE_{test}^{R} = 3.70$$

Lasso

$$y = 1 + 2x_1 + 2x_2 + \epsilon \qquad n = 960 \quad p = 802$$

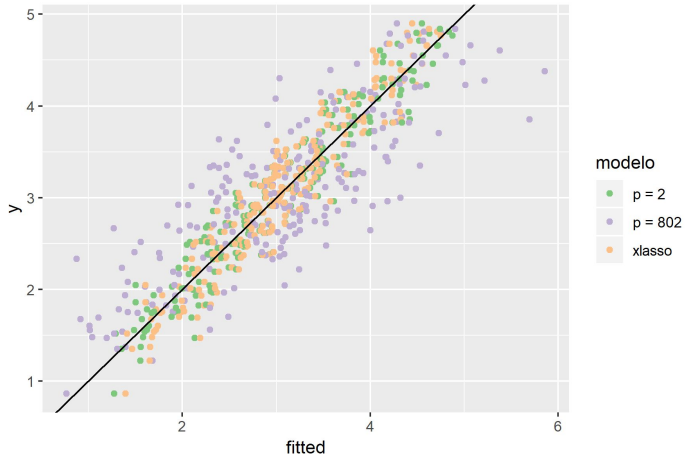


	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	x1	2.09	0.0677	30.9	1.20e-68
2	x2	2.04	0.0699	29.2	2.13e-65
3	(Intercept)	-0.897	0.967	-0.928	3.55e- 1
4	xnoc_101	0.206	0.0666	3.10	2.31e- 3
5	xnoc_388	0.199	0.0687	2.90	4.23e- 3
6	xnoc_561	0.181	0.0653	2.77	6.34e- 3
7	xnoc_218	-0.176	0.0684	-2.58	1.09e- 2
8	xnoc_13	-0.172	0.0665	-2.59	1.06e- 2
9	xnoc_787	0.166	0.0646	2.57	1.10e- 2
10	xnoc_64	0.165	0.0700	2.36	1.96e- 2

... with 793 more rows

$$RMSE_{test}^{p=2} = 3.56$$

$$RMSE_{test}^{p=802} = 10.10$$



	term	step	estimate	lambda	dev.ratio
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	x1	1	1.90	0.0398	0.916
2	x2	1	1.88	0.0398	0.916
3	(Intercept)	1	1.12	0.0398	0.916

$$\lambda_{1se} = 0.039$$

$$RMSE_{test}^{L} = 4.09$$

En resumen:

- El modelo lineal tiene buen rendimiento cuando:

Fuente: An Introduction to Statistical Learning (James *et al*, 2021)

En resumen:

- El modelo lineal tiene buen rendimiento cuando:
  → La verdadera relación es lineal (sesgo bajo)
  → La dimensionalidad (p/n) es baja (varianza baja)

- Podemos usar **regularización** (restringir la norma de los coeficientes) para **reducir la varianza** a costa de **aumentar el sesgo**. Es una alternativa elegante y computacionalmente eficiente a ***subset selection***.

- Usamos el **hiperparámetro lambda** para controlar el peso de la regularización

- La variante **ridge** incluye siempre todas las covariables, mientras que **lasso** hace **variable selection**. Ninguno domina al otro — en la práctica podemos elegir usando un validation set o cross validation.

- Ridge/Lasso funcionan aún cuando p/n>=1. Pero OJO: aún con regularización, incluir muchas **variables irrelevantes** deteriora el rendimiento (como vimos en el último ejemplo) y la interpretabilidad (e.g. el set de covariables que quedan en lasso es inestable)

- La regularización se usa en **muchos otros contextos** (deep learning, árboles de decisión, boosting, etc.)

Fuente: An Introduction to Statistical Learning (James *et al*, 2021)