Entrega de ejercicios Tema 3

Blanca Cano Camarero

14 de diciembre de 2022

Indice de contenidos

Ejercicio 2 **2**

¿De cuántos países consta la muestra utilizada? 3

¿Cuánto vale la suma de cuadrados que se utiliza para medir la variabilidad explicada por las tres variables regresoras? 4

¿Cuánto vale la varianza muestral de la variable respuesta ? 4

Contraste de la complejidad del modelo 5

Contrasta a nivel $\alpha = 0.05$ la hipótesis nula $H_0 : \beta_1 = 0$ 5

Contrasta a nivel $\alpha = 0.05$ la hipótesis nula $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ 6

Estima la correlación entre $\hat{\beta}_1$ y $\hat{\beta}_2$. 7

Calcula intervalos de confianza al nivel 90% para todos los β_i del modelo. 7

Predice el valor de la esperanza de vida de los hombres en un país para el que el índice de natalidad es 29, la mortalidad infantil vale 50 y el logaritmo de su pnb vale 7. Calcula un intervalo de confianza del 95 % para el valor esperado de dicha variable. 8

Ejercicio 9 **10**

Selecciona el modelo óptimo 19

Mejor modelo de acorde a R cuadrado ajustado 21

Cp 23

BIC 27

Usando ahora el método iterativo 30

Mejor modelo de acorde a R cuadrado ajustado con modelo iterativo hacia delante 32

Cp 34

BIC 38

Lasso 41

Ejercicio 12 **48**

Representación gráfica 49

Ajuste de regresión de mínimos cuadrados 50
Representación gráfica del error de predicción de validación cruzada generalizando en dunción de los grados de liberta utilizados. 50
Comentario de los resultados 54
Comportamiento general: Decrecemiento, mínimo y crecimiento más lento . . . 55
Gráficos desplazados 55
A σ menor admita más grados de libertad antes de volver a crecer el error. . . 55
Cremiento mayor del error de $\sigma = 1$ frente a $\sigma = 0.5$ 55

Ejercicio 2

```r
library(magrittr)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(purrr)
```

```
Attaching package: 'purrr'

The following object is masked from 'package:magrittr':

    set_names
```

```r
natalidad <- read.table("https://verso.mat.uam.es/~joser.berrendero/datos/natalidad.txt",
  mutate(log_pnb = log(pnb))

head(natalidad)
```

```
   nat mort mortinf esph espm  pnb  log_pnb
1 24.7  5.7    30.8 69.6 75.5  600 6.396930
2 12.5 11.9    14.4 68.3 74.7 2250 7.718685
3 13.4 11.7    11.3 71.8 77.7 2980 7.999679
4 11.6 13.4    14.8 65.4 73.8 2780 7.930206
5 14.3 10.2    16.0 67.2 75.7 1690 7.432484
6 13.6 10.7    26.9 66.5 72.4 1640 7.402452
```

Se desea estudiar la esperanza de vida de los hombres como función lineal de la tasa de natalidad, la tasa de mortalidad infantil y el logaritmo del producto nacional bruto. Para ello se ajusta un modelo de regresión lineal múltiple, con los resultados siguientes:

```r
reg <- lm(esph ~ nat  + mortinf + log_pnb, data = natalidad)
summary(reg)
```

```
Call:
lm(formula = esph ~ nat + mortinf + log_pnb, data = natalidad)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.4893 -2.1660  0.1581  2.0663  7.9084 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 65.95088    3.25642  20.253  < 2e-16 ***
nat         -0.14621    0.04762  -3.071  0.00285 ** 
mortinf     -0.13312    0.01537  -8.663  2.2e-13 ***
log_pnb      0.94478    0.32989   2.864  0.00524 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.128 on 87 degrees of freedom
Multiple R-squared:    0.9, Adjusted R-squared:  0.8966 
F-statistic: 261.1 on 3 and 87 DF,  p-value: < 2.2e-16
```

¿De cuántos países consta la muestra utilizada?

Cada fila se corresponde con un país distinto luego habrá:

```r
n <- nrow(natalidad)
n
```

```
[1] 91
```

¿Cuánto vale la suma de cuadrados que se utiliza para medir la variabilidad explicada por las tres variables regresoras?

Se busca calcular la suma de cuadrados explicada que se calcula como

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

Este dato no aparece directamente en el *sumary* del la regresión lineal, luego habrá que calcularlo de manera manual.

```r
media_natalidad <- mean(natalidad$nat)

p <- predict(reg, natalidad)
SCE <- Reduce('+',map(p, function(y_hat) (y_hat - media_natalidad)^2))
cat("La suma de cuadrados resultante es",SCE, '\n')
```

```
La suma de cuadrados resultante es 100388.8
```

¿Cuánto vale la varianza muestral de la variable respuesta ?

Esta se define como

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

Que por definición de la suma de cuadrados total equivale a conocer la suma de cuadrados total (SCT)

$$VMR = \frac{SCT}{n-1}$$

Sabiendo que el coeficiente de determinación viene dado por

$$r^2 = \frac{SCE}{SCT}$$

Este valor se puede calcular como gracias a la información provista por el resumen de la regresión lineal

$$r^2 = (\text{Multiple R-squared})^2 = 0.9^2 = 0.81.$$

```
r_squared <- (summary(reg)$adj.r.squared)^2
SCT <- SCE * r_squared
VMR = SCT/(n-1)
cat('VMR = ', VMR)
```

```
VMR =  896.6372
```

Contraste de la complejidad del modelo

Se desea que contrastar que $H_0 : \beta_1 = 0$
esto viene determinado por el sistema matricial A que es un vector fila con un uno en 1 y el resto 0.

Debe de satisfacer que $A\beta = 0$.

Definimos el modelo reducido $M0$ que resulta de imoner las restricciones de H_0.

Bajo $H_0 : A\beta = 0$ se verifica que

$$\frac{(SCR_0 - SCR)/k}{SCR/(n-p-1)} \equiv F_{k,n-p-1},$$

donde k es el número de restricciones (rango de A ciertamente), $p+1$ el número de β (variables del modelo a ajustar) y n el número de observaciones.

La región crítica del contraste para nivel α es

$$R = \left\{ \frac{(SCR_0 - SCR)/k}{SCR/(n-p-1)} > F_{k,n-p-1} \right\}$$

Utilizaremos el comando `anova` para R.

Contrasta a nivel $\alpha = 0.05$ la hipótesis nula $H_0 : \beta_1 = 0$

Solución

Al suponer que $\beta_1 = 0$ entonces ahora el modelo precindiría de natalidad, esto sería

```
reg0 <- lm(esph ~  mortinf + log_pnb, data = natalidad)
```

```r
anova(reg0, reg)
```

```
Analysis of Variance Table

Model 1: esph ~ mortinf + log_pnb
Model 2: esph ~ nat + mortinf + log_pnb
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     88 943.73
2     87 851.46  1    92.277 9.4286 0.00285 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

De aquí deducimos que la región crítica $F = 9.411$ y lo que buscábamos, que la probabilidad de pertenecer a la región crítica siendo la H_0 cierta es de

$$Pr(> F) = 0.00285$$

Como $ 0.00285 < 0.05$ rechazamos entonces la hipótesis nula.

Contrasta a nivel $\alpha = 0.05$ la hipótesis nula $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$

Contrasteremos que solo se pueda aproximar con la media

```r
reg0 <- lm(esph ~ 1, data = natalidad) # ahora sería solo con la media
anova(reg0, reg)
```

```
Analysis of Variance Table

Model 1: esph ~ 1
Model 2: esph ~ nat + mortinf + log_pnb
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     90 8516.5
2     87  851.5  3    7665.1 261.07 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como podemos observar, ahora

$$Pr(> F) < 2.2e - 16$$

es decir prácticamente 0 luego se rechaza la hipótesis nula.

Ya que si la media fuera un buen estimador se podría pensar que los modelos no tienen nada que ver con el modelo.

Estima la correlación entre $\hat{\beta}_1$ y $\hat{\beta}_2$.

Esto se hará con el comando `vcoc` que devuelve la matriz de covarianza de los parámetros ajustados.

$\hat{\beta}_1$ se corresponde a **`nat`** y $\hat{\beta}_2$ se corresponde con **`mortinf`** luego mirando la respectiva fila y columan tenemos que

```
vcov(reg)
```

```
            (Intercept)           nat       mortinf      log_pnb
(Intercept) 10.60424770 -0.0636441427 -0.0154732224 -1.033908908
nat         -0.06364414  0.0022673659 -0.0004865478  0.003160908
mortinf     -0.01547322 -0.0004865478  0.0002361416  0.002230265
log_pnb     -1.03390891  0.0031609079  0.0022302652  0.108830676
```

$$cov(\hat{\beta}_1, \hat{\beta}_2) = -0.0004865478$$

Puesto que el restultano es cercano a 0 podemos pensar que el signo de una es independiente del de la otra, es decir independientes.

Por lo tanto apriori no podríamos explicar una varia con la otra con un modelo lineal.

Calcula intervalos de confianza al nivel 90% para todos los β_i del modelo.

Usaremos la función `confint`

```
confint(reg, level=0.90)
```

```
                   5 %        95 %
(Intercept) 60.5369009 71.36485887
nat         -0.2253786 -0.06704702
mortinf     -0.1586652 -0.10756848
log_pnb      0.3963072  1.49324572
```

esto no da un rango de incertidumbre en el que puede oscilar cada uno de los respectivos parámetros con un 90% de probabilidad.

$$\beta_0 \in [60.537, 71.365],$$
$$\beta_{\text{nat}} \in [-0.225 - 0.067],$$
$$\beta_{\text{mortinf}} \in [-0.158, -0.108],$$
$$\beta_{\text{log pnb}} \in [0.396, 1.493].$$

Ya

$$P(\beta_{inf} \leq \beta \leq \beta_{sup}) = 1 - \alpha.$$

Donde β_{inf} y β_{sup} son los elementos inferiores y superiores del intervalo.

Cuando mayor sea el intervalo más incertidumbre tendremos de la corrección del parámetro.

Predice el valor de la esperanza de vida de los hombres en un país para el que el índice de natalidad es 29, la mortalidad infantil vale 50 y el logaritmo de su pnb vale 7. Calcula un intervalo de confianza del 95 % para el valor esperado de dicha variable.

Para ello usaremos la función `predict`

```
# Debemos de escribir el data frame con el nombre exacto
#bde las columnas con las que se creo el modelo de regresión.
# Estas son:
# nat     mortinf     log_pnb
new_data <- data.frame(
  nat = 29,
  mortinf = 50,
  log_pnb = 7
)
# Realizamos la predicción
predict(reg, new_data, interval = "confidence", level = 0.95 )
```

```
      fit      lwr    upr
1 61.6683 60.88761 62.449
```

Por lo tanto la esperanza de visa de los hombres de ese pais es de 61.6683 años. El intervalo de confianza es de $[60.88761, 62.449]$ es decir valores sobre los que podría oscilar el valor con un 95% de confianza.

Además cabe mencionar que la predicción del intervalo se basa en que el error residual están distribuidos bajo una distribución normal y varianza constante. Luego solo deberíamos de usar estos intervalos si tuvieramos razones para pensar que se da esta condición.

Ejercicio 9

Genera aleatoriamente una variable regresora X y un vector aleatorio ϵ de longitud $n = 100$, con distribución normal estándar e independientes.

Genera la variable respuesta de acuerdo con el modelo:

$$Y = X + X^2 + X^3 + \epsilon.$$

```
library(purrr)
library(ggplot2)
library(ramify)
```

```
Attaching package: 'ramify'


The following object is masked from 'package:purrr':

    flatten


The following object is masked from 'package:graphics':

    clip
```

```
n <- 100
X_distribution <- rnorm
error_distribution <- rnorm
modelo_1 <- function (x) {
  return (x + x^2 + x^3)
}

Y_distribution <- function (n, t){
  x <- X_distribution(n)
  e <- error_distribution(n)
  return (sapply(x, t) + e)
}


Y_1 <- Y_distribution(n, modelo_1)
```

```r
X <- X_distribution(n)
# Generamos las potencias
datos <- data.frame('1'=X)
for (i in 2:10) {
  datos[i] <- datos[1]*datos[i-1]
}
# Modelo1
Y <- unlist(as.list(datos[1]+datos[2] + datos[3] + error_distribution(n)))
datos$'y' <- Y
datos
```

```
            X1         X1.1          X1.2         X1.3          X1.4
1  -0.249288975 6.214499e-02 -1.549206e-02 3.862000e-03 -9.627541e-04
2   2.719948173 7.398118e+00  2.012250e+01 5.473215e+01  1.488686e+02
3  -0.540322260 2.919481e-01 -1.577461e-01 8.523372e-02 -4.605368e-02
4   0.597949839 3.575440e-01  2.137934e-01 1.278377e-01  7.644054e-02
5  -1.661442325 2.760391e+00 -4.586230e+00 7.619756e+00 -1.265979e+01
6   0.347353356 1.206544e-01  4.190969e-02 1.455747e-02  5.056587e-03
7  -0.070343173 4.948162e-03 -3.480694e-04 2.448431e-05 -1.722304e-06
8   1.509992028 2.280076e+00  3.442896e+00 5.198746e+00  7.850065e+00
9   0.754898239 5.698714e-01  4.301949e-01 3.247534e-01  2.451557e-01
10  0.300137106 9.008228e-02  2.703704e-02 8.114818e-03  2.435558e-03
11 -0.583268699 3.402024e-01 -1.984294e-01 1.157377e-01 -6.750615e-02
12  0.603773414 3.645423e-01  2.201010e-01 1.328911e-01  8.023612e-02
13 -0.750063001 5.625945e-01 -4.219813e-01 3.165126e-01 -2.374044e-01
14  0.721959946 5.212262e-01  3.763044e-01 2.716767e-01  1.961397e-01
15 -0.584298400 3.414046e-01 -1.994822e-01 1.165571e-01 -6.810414e-02
16  1.052788606 1.108364e+00  1.166873e+00 1.228470e+00  1.293320e+00
17 -0.998951390 9.979039e-01 -9.968575e-01 9.958122e-01 -9.947679e-01
18 -0.372286959 1.385976e-01 -5.159807e-02 1.920929e-02 -7.151368e-03
19  1.548450133 2.397698e+00  3.712716e+00 5.748955e+00  8.901970e+00
20 -0.082872218 6.867805e-03 -5.691502e-04 4.716674e-05 -3.908812e-06
21  1.739385975 3.025464e+00  5.262449e+00 9.153430e+00  1.592135e+01
22 -0.387785829 1.503778e-01 -5.831440e-02 2.261350e-02 -8.769194e-03
23 -0.017993368 3.237613e-04 -5.825556e-06 1.048214e-07 -1.886089e-09
24 -0.573890931 3.293508e-01 -1.890114e-01 1.084719e-01 -6.225107e-02
25  0.016618548 2.761762e-04  4.589647e-06 7.627327e-08  1.267551e-09
26 -0.752426854 5.661462e-01 -4.259836e-01 3.205215e-01 -2.411690e-01
27  0.368611069 1.358741e-01  5.008470e-02 1.846178e-02  6.805215e-03
28 -0.760698033 5.786615e-01 -4.401867e-01 3.348491e-01 -2.547191e-01
29  0.123307392 1.520471e-02  1.874853e-03 2.311833e-04  2.850661e-05
30 -0.678227975 4.599932e-01 -3.119802e-01 2.115937e-01 -1.435088e-01
```

31	0.138268530	1.911819e-02	2.643444e-03	3.655051e-04	5.053785e-05
32	-1.179964749	1.392317e+00	-1.642885e+00	1.938546e+00	-2.287416e+00
33	1.522911457	2.319259e+00	3.532027e+00	5.378964e+00	8.191685e+00
34	-0.714340100	5.102818e-01	-3.645147e-01	2.603875e-01	-1.860052e-01
35	0.669082222	4.476710e-01	2.995287e-01	2.004093e-01	1.340903e-01
36	-0.068519411	4.694910e-03	-3.216924e-04	2.204218e-05	-1.510317e-06
37	0.174132164	3.032201e-02	5.280037e-03	9.194243e-04	1.601013e-04
38	-0.277339699	7.691731e-02	-2.133222e-02	5.916272e-03	-1.640817e-03
39	0.418217836	1.749062e-01	7.314887e-02	3.059216e-02	1.279419e-02
40	0.040702146	1.656665e-03	6.742981e-05	2.744538e-06	1.117086e-07
41	0.892425344	7.964230e-01	7.107481e-01	6.342896e-01	5.660561e-01
42	-1.605262594	2.576868e+00	-4.136550e+00	6.640249e+00	-1.065934e+01
43	-0.114175176	1.303597e-02	-1.488384e-03	1.699365e-04	-1.940253e-05
44	0.611123276	3.734717e-01	2.282372e-01	1.394811e-01	8.524013e-02
45	-0.163374221	2.669114e-02	-4.360644e-03	7.124168e-04	-1.163905e-04
46	-1.626772213	2.646388e+00	-4.305070e+00	7.003369e+00	-1.139289e+01
47	2.943170092	8.662250e+00	2.549448e+01	7.503458e+01	2.208395e+02
48	1.351598717	1.826819e+00	2.469126e+00	3.337268e+00	4.510647e+00
49	-1.067244781	1.139011e+00	-1.215604e+00	1.297347e+00	-1.384587e+00
50	0.341564306	1.166662e-01	3.984900e-02	1.361100e-02	4.649031e-03
51	0.555141066	3.081816e-01	1.710843e-01	9.497590e-02	5.272502e-02
52	0.126927563	1.611061e-02	2.044880e-03	2.595516e-04	3.294426e-05
53	1.876951367	3.522946e+00	6.612399e+00	1.241115e+01	2.329513e+01
54	-0.632282016	3.997805e-01	-2.527741e-01	1.598245e-01	-1.010541e-01
55	0.685324745	4.696700e-01	3.218765e-01	2.205899e-01	1.511757e-01
56	0.775961856	6.021168e-01	4.672197e-01	3.625446e-01	2.813208e-01
57	0.076306702	5.822713e-03	4.443120e-04	3.390398e-05	2.587101e-06
58	-0.735561372	5.410505e-01	-3.979759e-01	2.927357e-01	-2.153251e-01
59	-1.035096938	1.071426e+00	-1.109029e+00	1.147953e+00	-1.188243e+00
60	0.674329885	4.547208e-01	3.066318e-01	2.067710e-01	1.394319e-01
61	-0.585580004	3.429039e-01	-2.007977e-01	1.175831e-01	-6.885432e-02
62	-0.638191257	4.072881e-01	-2.599277e-01	1.658836e-01	-1.058655e-01
63	-0.429974662	1.848782e-01	-7.949295e-02	3.417995e-02	-1.469651e-02
64	-2.217300827	4.916423e+00	-1.090119e+01	2.417121e+01	-5.359485e+01
65	1.148088074	1.318106e+00	1.513302e+00	1.737404e+00	1.994693e+00
66	-0.427459889	1.827220e-01	-7.810631e-02	3.338731e-02	-1.427174e-02
67	-0.854097480	7.294825e-01	-6.230492e-01	5.321447e-01	-4.545035e-01
68	-0.563713905	3.177734e-01	-1.791333e-01	1.009799e-01	-5.692378e-02
69	-0.048763047	2.377835e-03	-1.159505e-04	5.654098e-06	-2.757111e-07
70	0.490715344	2.408015e-01	1.181650e-01	5.798539e-02	2.845432e-02
71	0.086216320	7.433254e-03	6.408678e-04	5.525326e-05	4.763733e-06
72	-0.770394280	5.935073e-01	-4.572347e-01	3.522510e-01	-2.713721e-01
73	0.055846522	3.118834e-03	1.741760e-04	9.727126e-06	5.432261e-07

74	0.324076788	1.050258e-01	3.403641e-02	1.103041e-02	3.574700e-03
75	-0.062103046	3.856788e-03	-2.395183e-04	1.487482e-05	-9.237714e-07
76	-0.578689046	3.348810e-01	-1.937920e-01	1.121453e-01	-6.489725e-02
77	0.883064234	7.798024e-01	6.886156e-01	6.080918e-01	5.369842e-01
78	-0.948218965	8.991192e-01	-8.525619e-01	8.084153e-01	-7.665548e-01
79	-0.097289627	9.465272e-03	-9.208727e-04	8.959137e-05	-8.716311e-06
80	-0.497843205	2.478479e-01	-1.233894e-01	6.142856e-02	-3.058179e-02
81	-0.106839247	1.141462e-02	-1.219530e-03	1.302937e-04	-1.392048e-05
82	0.810029885	6.561484e-01	5.314998e-01	4.305307e-01	3.487428e-01
83	1.098246303	1.206145e+00	1.324644e+00	1.454786e+00	1.597713e+00
84	-0.203280626	4.132301e-02	-8.400168e-03	1.707591e-03	-3.471203e-04
85	0.288839102	8.342803e-02	2.409728e-02	6.960236e-03	2.010388e-03
86	1.294466746	1.675644e+00	2.169066e+00	2.807783e+00	3.634582e+00
87	0.972215436	9.452029e-01	9.189408e-01	8.934084e-01	8.685855e-01
88	0.146250069	2.138908e-02	3.128155e-03	4.574929e-04	6.690836e-05
89	-0.006086582	3.704648e-05	-2.254865e-07	1.372442e-09	-8.353481e-12
90	1.015898347	1.032049e+00	1.048457e+00	1.065126e+00	1.082060e+00
91	-1.563649505	2.445000e+00	-3.823123e+00	5.978024e+00	-9.347534e+00
92	1.004122856	1.008263e+00	1.012420e+00	1.016594e+00	1.020785e+00
93	-0.525554537	2.762076e-01	-1.451621e-01	7.629062e-02	-4.009488e-02
94	3.408246767	1.161615e+01	3.959069e+01	1.349348e+02	4.598913e+02
95	-0.009642257	9.297312e-05	-8.964707e-07	8.644000e-09	-8.334767e-11
96	0.699846857	4.897856e-01	3.427749e-01	2.398900e-01	1.678862e-01
97	-0.947872170	8.984617e-01	-8.516268e-01	8.072333e-01	-7.651540e-01
98	-0.713247249	5.087216e-01	-3.628443e-01	2.587977e-01	-1.845868e-01
99	-0.880647746	7.755405e-01	-6.829780e-01	6.014630e-01	-5.296770e-01
100	-0.791794172	6.269380e-01	-4.964059e-01	3.930513e-01	-3.112157e-01

	X1.5	X1.6	X1.7	X1.8	X1.9
1	2.400040e-04	-5.983034e-05	1.491505e-05	-3.718156e-06	9.268954e-07
2	4.049149e+02	1.101348e+03	2.995608e+03	8.147899e+03	2.216186e+04
3	2.488383e-02	-1.344529e-02	7.264787e-03	-3.925326e-03	2.120941e-03
4	4.570761e-02	2.733086e-02	1.634248e-02	9.771985e-03	5.843157e-03
5	2.103350e+01	-3.494595e+01	5.806069e+01	-9.646448e+01	1.602702e+02
6	1.756423e-03	6.100993e-04	2.119200e-04	7.361113e-05	2.556907e-05
7	1.211523e-07	-8.522239e-09	5.994813e-10	-4.216942e-11	2.966331e-12
8	1.185354e+01	1.789874e+01	2.702696e+01	4.081050e+01	6.162353e+01
9	1.850676e-01	1.397072e-01	1.054647e-01	7.961515e-02	6.010134e-02
10	7.310013e-04	2.194006e-04	6.585026e-05	1.976411e-05	5.931942e-06
11	3.937423e-02	-2.296575e-02	1.339520e-02	-7.813004e-03	4.557081e-03
12	4.844444e-02	2.924946e-02	1.766005e-02	1.066267e-02	6.437835e-03
13	1.780682e-01	-1.335624e-01	1.001802e-01	-7.514147e-02	5.636084e-02
14	1.416050e-01	1.022331e-01	7.380824e-02	5.328659e-02	3.847078e-02
15	3.979314e-02	-2.325107e-02	1.358556e-02	-7.938022e-03	4.638173e-03

16	1.361592e+00	1.433469e+00	1.509140e+00	1.588805e+00	1.672676e+00
17	9.937248e-01	-9.926828e-01	9.916418e-01	-9.906020e-01	9.895632e-01
18	2.662361e-03	-9.911623e-04	3.689968e-04	-1.373727e-04	5.114206e-05
19	1.378426e+01	2.134423e+01	3.305048e+01	5.117702e+01	7.924507e+01
20	3.239319e-07	-2.684496e-08	2.224701e-09	-1.843659e-10	1.527881e-11
21	2.769337e+01	4.816946e+01	8.378528e+01	1.457349e+02	2.534893e+02
22	3.400569e-03	-1.318693e-03	5.113703e-04	-1.983021e-04	7.689876e-05
23	3.393710e-11	-6.106427e-13	1.098752e-14	-1.977025e-16	3.557333e-18
24	3.572532e-02	-2.050244e-02	1.176616e-02	-6.752495e-03	3.875195e-03
25	2.106486e-11	3.500673e-13	5.817611e-15	9.668025e-17	1.606685e-18
26	1.814620e-01	-1.365369e-01	1.027340e-01	-7.729984e-02	5.816247e-02
27	2.508478e-03	9.246526e-04	3.408372e-04	1.256364e-04	4.631095e-05
28	1.937643e-01	-1.473961e-01	1.121239e-01	-8.529246e-02	6.488181e-02
29	3.515076e-06	4.334348e-07	5.344571e-08	6.590252e-09	8.126267e-10
30	9.733167e-02	-6.601306e-02	4.477191e-02	-3.036556e-02	2.059477e-02
31	6.987794e-06	9.661920e-07	1.335939e-07	1.847184e-08	2.554074e-09
32	2.699070e+00	-3.184808e+00	3.757961e+00	-4.434261e+00	5.232272e+00
33	1.247521e+01	1.899864e+01	2.893325e+01	4.406278e+01	6.710371e+01
34	1.328710e-01	-9.491508e-02	6.780165e-02	-4.843344e-02	3.459794e-02
35	8.971745e-02	6.002835e-02	4.016390e-02	2.687295e-02	1.798022e-02
36	1.034860e-07	-7.090802e-09	4.858576e-10	-3.329068e-11	2.281057e-12
37	2.787879e-05	4.854595e-06	8.453411e-07	1.472011e-07	2.563244e-08
38	4.550638e-04	-1.262072e-04	3.500228e-05	-9.707522e-06	2.692281e-06
39	5.350758e-03	2.237782e-03	9.358805e-04	3.914019e-04	1.636913e-04
40	4.546779e-09	1.850637e-10	7.532488e-12	3.065884e-13	1.247881e-14
41	5.051628e-01	4.508201e-01	4.023233e-01	3.590435e-01	3.204195e-01
42	1.711104e+01	-2.746772e+01	4.409290e+01	-7.078069e+01	1.136216e+02
43	2.215288e-06	-2.529309e-07	2.887843e-08	-3.297199e-09	3.764583e-10
44	5.209223e-02	3.183477e-02	1.945497e-02	1.188939e-02	7.265881e-03
45	1.901521e-05	-3.106596e-06	5.075376e-07	-8.291856e-08	1.354676e-08
46	1.853363e+01	-3.014999e+01	4.904717e+01	-7.978858e+01	1.297978e+02
47	6.499683e+02	1.912967e+03	5.630188e+03	1.657060e+04	4.877010e+04
48	6.096585e+00	8.240136e+00	1.113736e+01	1.505324e+01	2.034594e+01
49	1.477693e+00	-1.577060e+00	1.683109e+00	-1.796290e+00	1.917081e+00
50	1.587943e-03	5.423846e-04	1.852592e-04	6.327794e-05	2.161348e-05
51	2.926983e-02	1.624888e-02	9.020422e-03	5.007606e-03	2.779928e-03
52	4.181534e-06	5.307519e-07	6.736705e-08	8.550736e-09	1.085324e-09
53	4.372382e+01	8.206749e+01	1.540367e+02	2.891194e+02	5.426630e+02
54	6.389472e-02	-4.039948e-02	2.554387e-02	-1.615093e-02	1.021194e-02
55	1.036045e-01	7.100270e-02	4.865991e-02	3.334784e-02	2.285410e-02
56	2.182942e-01	1.693880e-01	1.314386e-01	1.019914e-01	7.914140e-02
57	1.974132e-07	1.506395e-08	1.149480e-09	8.771304e-11	6.693093e-12
58	1.583848e-01	-1.165017e-01	8.569418e-02	-6.303333e-02	4.636488e-02

59	1.229946e+00	-1.273114e+00	1.317796e+00	-1.364047e+00	1.411920e+00
60	9.402307e-02	6.340257e-02	4.275425e-02	2.883047e-02	1.944124e-02
61	4.031971e-02	-2.361042e-02	1.382579e-02	-8.096105e-03	4.740917e-03
62	6.756241e-02	-4.311774e-02	2.751736e-02	-1.756134e-02	1.120749e-02
63	6.319128e-03	-2.717065e-03	1.168269e-03	-5.023261e-04	2.159875e-04
64	1.188359e+02	-2.634950e+02	5.842476e+02	-1.295453e+03	2.872408e+03
65	2.290083e+00	2.629217e+00	3.018573e+00	3.465587e+00	3.978800e+00
66	6.100595e-03	-2.607760e-03	1.114713e-03	-4.764950e-04	2.036825e-04
67	3.881903e-01	-3.315523e-01	2.831780e-01	-2.418616e-01	2.065734e-01
68	3.208873e-02	-1.808886e-02	1.019694e-02	-5.748158e-03	3.240317e-03
69	1.344451e-08	-6.555953e-10	3.196883e-11	-1.558897e-12	7.601659e-14
70	1.396297e-02	6.851844e-03	3.362305e-03	1.649935e-03	8.096483e-04
71	4.107115e-07	3.541004e-08	3.052923e-09	2.632118e-10	2.269315e-11
72	2.090635e-01	-1.610614e-01	1.240807e-01	-9.559110e-02	7.364283e-02
73	3.033729e-08	1.694232e-09	9.461697e-11	5.284029e-12	2.950946e-13
74	1.158477e-03	3.754356e-04	1.216700e-04	3.943041e-05	1.277848e-05
75	5.736902e-08	-3.562791e-09	2.212602e-10	-1.374093e-11	8.533536e-13
76	3.755533e-02	-2.173286e-02	1.257657e-02	-7.277921e-03	4.211653e-03
77	4.741915e-01	4.187416e-01	3.697757e-01	3.265357e-01	2.883520e-01
78	7.268618e-01	-6.892241e-01	6.535354e-01	-6.196946e-01	5.876062e-01
79	8.480066e-07	-8.250225e-08	8.026613e-09	-7.809062e-10	7.597407e-11
80	1.522494e-02	-7.579631e-03	3.773468e-03	-1.878595e-03	9.352460e-04
81	1.487253e-06	-1.588970e-07	1.697644e-08	-1.813750e-09	1.937797e-10
82	2.824921e-01	2.288270e-01	1.853567e-01	1.501445e-01	1.216215e-01
83	1.754682e+00	1.927073e+00	2.116401e+00	2.324330e+00	2.552687e+00
84	7.056282e-05	-1.434405e-05	2.915868e-06	-5.927396e-07	1.204925e-07
85	5.806787e-04	1.677227e-04	4.844488e-05	1.399278e-05	4.041661e-06
86	4.704846e+00	6.090266e+00	7.883647e+00	1.020512e+01	1.321019e+01
87	8.444522e-01	8.209895e-01	7.981786e-01	7.760016e-01	7.544407e-01
88	9.785353e-06	1.431108e-06	2.092997e-07	3.061010e-08	4.476729e-09
89	5.084415e-14	-3.094671e-16	1.883597e-18	-1.146467e-20	6.978065e-23
90	1.099263e+00	1.116739e+00	1.134494e+00	1.152530e+00	1.170853e+00
91	1.461627e+01	-2.285472e+01	3.573677e+01	-5.587978e+01	8.737639e+01
92	1.024994e+00	1.029219e+00	1.033463e+00	1.037724e+00	1.042002e+00
93	2.107205e-02	-1.107451e-02	5.820259e-03	-3.058864e-03	1.607600e-03
94	1.567423e+03	5.342164e+03	1.820741e+04	6.205536e+04	2.115000e+05
95	8.036596e-13	-7.749093e-15	7.471874e-17	-7.204573e-19	6.946834e-21
96	1.174947e-01	8.222826e-02	5.754719e-02	4.027422e-02	2.818579e-02
97	7.252682e-01	-6.874615e-01	6.516257e-01	-6.176578e-01	5.854607e-01
98	1.316560e-01	-9.390327e-02	6.697625e-02	-4.777063e-02	3.407227e-02
99	4.664589e-01	-4.107860e-01	3.617577e-01	-3.185811e-01	2.805578e-01
100	2.464188e-01	-1.951130e-01	1.544893e-01	-1.223237e-01	9.685521e-02

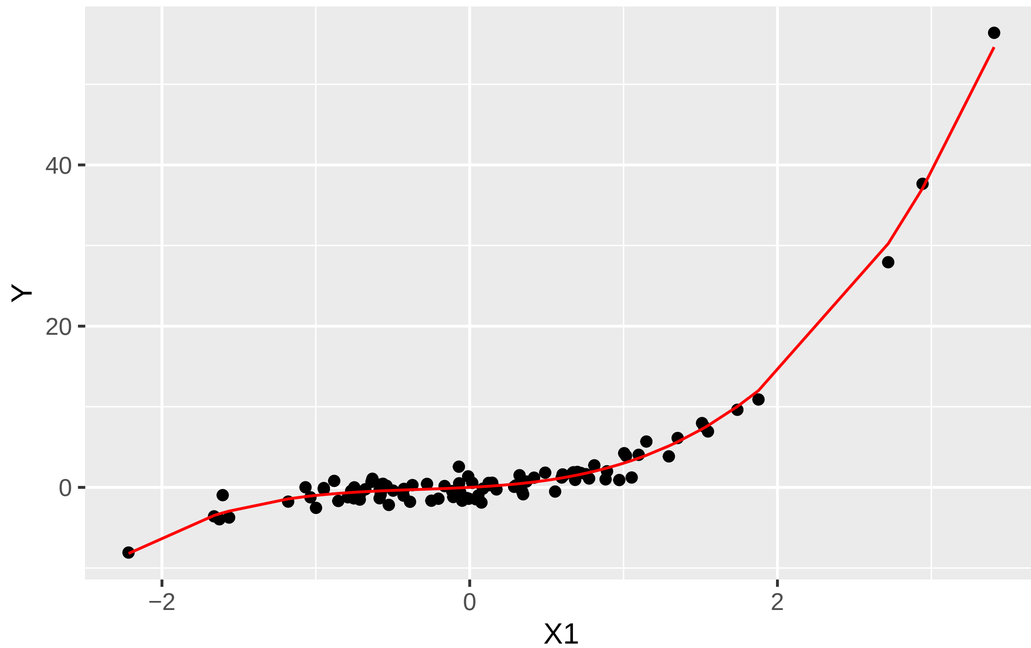
y

1	-1.652793e+00
2	2.793025e+01
3	1.924381e-01
4	1.222683e+00
5	-3.594376e+00
6	-8.578417e-01
7	2.568515e+00
8	7.975119e+00
9	1.634300e+00
10	2.412135e-01
11	3.521703e-01
12	1.603093e+00
13	7.577461e-05
14	1.786755e+00
15	-2.189142e-01
16	1.224788e+00
17	-2.539526e+00
18	2.836117e-01
19	6.937991e+00
20	-7.924957e-01
21	9.625496e+00
22	-1.785452e+00
23	-1.329062e+00
24	-3.947830e-01
25	5.630801e-01
26	-1.364711e+00
27	7.222642e-01
28	-1.063443e+00
29	5.520529e-01
30	-2.326748e-01
31	3.778512e-01
32	-1.766943e+00
33	7.554110e+00
34	-1.516353e+00
35	1.793988e+00
36	5.052421e-01
37	-2.542546e-01
38	4.315936e-01
39	1.203060e+00
40	-1.476333e+00
41	1.994649e+00
42	-9.660343e-01
43	-4.632757e-01

```
44   1.491484e+00
45   1.678068e-01
46  -3.959156e+00
47   3.765797e+01
48   6.123222e+00
49   1.188568e-02
50  -5.240437e-01
51  -5.170579e-01
52   2.701110e-01
53   1.089878e+01
54   1.066949e+00
55   9.278190e-01
56   1.099548e+00
57  -1.882503e+00
58  -1.043253e+00
59  -1.232200e+00
60   1.874850e+00
61  -1.353800e+00
62   6.904728e-01
63  -1.021069e+00
64  -8.084900e+00
65   5.687279e+00
66  -1.855100e-01
67  -1.685620e+00
68   4.415870e-01
69  -1.651114e+00
70   1.822685e+00
71  -1.585063e-01
72  -4.329000e-01
73  -1.010974e+00
74   1.504283e+00
75  -5.948865e-01
76  -8.777610e-01
77   9.956773e-01
78  -8.568993e-02
79  -5.223629e-01
80  -4.019738e-01
81  -1.194017e+00
82   2.734624e+00
83   4.037946e+00
84  -1.409521e+00
85   7.213761e-02
86   3.845031e+00
```

```
87   9.110551e-01
88   5.759779e-01
89  -1.408581e+00
90   3.918231e+00
91  -3.743037e+00
92   4.239808e+00
93  -2.178254e+00
94   5.638923e+01
95   1.364673e+00
96   1.915726e+00
97  -2.180457e-01
98  -8.494400e-01
99   8.020847e-01
100 -1.226764e+00
```

```r
datos2 <- datos
#plot(X, Y, type='p')

datos2$'Y_ideal' <- sapply(X, function(x)(x*(1 + x*(1 + x))))

p <- ggplot(datos2) +
  geom_point( aes(X1, Y), colour = 'black' ) +
  geom_line(aes(X1, Y_ideal), colour = 'red' )


p
```



Selecciona el modelo óptimo

entre todos los submodelos que contienen como variabels regresoras $X, X^2, X^3, \dots, X^{10}$. ¿Cuál es el mejor modelo de acuerdo con los criterios C_p, BIC y R_a^2?

Solución Para poder realizar estas comparaciones utilizaremos la función `leaps::regsubsets`.

Donde los argumentos que nos interesan son: - `datos` - `method` donde indicaremos de si se trata de `exhaustive` o `forward`.

Finalmente consultaremos los valores respectivos consultando los atributos de la salida:

- `adjr2` para Adjusted r-squared.
- `cp` para Mallows's CP.
- `bic` para Schwartz's information criterion BIC.

```
modelo_exhaustivo <- leaps::regsubsets(
  y ~ .,
  data=datos
  )
resumen_1 <- summary(modelo_exhaustivo)
resumen_1
```

```
Subset selection object
Call: regsubsets.formula(y ~ ., data = datos)
10 Variables  (and intercept)
     Forced in Forced out
X1       FALSE      FALSE
X1.1     FALSE      FALSE
X1.2     FALSE      FALSE
X1.3     FALSE      FALSE
X1.4     FALSE      FALSE
X1.5     FALSE      FALSE
X1.6     FALSE      FALSE
X1.7     FALSE      FALSE
X1.8     FALSE      FALSE
X1.9     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
1  ( 1 ) " " " "  "*"  " "  " "  " "  " "  " "  " "  " " 
2  ( 1 ) " " "*"  "*"  " "  " "  " "  " "  " "  " "  " " 
3  ( 1 ) "*" "*"  "*"  " "  " "  " "  " "  " "  " "  " " 
4  ( 1 ) "*" "*"  "*"  " "  " "  " "  "*"  " "  " "  " " 
5  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  " "  " "  " "  " " 
6  ( 1 ) "*" "*"  "*"  " "  " "  "*"  " "  "*"  " "  "*" 
7  ( 1 ) "*" "*"  "*"  " "  "*"  "*"  " "  "*"  " "  "*" 
8  ( 1 ) "*" "*"  " "  "*"  "*"  " "  "*"  "*"  "*"  "*" 
```

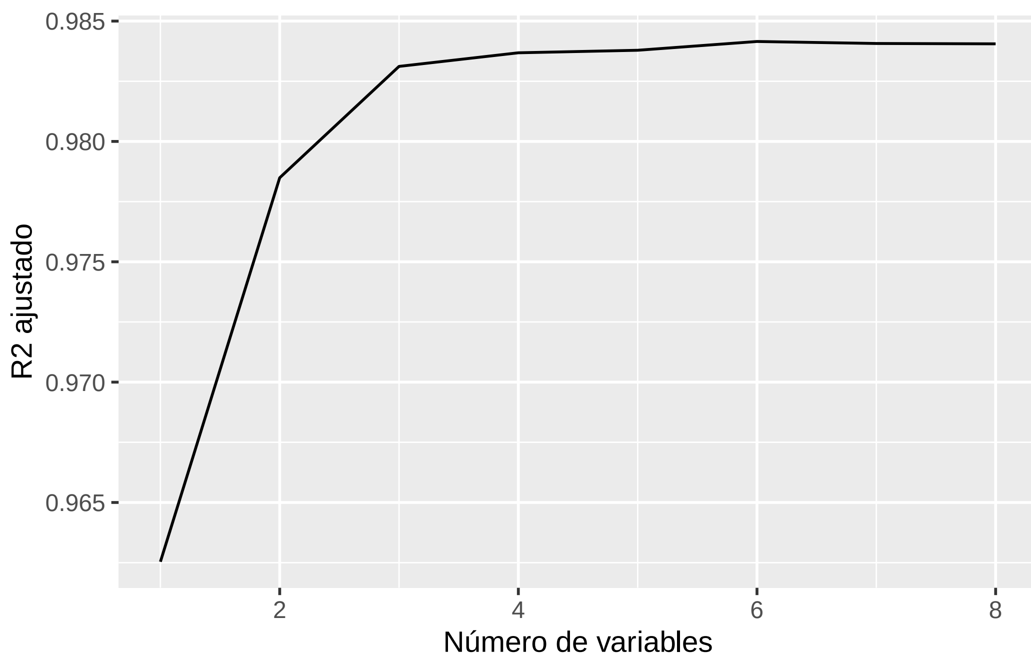
```r
library("ggplot2")

pinta_ajuste <- function(y_criterio, label){
  plot_data <- data.frame(
    x = 1:8,
    y = y_criterio
  )
  print(plot_data)
  y_label <- label

  ggp <- ggplot(plot_data, aes(x, y)) +    # ggplot2 plot with default grid
    geom_line()
  ggp +
    xlab("Número de variables") +
    ylab(y_label) #+
```

```
    #scale_y_continuous(minor_breaks = seq(0, 10, 0.005))
}
```

Mejor modelo de acorde a R cuadrado ajustado

```
pinta_ajuste(resumen_1$adjr2, "R2 ajustado")
```

```
  x         y
1 1 0.9625363
2 2 0.9784907
3 3 0.9831177
4 4 0.9836825
5 5 0.9837876
6 6 0.9841519
7 7 0.9840697
8 8 0.9840552
```




```
cat("Se alcanza el máximo con ", which.max(resumen_1$adjr2), "parametros")
```

```
Se alcanza el máximo con  6 parametros
```

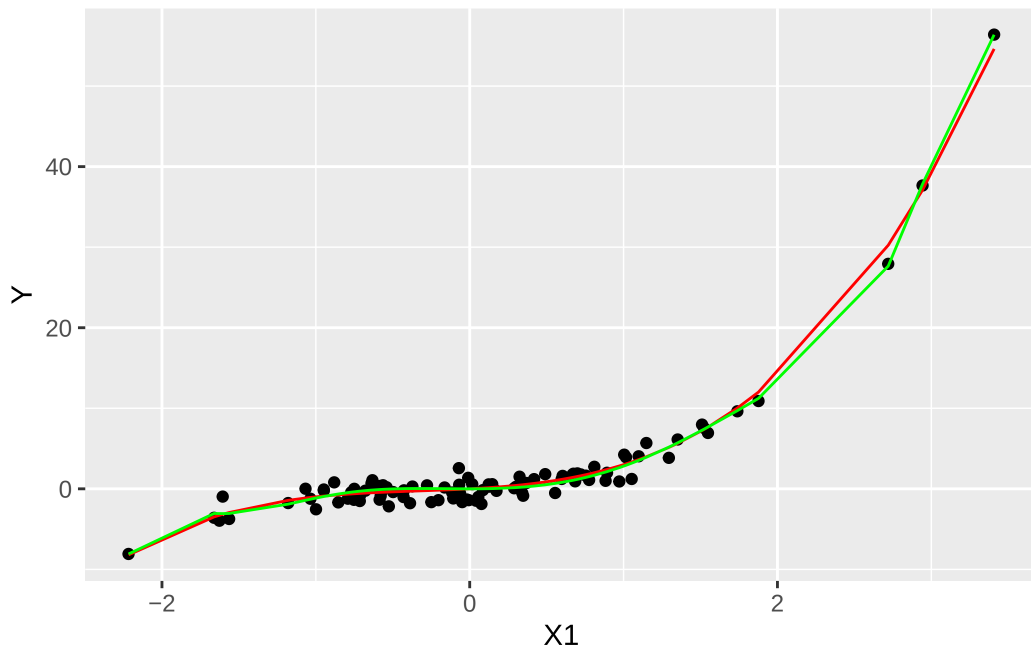
Será mejor donde alcance un máximo, esto es en $p = 8$ es decir utilizando 8 parámetros. Vamos a proceder a analizar los coeficientes.

```r
# Mejores elementos
#  X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
#7  ( 1 ) " " "*"  "*"  " "  "*"  " "  "*"  "*"  "*"  "*"

modelo_r2_a <- lm(y ~ X1.1 + X1.2+ X1.3  + X1.5 + X1.6 + X1.7 + X1.8+ X1.9 -1, data=datos)
summary(modelo_r2_a)
```

```
Call:
lm(formula = y ~ X1.1 + X1.2 + X1.3 + X1.5 + X1.6 + X1.7 + X1.8 +
    X1.9 - 1, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-2.14582 -0.79568 -0.04307  0.51860  2.56372

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
X1.1  1.122129   0.582944   1.925 0.057327 .
X1.2  2.149738   0.262967   8.175 1.56e-12 ***
X1.3 -0.475617   0.843862  -0.564 0.574385
X1.5  0.181357   0.342402   0.530 0.597622
X1.6 -0.197817   0.074504  -2.655 0.009343 **
X1.7  0.010886   0.036871   0.295 0.768471
X1.8  0.031248   0.013073   2.390 0.018874 *
X1.9 -0.006347   0.001729  -3.671 0.000405 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.013 on 92 degrees of freedom
Multiple R-squared:  0.9846,	Adjusted R-squared:  0.9832
F-statistic:   734 on 8 and 92 DF,  p-value: < 2.2e-16
```

```r
y_predit_r2_a <- predict(object=modelo_r2_a, newdata=datos)

datos2$'Y_r2a' <- y_predit_r2_a
```

```r
p <- ggplot(datos2) +
  geom_point( aes(X1, Y), colour = 'black' ) +
  geom_line(aes(X1, Y_ideal), colour = 'red' ) +
  geom_line(aes(X1, Y_r2a), colour = 'green' )


p
```



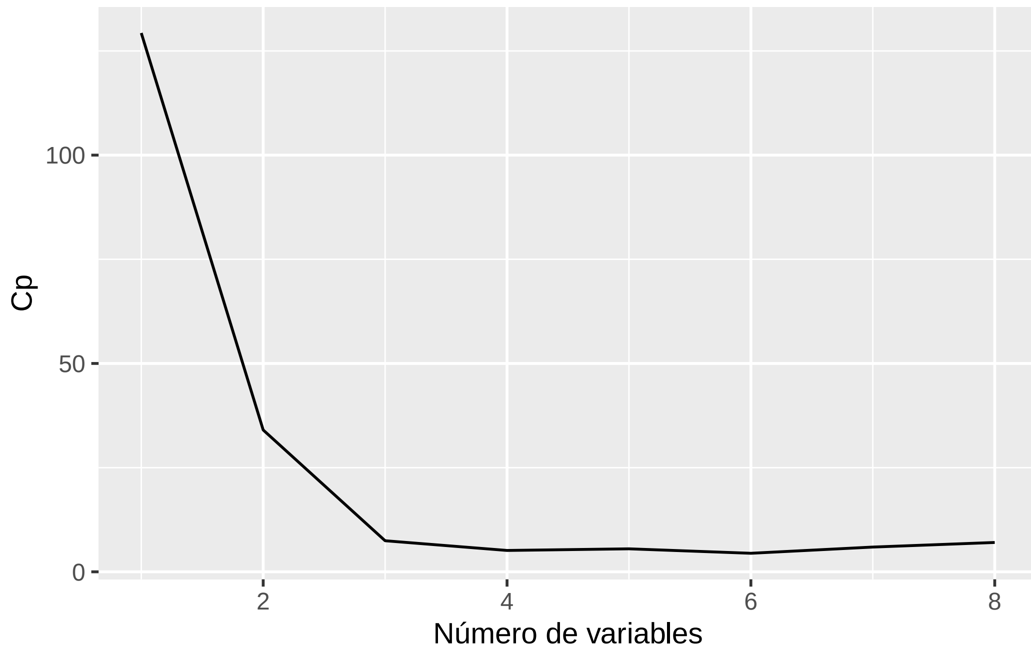
Como vemos más variables lo que hacen es acercarse al ruido.

Cp

```r
pinta_ajuste(resumen_1$cp, "Cp")
```

```
  x          y
1 1 129.305102
2 2  34.035725
3 3   7.457557
4 4   5.128409
5 5   5.521242
```

```
6 6   4.447157
7 7   5.938336
8 8   7.041769
```




```r
cat("Con el criterio CP, el mejor modelo tiene ", which.min(resumen_1$cp), "parametros\n")
```

```
Con el criterio CP, el mejor modelo tiene  6 parametros
```

```r
summary(modelo_exhaustivo)
```

```
Subset selection object
Call: regsubsets.formula(y ~ ., data = datos)
10 Variables  (and intercept)
     Forced in Forced out
X1       FALSE      FALSE
X1.1     FALSE      FALSE
X1.2     FALSE      FALSE
X1.3     FALSE      FALSE
X1.4     FALSE      FALSE
X1.5     FALSE      FALSE
```

```
X1.6     FALSE      FALSE
X1.7     FALSE      FALSE
X1.8     FALSE      FALSE
X1.9     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
1  ( 1 ) " " " "  "*"  " "  " "  " "  " "  " "  " "  " "
2  ( 1 ) " " "*"  "*"  " "  " "  " "  " "  " "  " "  " "
3  ( 1 ) "*" "*"  "*"  " "  " "  " "  " "  " "  " "  " "
4  ( 1 ) "*" "*"  "*"  " "  " "  " "  "*"  " "  " "  " "
5  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  " "  " "  " "  " "
6  ( 1 ) "*" "*"  "*"  " "  " "  "*"  " "  "*"  " "  "*"
7  ( 1 ) "*" "*"  "*"  " "  "*"  "*"  " "  "*"  " "  "*"
8  ( 1 ) "*" "*"  " "  "*"  "*"  " "  "*"  "*"  "*"  "*"
```

Vemos que con este criterio alcanza un mínimo en cinco variables. es decir que el mejor de los resultados sería

```
Selection Algorithm: exhaustive
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
6  ( 1 ) " " " "  "*"  "*"  "*"  "*"  " "  "*"  " "  "*"
```

que si entrenamos obtendríamos

```
modelo_cp<- lm(y ~ X1.2 + X1.3 + X1.4 + X1.5 + X1.7  + X1.9 -1, data=datos)
summary(modelo_cp)
```

```
Call:
lm(formula = y ~ X1.2 + X1.3 + X1.4 + X1.5 + X1.7 + X1.9 - 1,
    data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-2.00901 -0.76552 -0.00048  0.61253  2.56912

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
X1.2  1.835034   0.150362  12.204  < 2e-16 ***
X1.3  1.335177   0.204149   6.540 3.18e-09 ***
X1.4 -0.157263   0.037344  -4.211 5.82e-05 ***
```

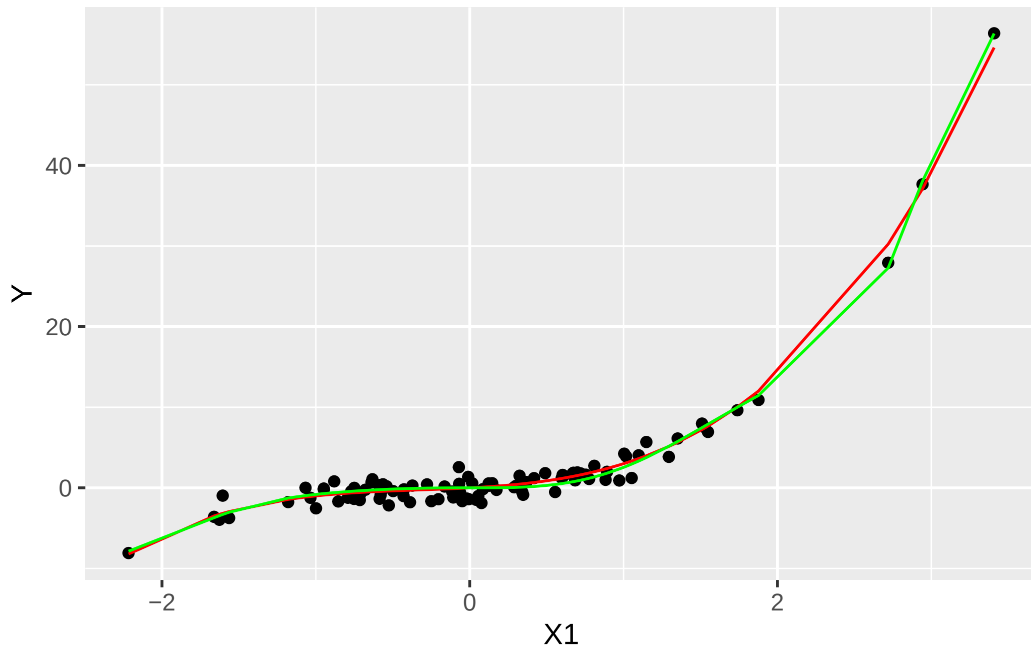
```
X1.5 -0.544717   0.108081  -5.040 2.25e-06 ***
X1.7  0.077957   0.017017   4.581 1.42e-05 ***
X1.9 -0.003261   0.000767  -4.252 5.00e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.021 on 94 degrees of freedom
Multiple R-squared:  0.984, Adjusted R-squared:  0.983
F-statistic: 962.3 on 6 and 94 DF,  p-value: < 2.2e-16
```

```r
X
```

```
  [1] -0.249288975  2.719948173 -0.540322260  0.597949839 -1.661442325
  [6]  0.347353356 -0.070343173  1.509992028  0.754898239  0.300137106
 [11] -0.583268699  0.603773414 -0.750063001  0.721959946 -0.584298400
 [16]  1.052788606 -0.998951390 -0.372286959  1.548450133 -0.082872218
 [21]  1.739385975 -0.387785829 -0.017993368 -0.573890931  0.016618548
 [26] -0.752426854  0.368611069 -0.760698033  0.123307392 -0.678227975
 [31]  0.138268530 -1.179964749  1.522911457 -0.714340100  0.669082222
 [36] -0.068519411  0.174132164 -0.277339699  0.418217836  0.040702146
 [41]  0.892425344 -1.605262594 -0.114175176  0.611123276 -0.163374221
 [46] -1.626772213  2.943170092  1.351598717 -1.067244781  0.341564306
 [51]  0.555141066  0.126927563  1.876951367 -0.632282016  0.685324745
 [56]  0.775961856  0.076306702 -0.735561372 -1.035096938  0.674329885
 [61] -0.585580004 -0.638191257 -0.429974662 -2.217300827  1.148088074
 [66] -0.427459889 -0.854097480 -0.563713905 -0.048763047  0.490715344
 [71]  0.086216320 -0.770394280  0.055846522  0.324076788 -0.062103046
 [76] -0.578689046  0.883064234 -0.948218965 -0.097289627 -0.497843205
 [81] -0.106839247  0.810029885  1.098246303 -0.203280626  0.288839102
 [86]  1.294466746  0.972215436  0.146250069 -0.006086582  1.015898347
 [91] -1.563649505  1.004122856 -0.525554537  3.408246767 -0.009642257
 [96]  0.699846857 -0.947872170 -0.713247249 -0.880647746 -0.791794172
```

```r
y_predit_cp <- predict(object=modelo_cp, newdata=datos)

datos2$'Y_cp' <- y_predit_cp

p <- ggplot(datos2) +
  geom_point( aes(X1, Y), colour = 'black' ) +
  geom_line(aes(X1, Y_ideal), colour = 'red' ) +
  geom_line(aes(X1, Y_cp), colour = 'green' )
```

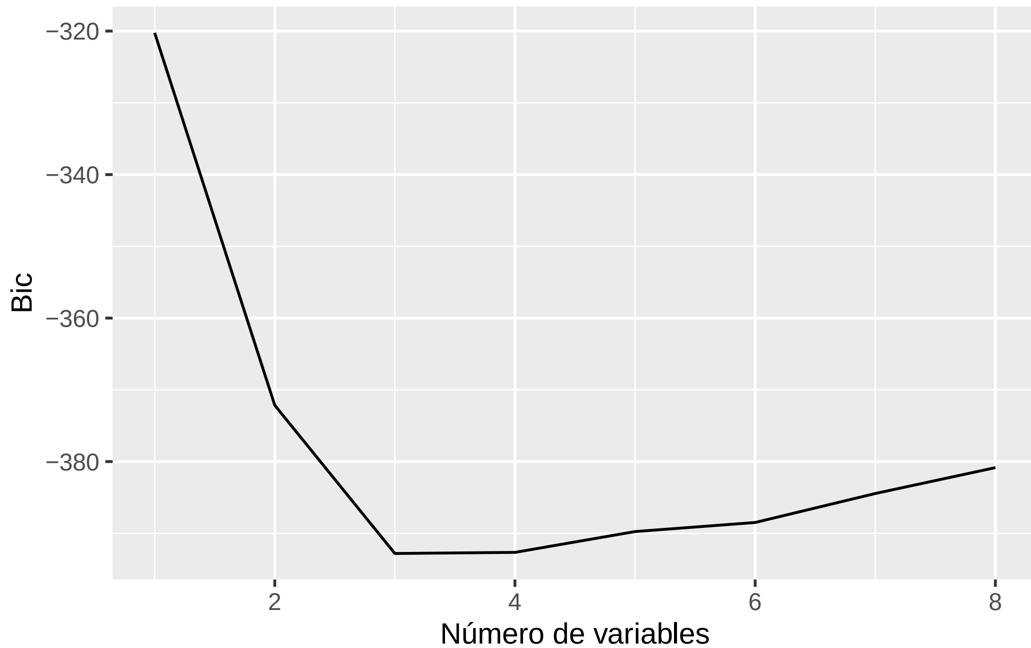
```r
p
```



BIC

```r
pinta_ajuste(resumen_1$bic, "Bic")
```

```
  x         y
1 1 -320.2431
2 2 -372.1525
3 3 -392.8052
4 4 -392.6503
5 5 -389.7491
6 6 -388.4862
7 7 -384.4452
8 8 -380.8419
```




```
cat("Con el criterio BIC, el mejor modelo tiene ", which.min(resumen_1$bic), "parametros\n
```

```
Con el criterio BIC, el mejor modelo tiene  3 parametros
```

```
summary(modelo_exhaustivo)
```

```
Subset selection object
Call: regsubsets.formula(y ~ ., data = datos)
10 Variables  (and intercept)
     Forced in Forced out
X1       FALSE      FALSE
X1.1     FALSE      FALSE
X1.2     FALSE      FALSE
X1.3     FALSE      FALSE
X1.4     FALSE      FALSE
X1.5     FALSE      FALSE
X1.6     FALSE      FALSE
X1.7     FALSE      FALSE
X1.8     FALSE      FALSE
X1.9     FALSE      FALSE
1 subsets of each size up to 8
```

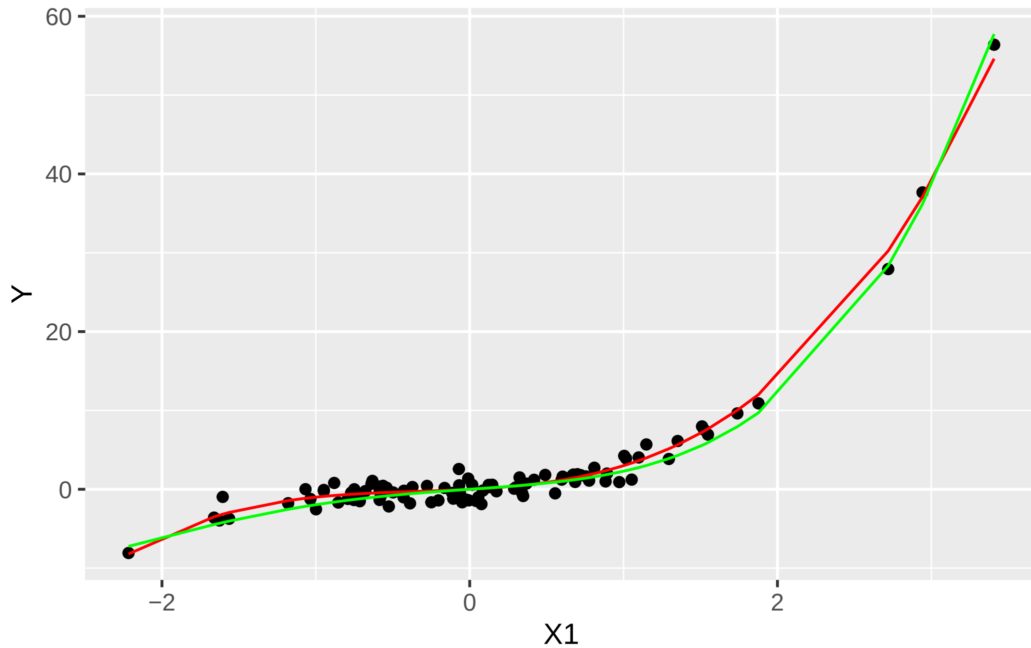
```
Selection Algorithm: exhaustive
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
1  ( 1 ) " " " "  "*"  " "  " "  " "  " "  " "  " "  " "
2  ( 1 ) " " "*"  "*"  " "  " "  " "  " "  " "  " "  " "
3  ( 1 ) "*" "*"  "*"  " "  " "  " "  " "  " "  " "  " "
4  ( 1 ) "*" "*"  "*"  " "  " "  " "  "*"  " "  " "  " "
5  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  " "  " "  " "  " "
6  ( 1 ) "*" "*"  "*"  " "  " "  "*"  " "  "*"  " "  "*"
7  ( 1 ) "*" "*"  "*"  " "  "*"  "*"  " "  "*"  " "  "*"
8  ( 1 ) "*" "*"  " "  "*"  "*"  " "  "*"  "*"  "*"  "*"
```

```
modelo_bic<- lm(y ~ X1 + X1.2 + X1.3 -1, data=datos)
summary(modelo_bic)
```

```
Call:
lm(formula = y ~ X1 + X1.2 + X1.3 - 1, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.9869 -0.3958  0.3586  1.0346  3.2532 

Coefficients:
     Estimate Std. Error t value Pr(>|t|)    
X1    1.36412    0.21405   6.373 6.25e-09 ***
X1.2  0.76293    0.07521  10.143  < 2e-16 ***
X1.3  0.16963    0.01915   8.857 3.92e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.172 on 97 degrees of freedom
Multiple R-squared:  0.9782,	Adjusted R-squared:  0.9776 
F-statistic:  1453 on 3 and 97 DF,  p-value: < 2.2e-16
```

```
y_predit_bic <- predict(object=modelo_bic, newdata=datos)

datos2$'Y_bic' <- y_predit_bic

p <- ggplot(datos2) +
  geom_point( aes(X1, Y), colour = 'black' ) +
  geom_line(aes(X1, Y_ideal), colour = 'red' ) +
```

```
  geom_line(aes(X1, Y_bic), colour = 'green' )

p
```



Usando ahora el método iterativo

```
modelo_iterativo_delante <- leaps::regsubsets(
  y ~ .,
  data=datos,
  method = "forward"
  )
resumen_2 <- summary(modelo_iterativo_delante)
resumen_2
```

```
Subset selection object
Call: regsubsets.formula(y ~ ., data = datos, method = "forward")
10 Variables  (and intercept)
     Forced in Forced out
X1       FALSE      FALSE
X1.1     FALSE      FALSE
X1.2     FALSE      FALSE
```

```
X1.3      FALSE      FALSE
X1.4      FALSE      FALSE
X1.5      FALSE      FALSE
X1.6      FALSE      FALSE
X1.7      FALSE      FALSE
X1.8      FALSE      FALSE
X1.9      FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: forward
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
1  ( 1 ) " " " "  "*"  " "  " "  " "  " "  " "  " "  " "
2  ( 1 ) " " "*"  "*"  " "  " "  " "  " "  " "  " "  " "
3  ( 1 ) "*" "*"  "*"  " "  " "  " "  " "  " "  " "  " "
4  ( 1 ) "*" "*"  "*"  " "  " "  " "  "*"  " "  " "  " "
5  ( 1 ) "*" "*"  "*"  "*"  " "  " "  "*"  " "  " "  " "
6  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  " "  " "  " "
7  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  " "  " "  "*"
8  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  "*"  " "  "*"
```

Observaciones:

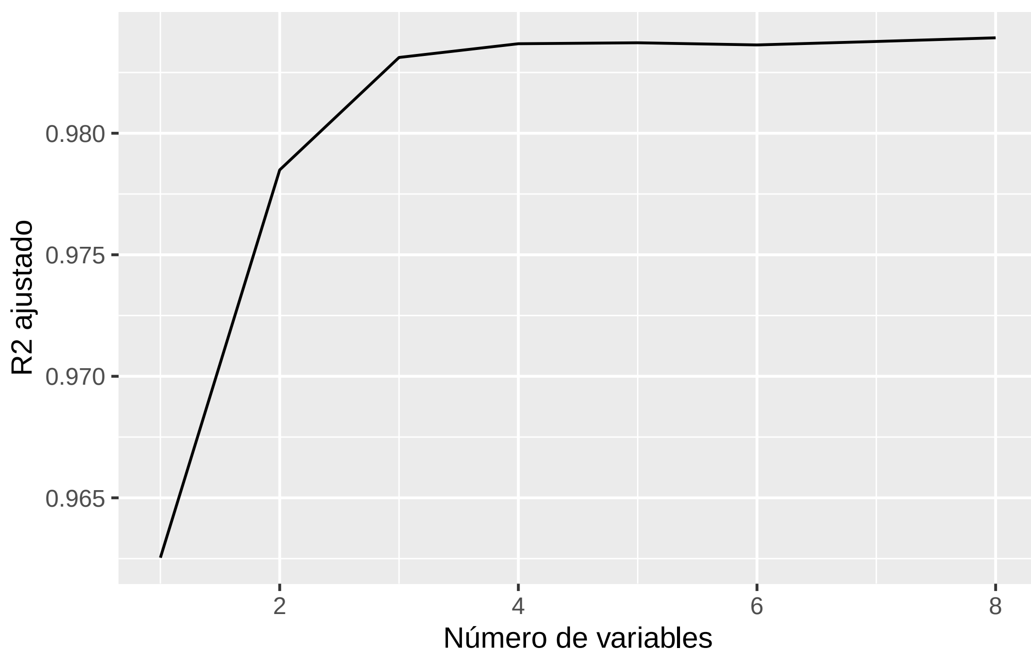
Como vemos ahora las variables seleccionadas para p parámetros están incluídas en el modelo con $p+1$ parámetros. Esto es lo esperado ya que es el comportamiento de forward con el fin de mejorar el rendimiendo.

```r
library("ggplot2")

pinta_ajuste <- function(y_criterio, label){
  plot_data <- data.frame(
    x = 1:8,
    y = y_criterio
  )
  print(plot_data)
  y_label <- label

  ggp <- ggplot(plot_data, aes(x, y)) +    # ggplot2 plot with default grid
    geom_line()
  ggp +
    xlab("Número de variables") +
    ylab(y_label) #+
    #scale_y_continuous(minor_breaks = seq(0, 10, 0.005))
}
```

Mejor modelo de acorde a R cuadrado ajustado con modelo iterativo hacia delante

```
pinta_ajuste(resumen_2$adjr2, "R2 ajustado")
```

```
  x         y
1 1 0.9625363
2 2 0.9784907
3 3 0.9831177
4 4 0.9836825
5 5 0.9837226
6 6 0.9836337
7 7 0.9837758
8 8 0.9839255
```




```
cat("Se alcanza el máximo con ", which.max(resumen_2$adjr2), "parametros")
```

```
Se alcanza el máximo con  8 parametros
```

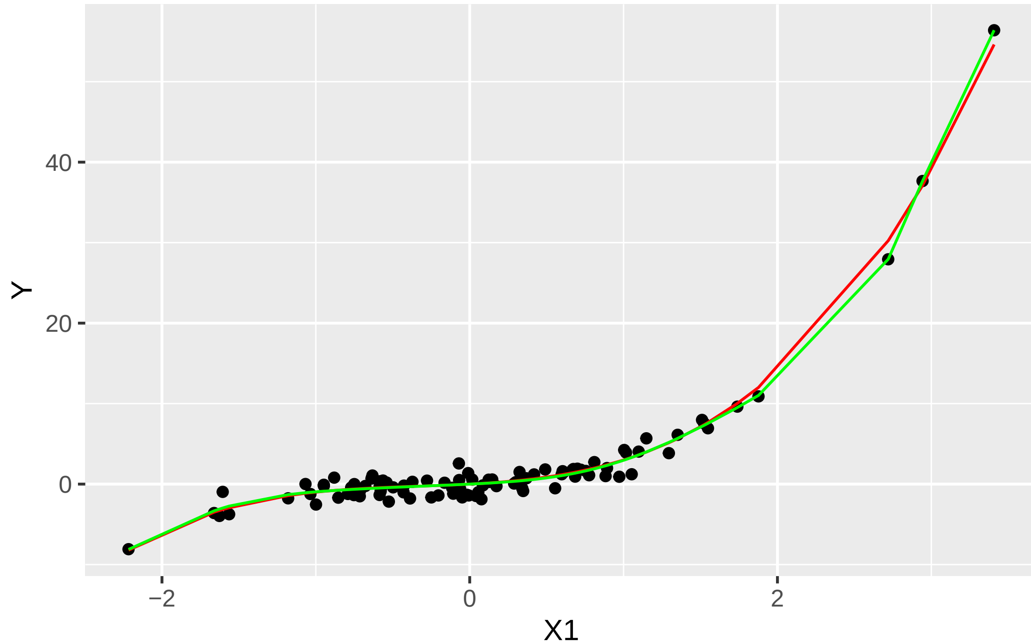
Será mejor donde alcance un máximo, esto es en $p = 8$ es decir utilizando 8 parámetros. Vamos a proceder a analizar los coeficientes.

```
#         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
#8  ( 1 ) "*" "*"  "*"  "*"  "*"  "*"  " "  "*"  " "  "*"


modelo_r2_a <- lm(y ~ X1 + X1.1 + X1.2 + X1.3 + X1.4+ X1.5 + X1.7 + X1.9 -1, data=datos)
summary(modelo_r2_a)
```

```
Call:
lm(formula = y ~ X1 + X1.1 + X1.2 + X1.3 + X1.4 + X1.5 + X1.7 +
    X1.9 - 1, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-2.08494 -0.71870  0.00298  0.54979  2.63032

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
X1    0.9134511  0.2969166   3.076 0.002759 **
X1.1  0.5671895  0.4550587   1.246 0.215778
X1.2  1.0853690  0.2829289   3.836 0.000229 ***
X1.3  0.7007772  0.4937337   1.419 0.159179
X1.4 -0.0385814  0.0525049  -0.735 0.464321
X1.5 -0.3107986  0.1744742  -1.781 0.078156 .
X1.7  0.0452944  0.0234354   1.933 0.056345 .
X1.9 -0.0019416  0.0009853  -1.971 0.051776 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9795 on 92 degrees of freedom
Multiple R-squared:  0.9856,    Adjusted R-squared:  0.9843
F-statistic: 785.7 on 8 and 92 DF,  p-value: < 2.2e-16
```

```
y_predit_r2_a <- predict(object=modelo_r2_a, newdata=datos)

datos2$'Y_r2a' <- y_predit_r2_a

p <- ggplot(datos2) +
  geom_point( aes(X1, Y), colour = 'black' ) +
  geom_line(aes(X1, Y_ideal), colour = 'red' ) +
  geom_line(aes(X1, Y_r2a), colour = 'green' )
```

```r
p
```

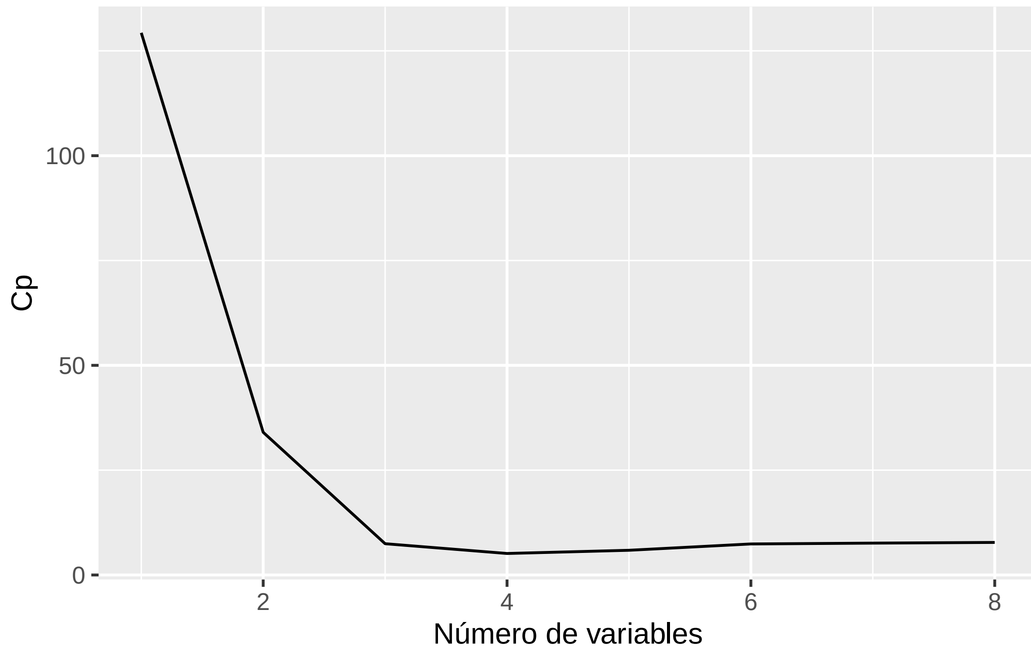


Como vemos más variables lo que hacen es acercarse al ruido.

Cp

```r
pinta_ajuste(resumen_2$cp, "Cp")
```

```
  x          y
1 1 129.305102
2 2  34.035725
3 3   7.457557
4 4   5.128409
5 5   5.895687
6 6   7.404182
7 7   7.597657
8 8   7.765890
```




```
cat("Con el criterio CP, el mejor modelo tiene ", which.min(resumen_2$cp), "parametros\n")
```

```
Con el criterio CP, el mejor modelo tiene  4 parametros
```

```
summary(modelo_iterativo_delante)
```

```
Subset selection object
Call: regsubsets.formula(y ~ ., data = datos, method = "forward")
10 Variables  (and intercept)
     Forced in Forced out
X1       FALSE      FALSE
X1.1     FALSE      FALSE
X1.2     FALSE      FALSE
X1.3     FALSE      FALSE
X1.4     FALSE      FALSE
X1.5     FALSE      FALSE
X1.6     FALSE      FALSE
X1.7     FALSE      FALSE
X1.8     FALSE      FALSE
X1.9     FALSE      FALSE
1 subsets of each size up to 8
```

```
Selection Algorithm: forward
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
1  ( 1 ) " " " "  "*"  " "  " "  " "  " "  " "  " "  " "
2  ( 1 ) " " "*"  "*"  " "  " "  " "  " "  " "  " "  " "
3  ( 1 ) "*" "*"  "*"  " "  " "  " "  " "  " "  " "  " "
4  ( 1 ) "*" "*"  "*"  " "  " "  " "  "*"  " "  " "  " "
5  ( 1 ) "*" "*"  "*"  "*"  " "  " "  "*"  " "  " "  " "
6  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  " "  " "  " "
7  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  " "  " "  "*"
8  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  "*"  " "  "*"
```

Vemos que con este criterio alcanza un mínimo en cinco variables. es decir que el mejor de los resultados sería

```
Selection Algorithm: exhaustive
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
6  ( 1 ) " " " "  "*"  "*"  "*"  "*"  " "  "*"  " "  "*"
```

que si entrenamos obtendríamos

```
modelo_cp<- lm(y ~ X1.2 + X1.3 + X1.4 + X1.5 + X1.7  + X1.9 -1, data=datos)
summary(modelo_cp)
```

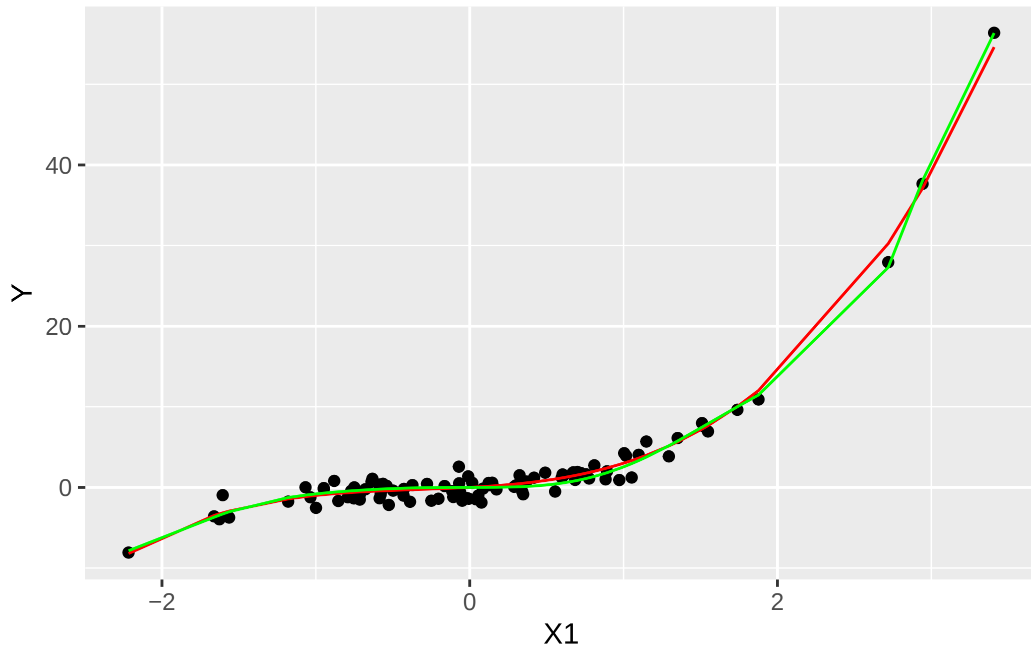
```
Call:
lm(formula = y ~ X1.2 + X1.3 + X1.4 + X1.5 + X1.7 + X1.9 - 1,
    data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-2.00901 -0.76552 -0.00048  0.61253  2.56912

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
X1.2  1.835034   0.150362  12.204  < 2e-16 ***
X1.3  1.335177   0.204149   6.540 3.18e-09 ***
X1.4 -0.157263   0.037344  -4.211 5.82e-05 ***
X1.5 -0.544717   0.108081  -5.040 2.25e-06 ***
X1.7  0.077957   0.017017   4.581 1.42e-05 ***
X1.9 -0.003261   0.000767  -4.252 5.00e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.021 on 94 degrees of freedom
Multiple R-squared:  0.984, Adjusted R-squared:  0.983 
F-statistic: 962.3 on 6 and 94 DF,  p-value: < 2.2e-16
```

```
X
```

```
  [1] -0.249288975  2.719948173 -0.540322260  0.597949839 -1.661442325
  [6]  0.347353356 -0.070343173  1.509992028  0.754898239  0.300137106
 [11] -0.583268699  0.603773414 -0.750063001  0.721959946 -0.584298400
 [16]  1.052788606 -0.998951390 -0.372286959  1.548450133 -0.082872218
 [21]  1.739385975 -0.387785829 -0.017993368 -0.573890931  0.016618548
 [26] -0.752426854  0.368611069 -0.760698033  0.123307392 -0.678227975
 [31]  0.138268530 -1.179964749  1.522911457 -0.714340100  0.669082222
 [36] -0.068519411  0.174132164 -0.277339699  0.418217836  0.040702146
 [41]  0.892425344 -1.605262594 -0.114175176  0.611123276 -0.163374221
 [46] -1.626772213  2.943170092  1.351598717 -1.067244781  0.341564306
 [51]  0.555141066  0.126927563  1.876951367 -0.632282016  0.685324745
 [56]  0.775961856  0.076306702 -0.735561372 -1.035096938  0.674329885
 [61] -0.585580004 -0.638191257 -0.429974662 -2.217300827  1.148088074
 [66] -0.427459889 -0.854097480 -0.563713905 -0.048763047  0.490715344
 [71]  0.086216320 -0.770394280  0.055846522  0.324076788 -0.062103046
 [76] -0.578689046  0.883064234 -0.948218965 -0.097289627 -0.497843205
 [81] -0.106839247  0.810029885  1.098246303 -0.203280626  0.288839102
 [86]  1.294466746  0.972215436  0.146250069 -0.006086582  1.015898347
 [91] -1.563649505  1.004122856 -0.525554537  3.408246767 -0.009642257
 [96]  0.699846857 -0.947872170 -0.713247249 -0.880647746 -0.791794172
```

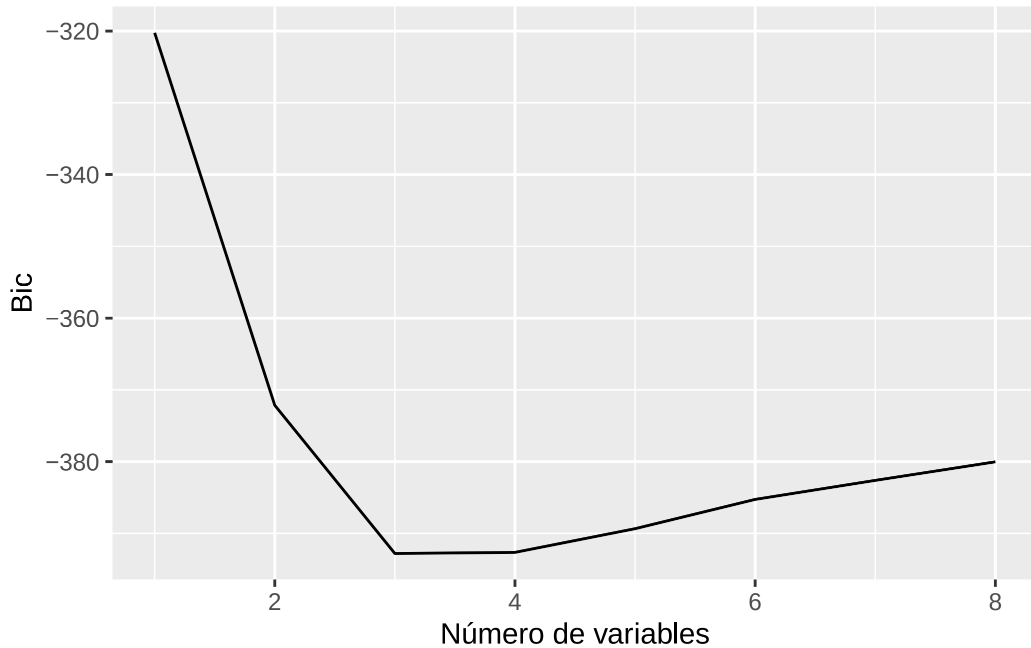
```
y_predit_cp <- predict(object=modelo_cp, newdata=datos)

datos2$'Y_cp' <- y_predit_cp

p <- ggplot(datos2) +
  geom_point( aes(X1, Y), colour = 'black' ) +
  geom_line(aes(X1, Y_ideal), colour = 'red' ) +
  geom_line(aes(X1, Y_cp), colour = 'green' )

p
```



BIC

```
pinta_ajuste(resumen_2$bic, "Bic")
```

```
  x         y
1 1 -320.2431
2 2 -372.1525
3 3 -392.8052
4 4 -392.6503
5 5 -389.3495
6 6 -385.2692
7 7 -382.6170
8 8 -380.0319
```




```
cat("Con el criterio BIC, el mejor modelo tiene ", which.min(resumen_1$bic), "parametros\n
```

```
Con el criterio BIC, el mejor modelo tiene  3 parametros
```

```
summary(modelo_iterativo_delante)
```

```
Subset selection object
Call: regsubsets.formula(y ~ ., data = datos, method = "forward")
10 Variables  (and intercept)
     Forced in Forced out
X1       FALSE      FALSE
X1.1     FALSE      FALSE
X1.2     FALSE      FALSE
X1.3     FALSE      FALSE
X1.4     FALSE      FALSE
X1.5     FALSE      FALSE
X1.6     FALSE      FALSE
X1.7     FALSE      FALSE
X1.8     FALSE      FALSE
X1.9     FALSE      FALSE
1 subsets of each size up to 8
```

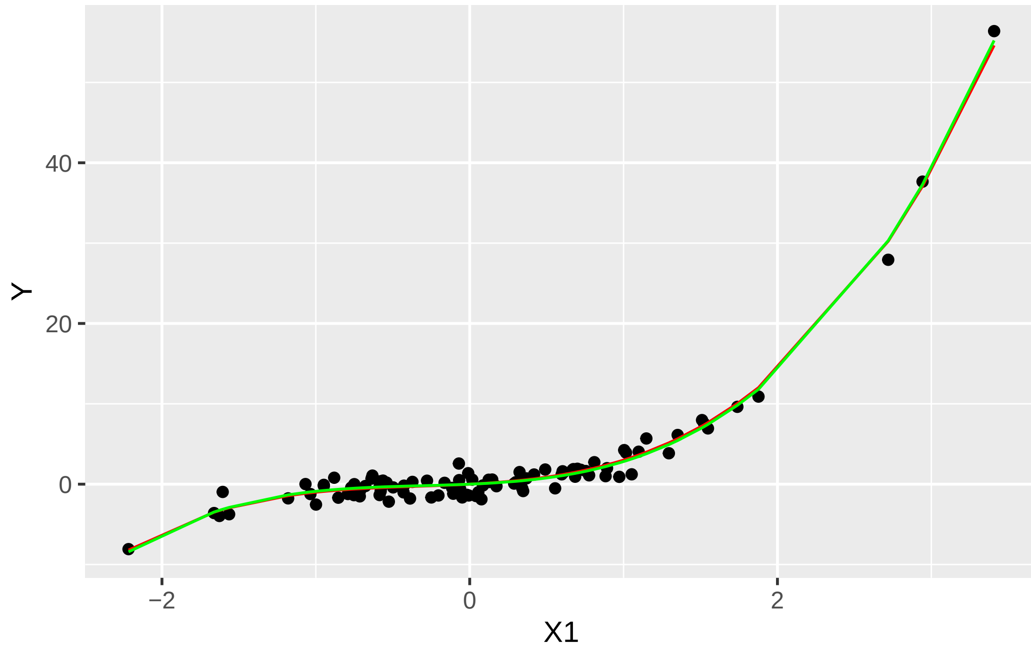
```
Selection Algorithm: forward
         X1  X1.1 X1.2 X1.3 X1.4 X1.5 X1.6 X1.7 X1.8 X1.9
1  ( 1 ) " " " "  "*"  " "  " "  " "  " "  " "  " "  " "
2  ( 1 ) " " "*"  "*"  " "  " "  " "  " "  " "  " "  " "
3  ( 1 ) "*" "*"  "*"  " "  " "  " "  " "  " "  " "  " "
4  ( 1 ) "*" "*"  "*"  " "  " "  " "  "*"  " "  " "  " "
5  ( 1 ) "*" "*"  "*"  "*"  " "  " "  "*"  " "  " "  " "
6  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  " "  " "  " "
7  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  " "  " "  "*"
8  ( 1 ) "*" "*"  "*"  "*"  " "  "*"  "*"  "*"  " "  "*"
```

```r
modelo_bic<- lm(y ~ X1 + X1.1 +  X1.2  -1, data=datos)
summary(modelo_bic)
```

```
Call:
lm(formula = y ~ X1 + X1.1 + X1.2 - 1, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.37921 -0.85000 -0.07882  0.55116  2.62123 

Coefficients:
     Estimate Std. Error t value Pr(>|t|)    
X1    0.81206    0.15847   5.124 1.52e-06 ***
X1.1  0.96445    0.08048  11.984  < 2e-16 ***
X1.2  1.04190    0.03835  27.167  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 97 degrees of freedom
Multiple R-squared:  0.9841,    Adjusted R-squared:  0.9836 
F-statistic:  2005 on 3 and 97 DF,  p-value: < 2.2e-16
```

```r
y_predit_bic <- predict(object=modelo_bic, newdata=datos)

datos2$'Y_bic' <- y_predit_bic

p <- ggplot(datos2) +
  geom_point( aes(X1, Y), colour = 'black' ) +
  geom_line(aes(X1, Y_ideal), colour = 'red' ) +
```

```r
  geom_line(aes(X1, Y_bic), colour = 'green' )

p
```




```r
print('Los coefficientes del modelo son')
```

```
[1] "Los coefficientes del modelo son"
```

```r
coef(modelo_bic)
```

```
       X1      X1.1      X1.2 
0.8120638 0.9644500 1.0418972 
```

Lasso

```r
library(glmnet)
```

```
Loading required package: Matrix
```

```
Attaching package: 'Matrix'

The following objects are masked from 'package:ramify':

    tril, triu

Loaded glmnet 4.1-6
```

```r
x <- datos[c(1:10)]
x
```

```
            X1       X1.1          X1.2        X1.3          X1.4
1  -0.249288975 6.214499e-02 -1.549206e-02 3.862000e-03 -9.627541e-04
2   2.719948173 7.398118e+00  2.012250e+01 5.473215e+01  1.488686e+02
3  -0.540322260 2.919481e-01 -1.577461e-01 8.523372e-02 -4.605368e-02
4   0.597949839 3.575440e-01  2.137934e-01 1.278377e-01  7.644054e-02
5  -1.661442325 2.760391e+00 -4.586230e+00 7.619756e+00 -1.265979e+01
6   0.347353356 1.206544e-01  4.190969e-02 1.455747e-02  5.056587e-03
7  -0.070343173 4.948162e-03 -3.480694e-04 2.448431e-05 -1.722304e-06
8   1.509992028 2.280076e+00  3.442896e+00 5.198746e+00  7.850065e+00
9   0.754898239 5.698714e-01  4.301949e-01 3.247534e-01  2.451557e-01
10  0.300137106 9.008228e-02  2.703704e-02 8.114818e-03  2.435558e-03
11 -0.583268699 3.402024e-01 -1.984294e-01 1.157377e-01 -6.750615e-02
12  0.603773414 3.645423e-01  2.201010e-01 1.328911e-01  8.023612e-02
13 -0.750063001 5.625945e-01 -4.219813e-01 3.165126e-01 -2.374044e-01
14  0.721959946 5.212262e-01  3.763044e-01 2.716767e-01  1.961397e-01
15 -0.584298400 3.414046e-01 -1.994822e-01 1.165571e-01 -6.810414e-02
16  1.052788606 1.108364e+00  1.166873e+00 1.228470e+00  1.293320e+00
17 -0.998951390 9.979039e-01 -9.968575e-01 9.958122e-01 -9.947679e-01
18 -0.372286959 1.385976e-01 -5.159807e-02 1.920929e-02 -7.151368e-03
19  1.548450133 2.397698e+00  3.712716e+00 5.748955e+00  8.901970e+00
20 -0.082872218 6.867805e-03 -5.691502e-04 4.716674e-05 -3.908812e-06
21  1.739385975 3.025464e+00  5.262449e+00 9.153430e+00  1.592135e+01
22 -0.387785829 1.503778e-01 -5.831440e-02 2.261350e-02 -8.769194e-03
23 -0.017993368 3.237613e-04 -5.825556e-06 1.048214e-07 -1.886089e-09
24 -0.573890931 3.293508e-01 -1.890114e-01 1.084719e-01 -6.225107e-02
25  0.016618548 2.761762e-04  4.589647e-06 7.627327e-08  1.267551e-09
26 -0.752426854 5.661462e-01 -4.259836e-01 3.205215e-01 -2.411690e-01
27  0.368611069 1.358741e-01  5.008470e-02 1.846178e-02  6.805215e-03
28 -0.760698033 5.786615e-01 -4.401867e-01 3.348491e-01 -2.547191e-01
29  0.123307392 1.520471e-02  1.874853e-03 2.311833e-04  2.850661e-05
```

30	-0.678227975	4.599932e-01	-3.119802e-01	2.115937e-01	-1.435088e-01
31	0.138268530	1.911819e-02	2.643444e-03	3.655051e-04	5.053785e-05
32	-1.179964749	1.392317e+00	-1.642885e+00	1.938546e+00	-2.287416e+00
33	1.522911457	2.319259e+00	3.532027e+00	5.378964e+00	8.191685e+00
34	-0.714340100	5.102818e-01	-3.645147e-01	2.603875e-01	-1.860052e-01
35	0.669082222	4.476710e-01	2.995287e-01	2.004093e-01	1.340903e-01
36	-0.068519411	4.694910e-03	-3.216924e-04	2.204218e-05	-1.510317e-06
37	0.174132164	3.032201e-02	5.280037e-03	9.194243e-04	1.601013e-04
38	-0.277339699	7.691731e-02	-2.133222e-02	5.916272e-03	-1.640817e-03
39	0.418217836	1.749062e-01	7.314887e-02	3.059216e-02	1.279419e-02
40	0.040702146	1.656665e-03	6.742981e-05	2.744538e-06	1.117086e-07
41	0.892425344	7.964230e-01	7.107481e-01	6.342896e-01	5.660561e-01
42	-1.605262594	2.576868e+00	-4.136550e+00	6.640249e+00	-1.065934e+01
43	-0.114175176	1.303597e-02	-1.488384e-03	1.699365e-04	-1.940253e-05
44	0.611123276	3.734717e-01	2.282372e-01	1.394811e-01	8.524013e-02
45	-0.163374221	2.669114e-02	-4.360644e-03	7.124168e-04	-1.163905e-04
46	-1.626772213	2.646388e+00	-4.305070e+00	7.003369e+00	-1.139289e+01
47	2.943170092	8.662250e+00	2.549448e+01	7.503458e+01	2.208395e+02
48	1.351598717	1.826819e+00	2.469126e+00	3.337268e+00	4.510647e+00
49	-1.067244781	1.139011e+00	-1.215604e+00	1.297347e+00	-1.384587e+00
50	0.341564306	1.166662e-01	3.984900e-02	1.361100e-02	4.649031e-03
51	0.555141066	3.081816e-01	1.710843e-01	9.497590e-02	5.272502e-02
52	0.126927563	1.611061e-02	2.044880e-03	2.595516e-04	3.294426e-05
53	1.876951367	3.522946e+00	6.612399e+00	1.241115e+01	2.329513e+01
54	-0.632282016	3.997805e-01	-2.527741e-01	1.598245e-01	-1.010541e-01
55	0.685324745	4.696700e-01	3.218765e-01	2.205899e-01	1.511757e-01
56	0.775961856	6.021168e-01	4.672197e-01	3.625446e-01	2.813208e-01
57	0.076306702	5.822713e-03	4.443120e-04	3.390398e-05	2.587101e-06
58	-0.735561372	5.410505e-01	-3.979759e-01	2.927357e-01	-2.153251e-01
59	-1.035096938	1.071426e+00	-1.109029e+00	1.147953e+00	-1.188243e+00
60	0.674329885	4.547208e-01	3.066318e-01	2.067710e-01	1.394319e-01
61	-0.585580004	3.429039e-01	-2.007977e-01	1.175831e-01	-6.885432e-02
62	-0.638191257	4.072881e-01	-2.599277e-01	1.658836e-01	-1.058655e-01
63	-0.429974662	1.848782e-01	-7.949295e-02	3.417995e-02	-1.469651e-02
64	-2.217300827	4.916423e+00	-1.090119e+01	2.417121e+01	-5.359485e+01
65	1.148088074	1.318106e+00	1.513302e+00	1.737404e+00	1.994693e+00
66	-0.427459889	1.827220e-01	-7.810631e-02	3.338731e-02	-1.427174e-02
67	-0.854097480	7.294825e-01	-6.230492e-01	5.321447e-01	-4.545035e-01
68	-0.563713905	3.177734e-01	-1.791333e-01	1.009799e-01	-5.692378e-02
69	-0.048763047	2.377835e-03	-1.159505e-04	5.654098e-06	-2.757111e-07
70	0.490715344	2.408015e-01	1.181650e-01	5.798539e-02	2.845432e-02
71	0.086216320	7.433254e-03	6.408678e-04	5.525326e-05	4.763733e-06
72	-0.770394280	5.935073e-01	-4.572347e-01	3.522510e-01	-2.713721e-01

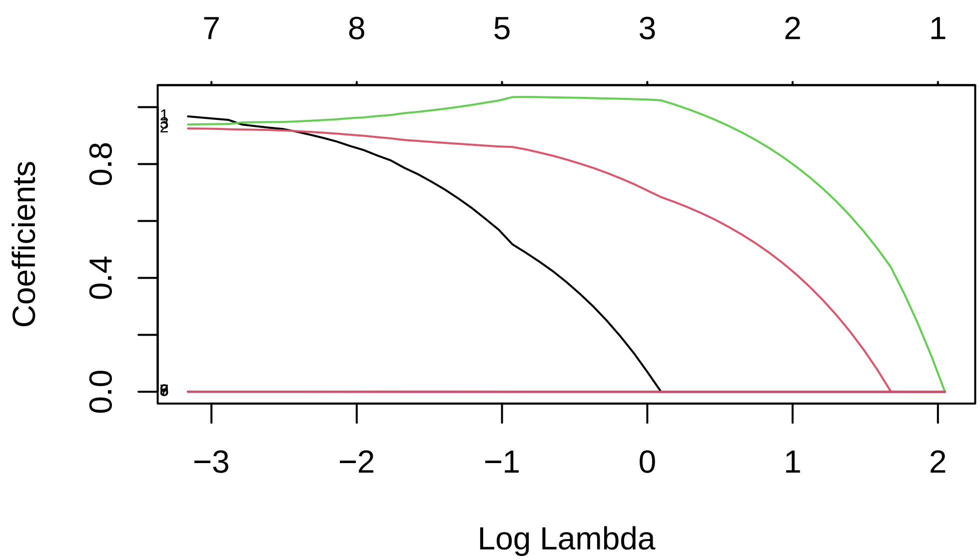
73	0.055846522	3.118834e-03	1.741760e-04	9.727126e-06	5.432261e-07
74	0.324076788	1.050258e-01	3.403641e-02	1.103041e-02	3.574700e-03
75	-0.062103046	3.856788e-03	-2.395183e-04	1.487482e-05	-9.237714e-07
76	-0.578689046	3.348810e-01	-1.937920e-01	1.121453e-01	-6.489725e-02
77	0.883064234	7.798024e-01	6.886156e-01	6.080918e-01	5.369842e-01
78	-0.948218965	8.991192e-01	-8.525619e-01	8.084153e-01	-7.665548e-01
79	-0.097289627	9.465272e-03	-9.208727e-04	8.959137e-05	-8.716311e-06
80	-0.497843205	2.478479e-01	-1.233894e-01	6.142856e-02	-3.058179e-02
81	-0.106839247	1.141462e-02	-1.219530e-03	1.302937e-04	-1.392048e-05
82	0.810029885	6.561484e-01	5.314998e-01	4.305307e-01	3.487428e-01
83	1.098246303	1.206145e+00	1.324644e+00	1.454786e+00	1.597713e+00
84	-0.203280626	4.132301e-02	-8.400168e-03	1.707591e-03	-3.471203e-04
85	0.288839102	8.342803e-02	2.409728e-02	6.960236e-03	2.010388e-03
86	1.294466746	1.675644e+00	2.169066e+00	2.807783e+00	3.634582e+00
87	0.972215436	9.452029e-01	9.189408e-01	8.934084e-01	8.685855e-01
88	0.146250069	2.138908e-02	3.128155e-03	4.574929e-04	6.690836e-05
89	-0.006086582	3.704648e-05	-2.254865e-07	1.372442e-09	-8.353481e-12
90	1.015898347	1.032049e+00	1.048457e+00	1.065126e+00	1.082060e+00
91	-1.563649505	2.445000e+00	-3.823123e+00	5.978024e+00	-9.347534e+00
92	1.004122856	1.008263e+00	1.012420e+00	1.016594e+00	1.020785e+00
93	-0.525554537	2.762076e-01	-1.451621e-01	7.629062e-02	-4.009488e-02
94	3.408246767	1.161615e+01	3.959069e+01	1.349348e+02	4.598913e+02
95	-0.009642257	9.297312e-05	-8.964707e-07	8.644000e-09	-8.334767e-11
96	0.699846857	4.897856e-01	3.427749e-01	2.398900e-01	1.678862e-01
97	-0.947872170	8.984617e-01	-8.516268e-01	8.072333e-01	-7.651540e-01
98	-0.713247249	5.087216e-01	-3.628443e-01	2.587977e-01	-1.845868e-01
99	-0.880647746	7.755405e-01	-6.829780e-01	6.014630e-01	-5.296770e-01
100	-0.791794172	6.269380e-01	-4.964059e-01	3.930513e-01	-3.112157e-01

	X1.5	X1.6	X1.7	X1.8	X1.9
1	2.400040e-04	-5.983034e-05	1.491505e-05	-3.718156e-06	9.268954e-07
2	4.049149e+02	1.101348e+03	2.995608e+03	8.147899e+03	2.216186e+04
3	2.488383e-02	-1.344529e-02	7.264787e-03	-3.925326e-03	2.120941e-03
4	4.570761e-02	2.733086e-02	1.634248e-02	9.771985e-03	5.843157e-03
5	2.103350e+01	-3.494595e+01	5.806069e+01	-9.646448e+01	1.602702e+02
6	1.756423e-03	6.100993e-04	2.119200e-04	7.361113e-05	2.556907e-05
7	1.211523e-07	-8.522239e-09	5.994813e-10	-4.216942e-11	2.966331e-12
8	1.185354e+01	1.789874e+01	2.702696e+01	4.081050e+01	6.162353e+01
9	1.850676e-01	1.397072e-01	1.054647e-01	7.961515e-02	6.010134e-02
10	7.310013e-04	2.194006e-04	6.585026e-05	1.976411e-05	5.931942e-06
11	3.937423e-02	-2.296575e-02	1.339520e-02	-7.813004e-03	4.557081e-03
12	4.844444e-02	2.924946e-02	1.766005e-02	1.066267e-02	6.437835e-03
13	1.780682e-01	-1.335624e-01	1.001802e-01	-7.514147e-02	5.636084e-02
14	1.416050e-01	1.022331e-01	7.380824e-02	5.328659e-02	3.847078e-02

15	3.979314e-02	-2.325107e-02	1.358556e-02	-7.938022e-03	4.638173e-03
16	1.361592e+00	1.433469e+00	1.509140e+00	1.588805e+00	1.672676e+00
17	9.937248e-01	-9.926828e-01	9.916418e-01	-9.906020e-01	9.895632e-01
18	2.662361e-03	-9.911623e-04	3.689968e-04	-1.373727e-04	5.114206e-05
19	1.378426e+01	2.134423e+01	3.305048e+01	5.117702e+01	7.924507e+01
20	3.239319e-07	-2.684496e-08	2.224701e-09	-1.843659e-10	1.527881e-11
21	2.769337e+01	4.816946e+01	8.378528e+01	1.457349e+02	2.534893e+02
22	3.400569e-03	-1.318693e-03	5.113703e-04	-1.983021e-04	7.689876e-05
23	3.393710e-11	-6.106427e-13	1.098752e-14	-1.977025e-16	3.557333e-18
24	3.572532e-02	-2.050244e-02	1.176616e-02	-6.752495e-03	3.875195e-03
25	2.106486e-11	3.500673e-13	5.817611e-15	9.668025e-17	1.606685e-18
26	1.814620e-01	-1.365369e-01	1.027340e-01	-7.729984e-02	5.816247e-02
27	2.508478e-03	9.246526e-04	3.408372e-04	1.256364e-04	4.631095e-05
28	1.937643e-01	-1.473961e-01	1.121239e-01	-8.529246e-02	6.488181e-02
29	3.515076e-06	4.334348e-07	5.344571e-08	6.590252e-09	8.126267e-10
30	9.733167e-02	-6.601306e-02	4.477191e-02	-3.036556e-02	2.059477e-02
31	6.987794e-06	9.661920e-07	1.335939e-07	1.847184e-08	2.554074e-09
32	2.699070e+00	-3.184808e+00	3.757961e+00	-4.434261e+00	5.232272e+00
33	1.247521e+01	1.899864e+01	2.893325e+01	4.406278e+01	6.710371e+01
34	1.328710e-01	-9.491508e-02	6.780165e-02	-4.843344e-02	3.459794e-02
35	8.971745e-02	6.002835e-02	4.016390e-02	2.687295e-02	1.798022e-02
36	1.034860e-07	-7.090802e-09	4.858576e-10	-3.329068e-11	2.281057e-12
37	2.787879e-05	4.854595e-06	8.453411e-07	1.472011e-07	2.563244e-08
38	4.550638e-04	-1.262072e-04	3.500228e-05	-9.707522e-06	2.692281e-06
39	5.350758e-03	2.237782e-03	9.358805e-04	3.914019e-04	1.636913e-04
40	4.546779e-09	1.850637e-10	7.532488e-12	3.065884e-13	1.247881e-14
41	5.051628e-01	4.508201e-01	4.023233e-01	3.590435e-01	3.204195e-01
42	1.711104e+01	-2.746772e+01	4.409290e+01	-7.078069e+01	1.136216e+02
43	2.215288e-06	-2.529309e-07	2.887843e-08	-3.297199e-09	3.764583e-10
44	5.209223e-02	3.183477e-02	1.945497e-02	1.188939e-02	7.265881e-03
45	1.901521e-05	-3.106596e-06	5.075376e-07	-8.291856e-08	1.354676e-08
46	1.853363e+01	-3.014999e+01	4.904717e+01	-7.978858e+01	1.297978e+02
47	6.499683e+02	1.912967e+03	5.630188e+03	1.657060e+04	4.877010e+04
48	6.096585e+00	8.240136e+00	1.113736e+01	1.505324e+01	2.034594e+01
49	1.477693e+00	-1.577060e+00	1.683109e+00	-1.796290e+00	1.917081e+00
50	1.587943e-03	5.423846e-04	1.852592e-04	6.327794e-05	2.161348e-05
51	2.926983e-02	1.624888e-02	9.020422e-03	5.007606e-03	2.779928e-03
52	4.181534e-06	5.307519e-07	6.736705e-08	8.550736e-09	1.085324e-09
53	4.372382e+01	8.206749e+01	1.540367e+02	2.891194e+02	5.426630e+02
54	6.389472e-02	-4.039948e-02	2.554387e-02	-1.615093e-02	1.021194e-02
55	1.036045e-01	7.100270e-02	4.865991e-02	3.334784e-02	2.285410e-02
56	2.182942e-01	1.693880e-01	1.314386e-01	1.019914e-01	7.914140e-02
57	1.974132e-07	1.506395e-08	1.149480e-09	8.771304e-11	6.693093e-12

58	1.583848e-01	-1.165017e-01	8.569418e-02	-6.303333e-02	4.636488e-02
59	1.229946e+00	-1.273114e+00	1.317796e+00	-1.364047e+00	1.411920e+00
60	9.402307e-02	6.340257e-02	4.275425e-02	2.883047e-02	1.944124e-02
61	4.031971e-02	-2.361042e-02	1.382579e-02	-8.096105e-03	4.740917e-03
62	6.756241e-02	-4.311774e-02	2.751736e-02	-1.756134e-02	1.120749e-02
63	6.319128e-03	-2.717065e-03	1.168269e-03	-5.023261e-04	2.159875e-04
64	1.188359e+02	-2.634950e+02	5.842476e+02	-1.295453e+03	2.872408e+03
65	2.290083e+00	2.629217e+00	3.018573e+00	3.465587e+00	3.978800e+00
66	6.100595e-03	-2.607760e-03	1.114713e-03	-4.764950e-04	2.036825e-04
67	3.881903e-01	-3.315523e-01	2.831780e-01	-2.418616e-01	2.065734e-01
68	3.208873e-02	-1.808886e-02	1.019694e-02	-5.748158e-03	3.240317e-03
69	1.344451e-08	-6.555953e-10	3.196883e-11	-1.558897e-12	7.601659e-14
70	1.396297e-02	6.851844e-03	3.362305e-03	1.649935e-03	8.096483e-04
71	4.107115e-07	3.541004e-08	3.052923e-09	2.632118e-10	2.269315e-11
72	2.090635e-01	-1.610614e-01	1.240807e-01	-9.559110e-02	7.364283e-02
73	3.033729e-08	1.694232e-09	9.461697e-11	5.284029e-12	2.950946e-13
74	1.158477e-03	3.754356e-04	1.216700e-04	3.943041e-05	1.277848e-05
75	5.736902e-08	-3.562791e-09	2.212602e-10	-1.374093e-11	8.533536e-13
76	3.755533e-02	-2.173286e-02	1.257657e-02	-7.277921e-03	4.211653e-03
77	4.741915e-01	4.187416e-01	3.697757e-01	3.265357e-01	2.883520e-01
78	7.268618e-01	-6.892241e-01	6.535354e-01	-6.196946e-01	5.876062e-01
79	8.480066e-07	-8.250225e-08	8.026613e-09	-7.809062e-10	7.597407e-11
80	1.522494e-02	-7.579631e-03	3.773468e-03	-1.878595e-03	9.352460e-04
81	1.487253e-06	-1.588970e-07	1.697644e-08	-1.813750e-09	1.937797e-10
82	2.824921e-01	2.288270e-01	1.853567e-01	1.501445e-01	1.216215e-01
83	1.754682e+00	1.927073e+00	2.116401e+00	2.324330e+00	2.552687e+00
84	7.056282e-05	-1.434405e-05	2.915868e-06	-5.927396e-07	1.204925e-07
85	5.806787e-04	1.677227e-04	4.844488e-05	1.399278e-05	4.041661e-06
86	4.704846e+00	6.090266e+00	7.883647e+00	1.020512e+01	1.321019e+01
87	8.444522e-01	8.209895e-01	7.981786e-01	7.760016e-01	7.544407e-01
88	9.785353e-06	1.431108e-06	2.092997e-07	3.061010e-08	4.476729e-09
89	5.084415e-14	-3.094671e-16	1.883597e-18	-1.146467e-20	6.978065e-23
90	1.099263e+00	1.116739e+00	1.134494e+00	1.152530e+00	1.170853e+00
91	1.461627e+01	-2.285472e+01	3.573677e+01	-5.587978e+01	8.737639e+01
92	1.024994e+00	1.029219e+00	1.033463e+00	1.037724e+00	1.042002e+00
93	2.107205e-02	-1.107451e-02	5.820259e-03	-3.058864e-03	1.607600e-03
94	1.567423e+03	5.342164e+03	1.820741e+04	6.205536e+04	2.115000e+05
95	8.036596e-13	-7.749093e-15	7.471874e-17	-7.204573e-19	6.946834e-21
96	1.174947e-01	8.222826e-02	5.754719e-02	4.027422e-02	2.818579e-02
97	7.252682e-01	-6.874615e-01	6.516257e-01	-6.176578e-01	5.854607e-01
98	1.316560e-01	-9.390327e-02	6.697625e-02	-4.777063e-02	3.407227e-02
99	4.664589e-01	-4.107860e-01	3.617577e-01	-3.185811e-01	2.805578e-01
100	2.464188e-01	-1.951130e-01	1.544893e-01	-1.223237e-01	9.685521e-02

```
y <- datos$y
modelo_lasso <- glmnet(x, y, alpha = 1, intercept=FALSE)   # alpha = 1 (lasso); alpha = 0
plot(modelo_lasso, xvar='lambda', label=TRUE)
```




```
coef(modelo_lasso, s = 0.1)
```

```
11 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) .
X1          9.009201e-01
X1.1        9.125028e-01
X1.2        9.525856e-01
X1.3        .
X1.4        8.155427e-06
X1.5        1.535048e-04
X1.6        3.372341e-04
X1.7        1.175133e-04
X1.8        6.702356e-08
X1.9        .
```

```
#help(coef)
```

La ventaja que tiene lasso es que permite eliminar ciertos coeficientes esto queda de manifiesto en que la mayoría son nulos.

Ejercicio 12

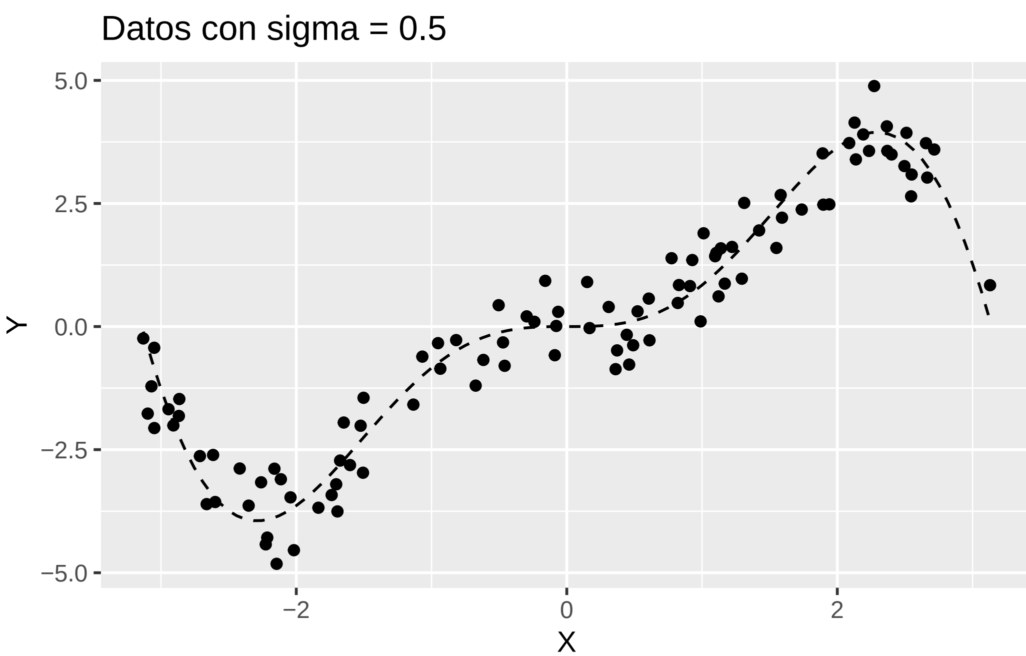
Se generan los siguientes dos conjuntos de datos:

```r
library(purrr)
library(ggplot2)
```

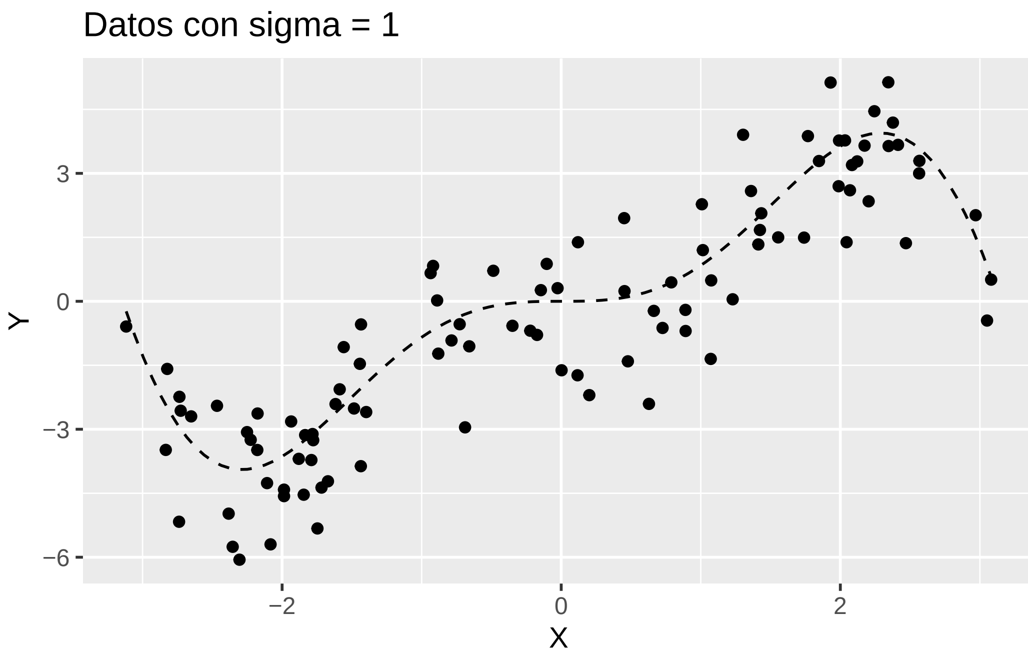
```r
n <- 100
sigma_1 <- 0.5
sigma_2 <- 1

fun_reg <- function (x) (x^2*sin(x))

generator <- function (n, sigma) {
  error <- rnorm(n)
  unif <- runif(n, min = -pi, max = pi)
  Y <- (
    unif*unif
    *
    sapply(unif, sin)
    +
    sigma * error
  )

  return (
    data.frame(
      X= unif,
      Y = Y
    )
  )

}
X_1 <- generator(n, sigma_1)
X_2 <- generator(n, sigma_2)
```

Representación gráfica

```
p <- ggplot(X_1) +
  geom_point( aes(X, Y), colour = 'black' )+
  ggtitle('Datos con sigma = 0.5') +
  geom_function(fun = 'fun_reg', linetype = 2)

q <- ggplot(X_2) +
  geom_point( aes(X, Y), colour = 'black' )+
  ggtitle('Datos con sigma = 1') +
  geom_function(fun = 'fun_reg', linetype = 2)
p
```




```
q
```



Puede verse como el sigma aumenta la dispersión respecto al eje Y.

Ajuste de regresión de mínimos cuadrados

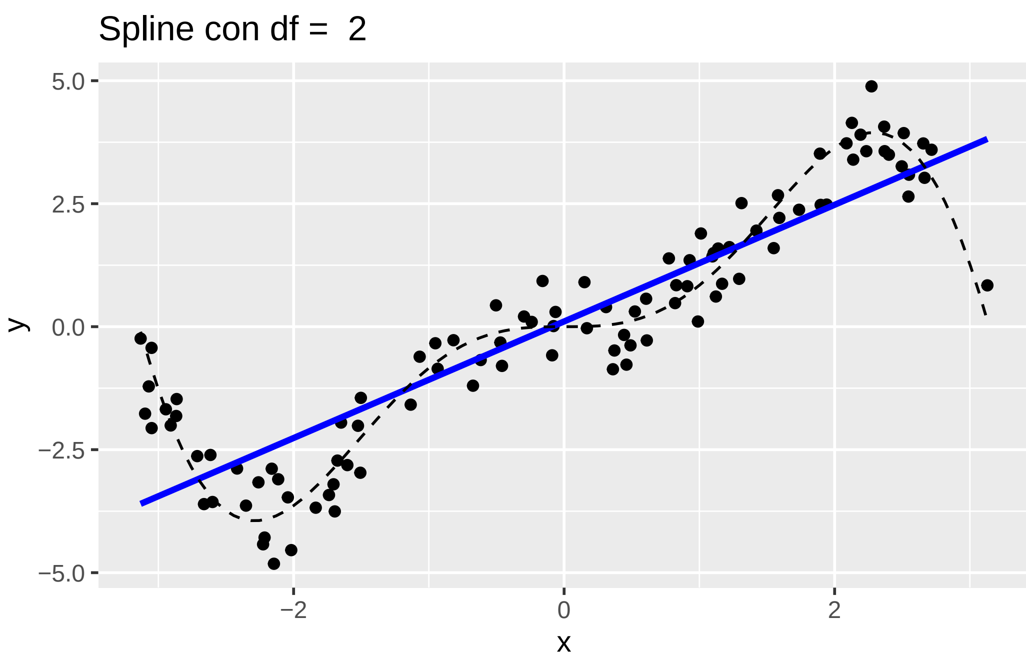
Para cada conjunto de datos se pretende ajustar una regresión de mínimos cuadrados penalizada prefijando uno de los grados de libertad efectivos.

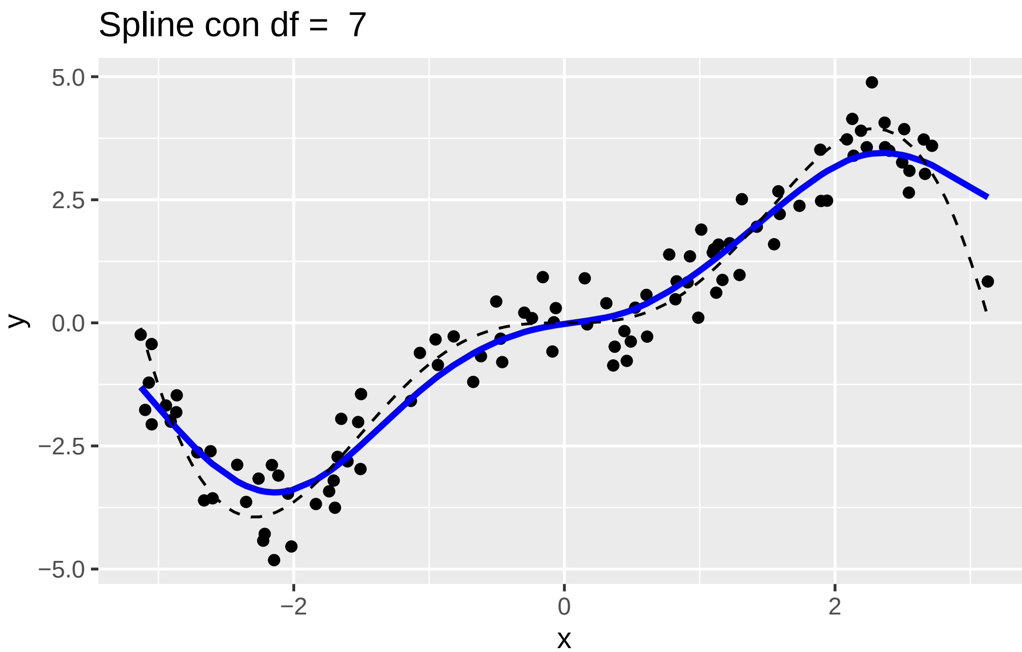
Representación gráfica del error de predicción de validación cruzada generalizando en dunción de los grados de liberta utilizados.

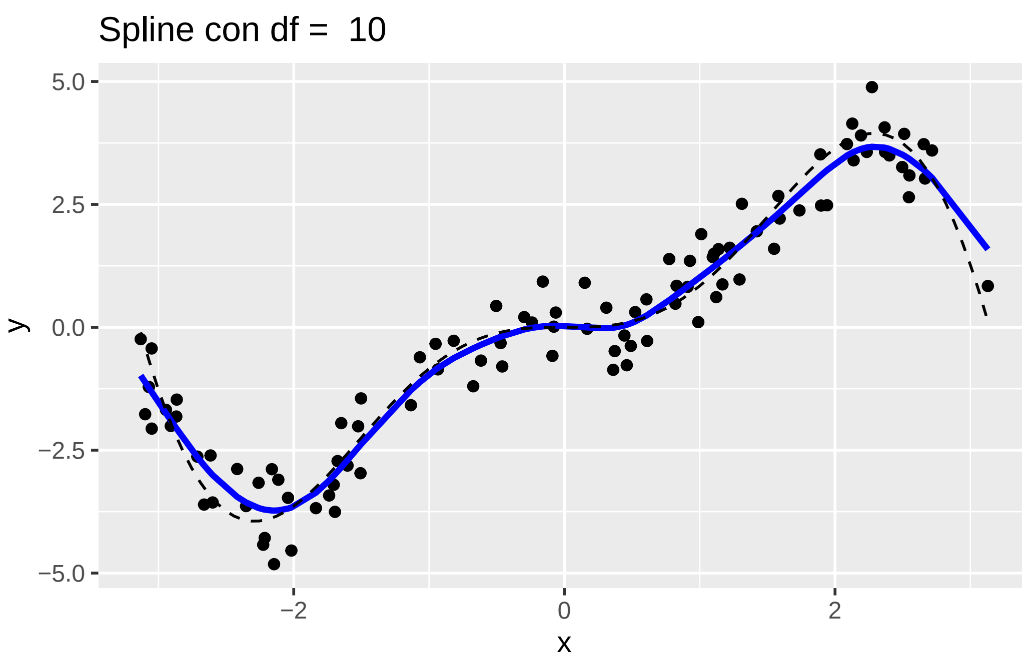
Imprimimos primero algunos ajustes:

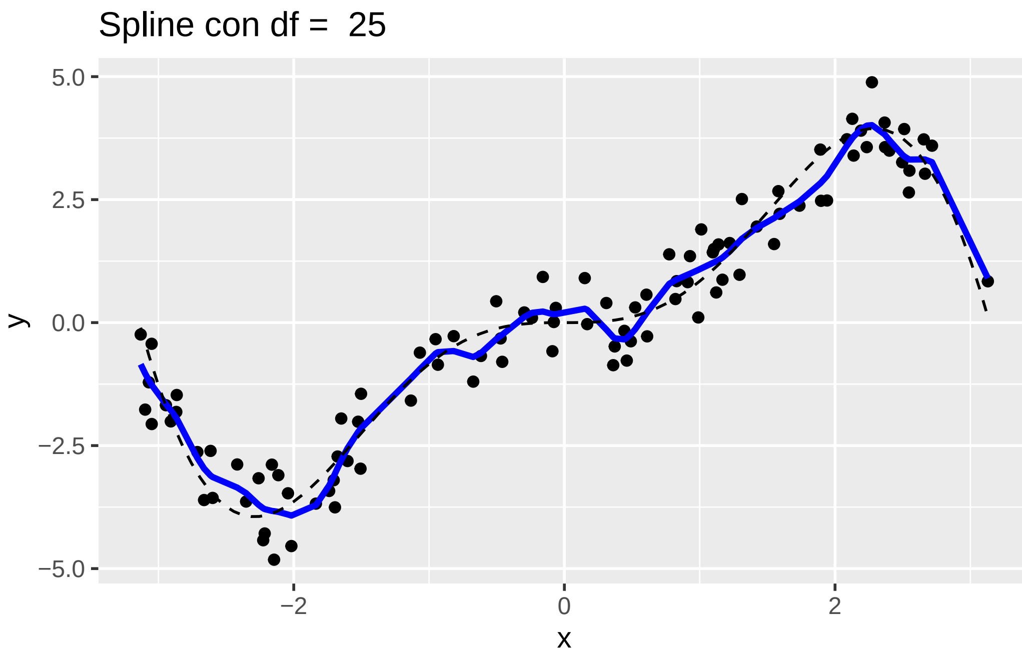
```
pinta_spline <- function (X, grados_libertad){
spline_1 <- smooth.spline(
  X$X, X$Y, df = grados_libertad
)
datos <- data.frame(
  x = X$X,
  y = X$Y,
  xfit = spline_1$x,
  yfit = spline_1$y
```

```
)

titulo <- paste('Spline con df = ', grados_libertad)
ggplot(datos) +
  geom_point(aes(x, y)) +
  geom_line(aes(xfit, yfit), color="blue", size = 1.1) +
  geom_function(fun = 'fun_reg', linetype = 2) +
  ggtitle(titulo)
}

for( df in c(2,7,10, 25) ){
  print(pinta_spline(X_1, df))
}
```



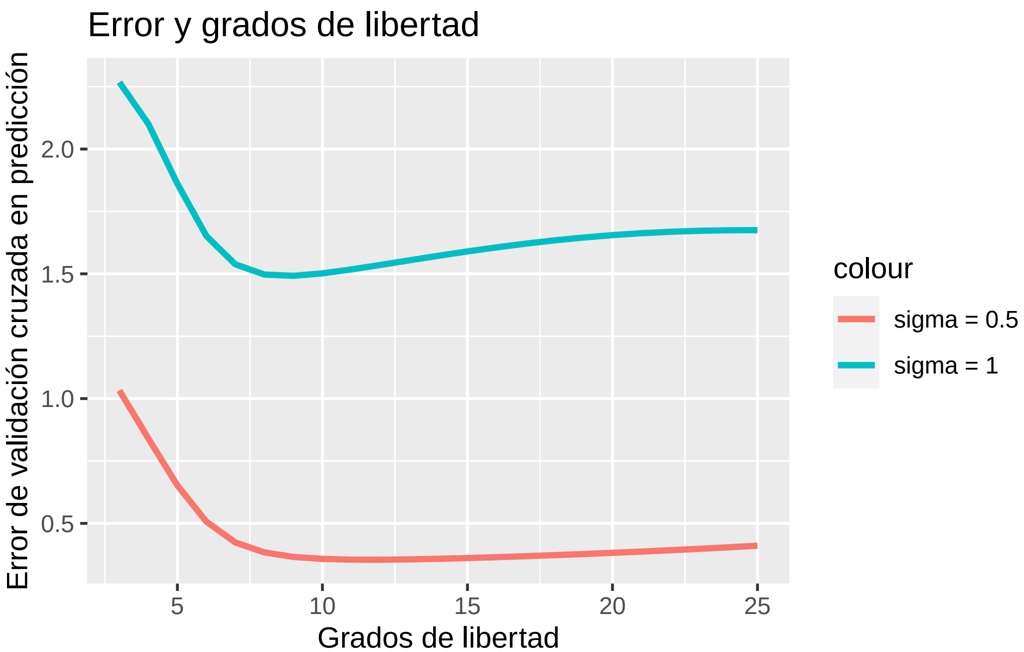

Spline con df = 7
5.0
2.5
0.0
−2.5
−5.0
y
−2
0
2
x


Spline con df = 10
5.0
2.5
0.0
−2.5
−5.0
y
−2
0
2
x



Bajo una inspección visual puede verse que a mayor número de grados de libertad mejor es el ajuste a los datos utilizados en regresión c, procedamos a costatarlo

```
cv_error <- function (X, grados_libertad){
  spline_1 <- smooth.spline(
    X$X, X$Y, df = grados_libertad
  )
  return (spline_1$cv.crit)
}

df <- c(3:25)
datos <- data.frame(
  x = df,
  y1 = sapply(df, function(d)(cv_error(X_1, d))),
  y2 = sapply(df, function(d)(cv_error(X_2, d)))
)

ggplot(datos) +
  geom_line(aes(x, y1, color="sigma = 0.5"), size = 1.1) +
  geom_line(aes(x, y2, color="sigma = 1") , size = 1.1) +
  labs(title = 'Error y grados de libertad',
       x = 'Grados de libertad',
       y = 'Error de validación cruzada en predicción')
```




```
cat("Pasa sigma = 0.5 alcanza un mínimo en ", datos$x[which.min(datos$y1)], " grados de li
```

```
Pasa sigma = 0.5 alcanza un mínimo en  12  grados de libertad.
```

```
cat("Pasa sigma = 1 alcanza un mínimo en ", datos$x[which.min(datos$y2)], " grados de libe
```

```
Pasa sigma = 1 alcanza un mínimo en  9  grados de libertad.
```

Comentario de los resultados

Hay varios fenómenos llamativos al observa esta gráfica: 1. Comportamiento general: Decrece, alcanza un mínimo y vuelve a crecer en menor medida. 2. El que los dos gráficos parezcan más o menos desplazados. 3. Que σ menor admita más grados de libertad antes de volver a crecer el error. 4. Que la gráfica de $\sigma = 1$ comience a crecer más que la $\sigma = 0.5$.

Todos estos fenómenos se pueden explicar con la relación entre los errores vista en teoría:

$$\text{Error}_{Test} = \text{Error}_{Training} + \frac{2\sigma^2}{n}\text{grados de libertad.}$$

(Notemos que grados de libertad se corresponde a la traza de la matriz M, que en nuestro caso particular se trata de los mínimos cuadrados)

Comportamiento general: Decrecemiento, mínimo y crecimiento más lento

Como rasgo general podemos observar como añadir más grados de libertad mejora el ajuste a los datos de *aprendizaje*, el error de *training* está decreciendo en mayor medida que el aumento de grados de libertad. A partir de 10 (o 13 si $\sigma = 1$) grados puede verse cómo el error de test comienza a crecer de nuevo, este fenómeno conocido como *sobreajuste a los datos de entrenamiento* no es más que el error de training se disminuye en menor proporción que el peso que suma el término $\frac{2\sigma^2}{n}$grados de libertad.

Gráficos desplazados

Esto está motivado por que los *datos training* de uno es mayor que la de otro. Este fenómenos es natural, ya que por la propia naturaleza de los datos sabemos que los datos de $\sigma = 1$ (la gráfica con mayor error) posee un ruido mayor (concretamente el de $\mathcal{N}(0, 1)$)

A σ menor admita más grados de libertad antes de volver a crecer el error.

Relacionado con lo anterior, si el *ruido* es menor será más similar a su desarrollo de Taylor admitiendo por tanto desarrollar más términos de la serie de Taylor (grados de libertad en nuestro caso).

Cremiento mayor del error de $\sigma = 1$ frente a $\sigma = 0.5$

Esto es claro resultado del segundo sumando de la relación mostrada:

$$\frac{2\sigma^2}{n}\text{grados de libertad}$$

ya que en ambos caso n y *grados de libertad* son iguales y solo difiere el valor de σ.