Construcción de modelos de regresión y reglas de predicción paramétricos y no-paramétricos.

Aplicación:Datos de inmisión SO_2 y NO_x

Técnicas de remuestreo. Curso 2010-2011

Leyenda Rodríguez, María

Índice

1. Introducción

En este documento, se trabajará con datos de inmisión minutales tomados en tiempo real de SO_2 y NO_x. Estos datos fueron tomados entre las 05:00 y las 09:00 del día 12 de Marzo de 2007 en la estación de inmisión G2, la cual es es propiedad de U.P.T. de As Pontes, propiedad de Endesa s.a. Notemos que por comodicidad, se ha trabajado con toda la muestra sino con una submuestra diseñada de tal forma que a cada valor de SO_2 le corresponda un solo valor de NO_x.

Dado que los valores de inmisión de SO_2 y NO_x presentan cointegración, tiene sentido plantear un el bootstrap de un modelo de regresion para estos valores. En primer lugar, se realizará un estudio descriptivo de las variables que se van a estudiar. En segundo lugar, se estudiará la relación entre SO_2 y NO_x mediante un modelo de regresión lineal de diseño fijo y heterocedástico, dónde SO_2 es la variable independiente y NO_x la variable dependiente. Para ello,se realizarán varias estimaciones de la densidad de los coeficientes del modelo de regresión; mediante la teoría clásica y mediante los bootstraps uniforme y Wild bootstrap. También se realizará la construcción de modelos no paramétricos de regresión utilizando los estimadores de Nadaraya-Watson y Local-lineal. Finalmente, se interpretarán los modelos de regresión construídos como una regla de predicción y se calcula el error real, el error aparante y el optimismo esperado mediante Boostrap uniforme.

2. Análisis descriptivo de las variables de inmisión de SO_2 y NO_x

Como nuestro objetivo es trabajar con modelo de regresión tendremos que observar, quien explica mejor si el SO_2 al NO_x o si el NO_x al SO_2. Para ello representamos los correspondientes gráficos de sipersión con sus respectivas rectas de regresión (Figura 1), de donde concluímos que los dos explican lo mismo un 80 %. Por lo que vamos a estudiar como explica el SO_2 al más NO_x ya que es más natural; pues, en general, una súbida de NO_x no tiene porque implicar una subida de SO_2 en cambio una súbida de SO_2 sí suele implicar una subida de NO_x.

A continuación, se representa el gráfico de dispersión de las variables en estudio (Figura 2) y la tabla de los estadísticos más representativos de las variables SO_2 y NO_x(Tabla 1) . A partir del gráfico de dispersión se observa una tendencia creciente y a partir de la tabla se observa que las variables tienen datos missing (-1) por tanto estos deben ser eliminados. Además también se puede observar que las medias de las dos variables de estudio son muy diferentes, pues la media de NO_2 es mucho mayor que la de la variable SO_2, lo que implica una diferencia de escala. Con el fin de obtener resulatados más visibles se construye una submuestra de las variables en estudio de modo que a cada valor de SO_2 le corresponda un único valor de NO_x. Los restadísticos más representativos para estas nuevas variables(Tabla 2) son muy similares a los anteriores, por tanto esta nueva muestra de tamño 36 representa bastante bien a la anterior de tamaño 239. A continuación, para observar la relación entre las variables representamos los gráficos de dispersión de la nueva muestra de NO_x frente a SO_2(Figura 3). En este gráfico de dispersión se observa una tendencia creciente. En este gráfico, se refeja mejor SO_2 a NO_x, por tanto a partir de ahora solo trabajeromos con la nueva submuestra contruída.

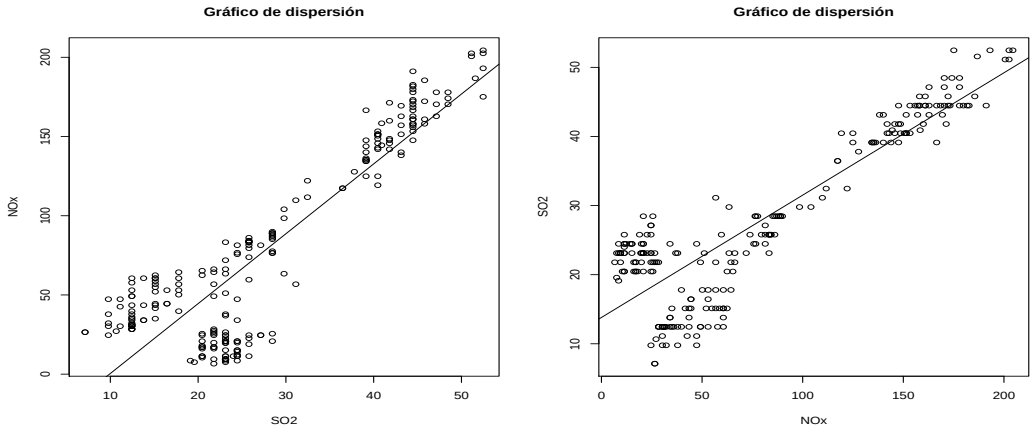

Figura 1: Gráfico de dispersión de NO_x frente a SO_2

Completamos el estudio con la estimación de las densidades de las variables de inmisión de SO_2 (Figura4)y NO_x(Figura5) mediante el histograma (1),el histograma movil (y el estimador de Parzen-Rosemblat (3). A la vista de estos dos gráficos se concluye que la estimación más suave es

Valores de interés	SO_2	NO_x
Mínimo	-1	-1
1º Cuantil	20.46	25.55
Mediana	24.46	58.68
Media	27.46	77.78
3ºCuantil	40.47	140.07
Maximo	52.47	204.42

Cuadro 1: Valores de los estadísticos de las variiables de inmisión SO_2 y NO_x

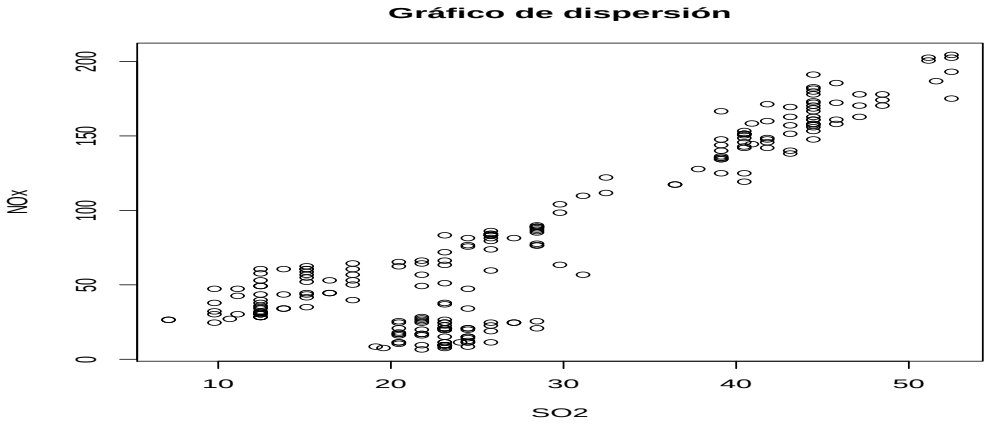

Figura 2: Gráfico de dispersión de NO_x frente a SO_2

Valores de interés	SO_2	NO_x
Mínimo	7.12	6.62
1º Cuantil	18.79	23.66
Mediana	27.80	50.63
Media	29.51	80.67
3ºCuantil	41.13	142.60
Maximo	52.47	202.50

Cuadro 2: Valores de los estadísticos de las variiables de inmisión SO_2 y NO_x

la proporcionada por el estimador de Parzen-Rosemblat. Además tras realizar el test de Shapiro-Wilks se obiene que SO_2 sigue una distribución normal (p.valor= 0.09588) y en cambio, NO_x no sigue una distribución normal

$$f_h(x) = \frac{\#(x_i \in A_j)}{\frac{max(x_i)-min(x_i)}{N}}, \hat{\forall} x \in I_j, j \in \{1, ..., N\} (1)$$

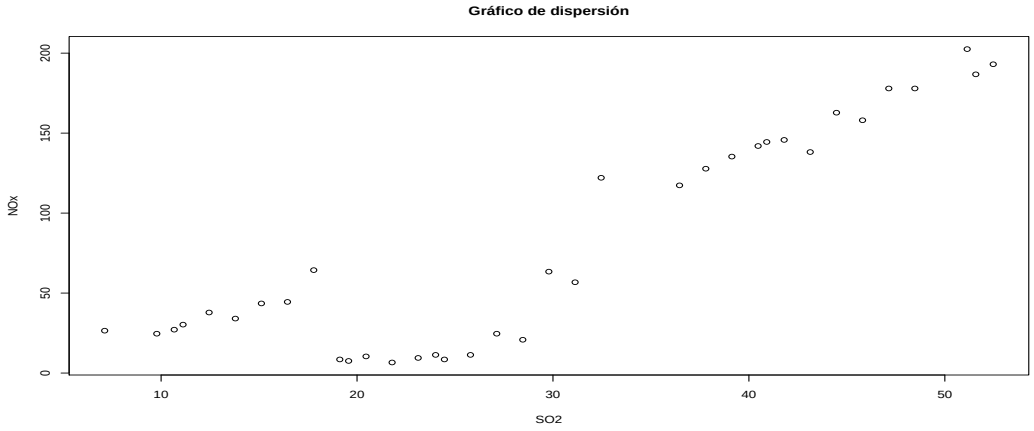

Figura 3: Gráfico de dispersión de NO_x frente a SO_2

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x-x_i}{h}), siendo K(\frac{x-x_i}{h}) = \frac{1}{2}I_{x_i \in (x-h, x+h)} \tag{2}$$

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x-x_i}{h}), siendo K una funcin de densidad cualquiera; en este caso N(0,1) \tag{3}$$

Notemos que h es el paramétro de suavización o ventana y se escoge de tal manera que minimice el error cuadrático medio asintótico (AMISE).

Figura 4: Estimación de la densidad de SO_2

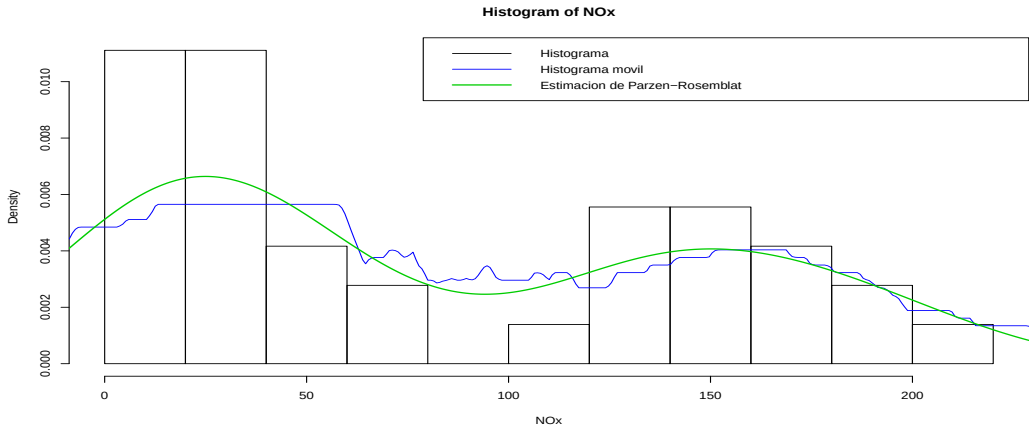

Figura 5: Estimación de la densidad de NO_x

3. Estudio de modelos de regresión paramétricos y no parametricos de NO_x frente a SO_2

En este trabajo, nos interesa analizar la relación existente entre dos variables, X e Y . El análisis de regresión estudia de que forma NO_x (la variable dependiente) se puede explicar a partir de SO_2. Si NO_x depende de SO_2 entonces $NO_x = \mathrm{m}(SO_2)$,donde m es una función. El análisis de la información empírica disponible nos debería de proporcionar información sobre m.

$$E(NO_x SO_2 = x) = \frac{\int yf(x,y)}{f_{SO_2}(x)} \tag{4}$$

3.1. Modelo de regresión paramétrico. Cálculo de la densidad de sus coeficientes mediante la Teoría Clásica, Boostrap y Wild-Boostrap

En regresión paramétrica habitualmente se supone que m depende linealmente de un vector de parámetros. En nuestro caso particular, hemos empleado la regresión lineal simple, por tanto hemos supuesto que $\mathrm{m}(SO_2) = \mathrm{a} + \mathrm{b}SO_2$. Construimos el modelo de regresión paramétrica (4) aunque dicho modelo no se ajusta bien a los datos pues solo explica el 80 % pues $\mathrm{R}^2 = 0{,}8028; conloqueconclumosqueesunbuenmodeloparamodelizarlarelacinentre\mathrm{NO}_x$ y SO_2, es decir, para explicar NO_x a partir de SO_2.

NO_x=-50.3627+ 4.4404 $*SO_2$ (5)

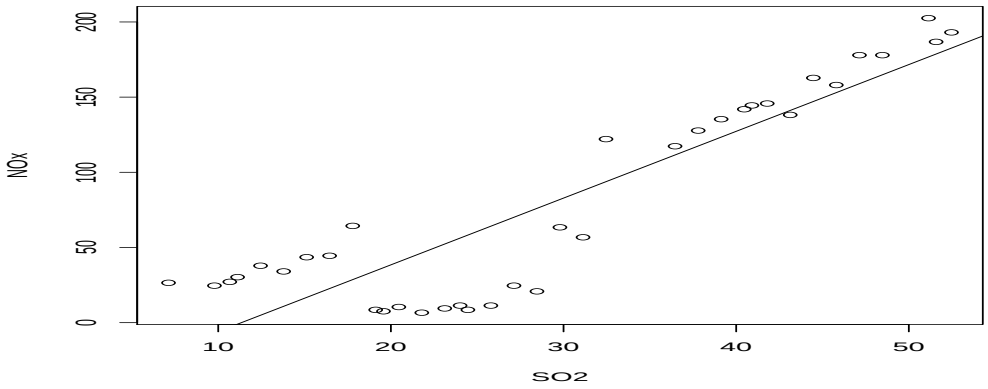

Figura 6: Modelo de regresion

En (Figura 7)se puede observar que la Teoría Clásica nos proporciona densidades normales, lo que parece que funcioa bastante vien pues SO_2 sigue una distribución normal y NO_x sigue

una distribución aproximadamente normal. En este caso, la teorí dice que Wild Boostrap funciona mejor que Boostrap uniforme. Lo que es debido a que el modelo de regresión que estamos considerando es heterocedástico, lo que significa que la varianza del error es función de la parte predictora del dato. Luego, el Boostrap uniformenormal. es inadecuado ya que no disponemos de n datos undependientes e identicamente distribuídos, sino de un único dato para cada uno de los n errores correspondientes; es decir, no disponemos de una muestra de n datos i.i.d. sino de n muestras de un dato. A pesar de todo esto no podemos afirmar nada ya que no tenemos las densidades reales con las que comparar.

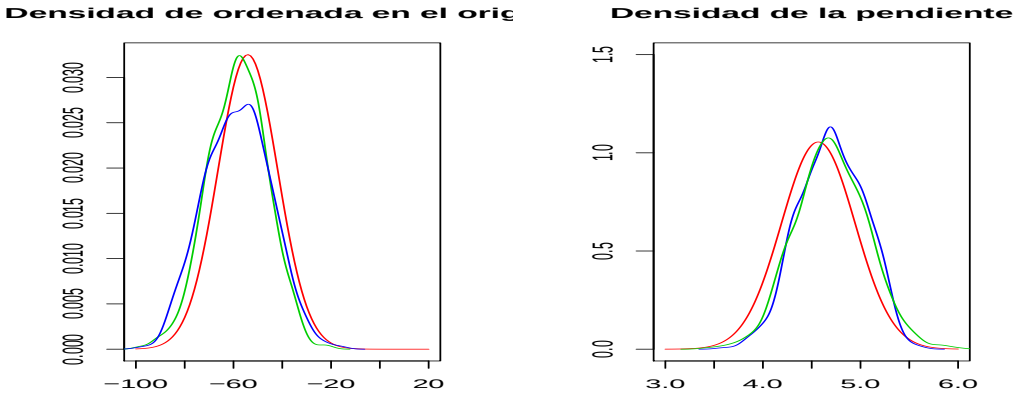

Figura 7: Densidades de los coeficientes estimadas por la Teoría Clásica (roja), boostrap uniforme (verde) y Wild Boostrap (azul)

3.2. Construcción de modelos de regresión no-paramétricos: Nadaraya-Watson, Local-lineal

El modelo anterior es muy restrictivo, pues no permite que el consumo se incremente hasta un cierto nivel a partir del cual baja o se mantiene estable. En los modelos de regresión no paramétricos no se impone ninguna restricción a priori sobre m. Obviamente existe un precio a pagar por esta flexibilidad. Para obtener un estimador no paramétrico de la regresión, basta con estimar m. Para ello habrá que estimar $f_{SO_2}(x)$ y $f(x, y)$. Para estimar la densidad bivariante f(x, y) es habitual emplear el estimador tipo núcleo con núcleo producto.

$$\hat{f}_{n,K}(x, y) = \frac{1}{n}\sum_{j=1}^{n} K_h(x - X_j)K_g(y - Y_j) \tag{6}$$

donde X=SO_2 e Y=NO_x y x son valores de SO_2 e y son valores de NO_x Por tanto el estimador del numerador de la media condicional sería

$$\hat{m}_{n,K}(x, y) = \frac{1}{n}\sum_{j=1}^{n} K_h(x - X_j)Y_j \tag{7}$$

Tanto el estimador Nadaraya-Watson como el estimador lineal-local son estimadores lineales. Luego, aplicaremos un resultado general para estimadores lineales. Sea

$$\hat{m}(x) = \sum_{j=1}^{n} l_j(x)Y_j \tag{8}$$

un estimador lineal. Se define

$$\hat{m}_{(-i)}(x) = \sum_{j=1}^{n} l_{j,(-i)}(x)Y_j \tag{9}$$

donde

$$l_{j,(-i)}(x) = \begin{cases} 0 sij = i \\ \frac{l_j(x)}{\sum_{k\neq i} l_k(x)} sij \neq i \end{cases} \tag{10}$$

La función de validación cruzada para un estimador lineal se define como

$$CV(h) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{m}_{-(i)}(X_i))^2 \tag{11}$$

Teorema

$$CV(h) = \frac{1}{n}\sum_{i=1}^{n}(\frac{Y_i - \hat{m}(X_i)}{1 - L_{ii}})^2 \tag{12}$$

donde L_{ii} es el elemento i-de la diagonal de la matriz de suavizado L necesaria para calcular el estimador en los puntos $(X_1, ..., X_n)$. Es decir

$$L_{ii} = l_i(x_i) \tag{13}$$

3.2.1. Estimador Nadaraya-Watson

El estimador de la función m resultante de reemplazar las cantidades desconocidas por sus estimadores en la fórmula de la esperanza condicional fue propuesto por Nadaraya y Watson en 1964

$$\hat{m}_{n,K}(x) = \frac{\sum_{j=1}^{n} K_h(x - X_j)Y_j}{\sum_{k=1}^{n} K_h(x - X_j)} = \sum_{j=1}^{n} W_{h,j}(x)Y_j \tag{14}$$

donde $K_h(u) = u/h$ y siendo $W_{h,j}$

$$W_{h,j}(x) = \frac{K_h(x - X_j)}{\sum_{k=1}^{n} K_h(x - X_j)} \tag{15}$$

Por tanto el estimador tipo núcleo de la función de regresión es una media (local) ponderada de los valores observados de la variable Y donde

$$\sum W_{h,j}(x) = 1 \tag{16}$$

Una posibilidad para elegir el par´ametro de suavizado es usar el método de validación cruzada, convenientemente adaptado al contexto de regresión. Para medir la bondad de ajuste que se consigue con una ventana h podríamos usar el error medio

$$\sum CV(h) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{m}_{n,K}(X_i))^2 \tag{17}$$

Esta medida de error global aproximaría el error de predicción. Sin embargo, la aproximación sería un tanto optimista ya que estaríamos usando el valor de Yi dos veces: una a la hora de medir el error, y otra a la hora de construir el estimador.

Para evaluar mejor el error de predicción se suele eliminar el dato i-ésimo cuando calculamos el error de predicción para Yi. Así la función de validación cruzada se define como

$$\sum CV(h) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{m}_{-(i),K}(X_i))^2 \tag{18}$$

donde $\hat{m}_{-(i),K}$ denota el estimador de Nadaraya-Watson construido a partir de la muestra original después de eliminar el par (X_i, Y_i). La idea sería tomar aquel h que haga que CV sea mínimo. Aunque se podría calcular directamente CV , esto requeriría evaluar, para cada h, n veces el estimador de Nadaraya-Watson,construido a partir de una muestra de (n - 1) puntos. Muchos de estos cálculos serían redundantes y se pueden simplificar.

Teorema La función de validación cruzada del estimador de Nadaraya-Watson se puede escribir de la siguiente forma

$$CV(h) = \frac{1}{n}\sum_{i=1}^{n}(\frac{Y_i - \hat{m}_{n,K}(X_i)}{1 - L_{ii}})^2 \tag{19}$$

donde L_{ii} es el elemento i-de la diagonal de la matriz de suavizado L necesaria para calcular el estimador en los puntos $(X_1, ..., X_n)$.$Esdecir L_{ii} = \frac{K(0)}{\sum_{k=1}^{n} K(X_i - X_k/h)}(20)$

Notemos que K es una función de densidad a la cual se le denomina función núcleo y en nuestro caso particular hemos escogido núcleo gausiano y h es el paramétro de suavizado o ventana, que la escogemos mediante validación cruzada, es decir , tomamos h de modo que haga que CV sea mínimo. En nuestro caso particular h=2.7. Por tanto, en estas condiciones hemos obtenido la estimación no paramétrica de la regresión (Figura 9)

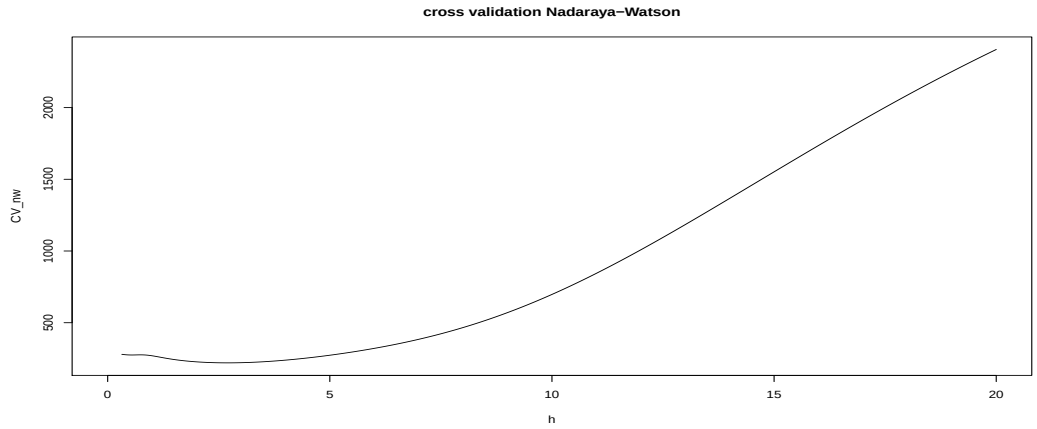

Figura 8: Función de validación cruzada para el estimador Nadaraya-Watson

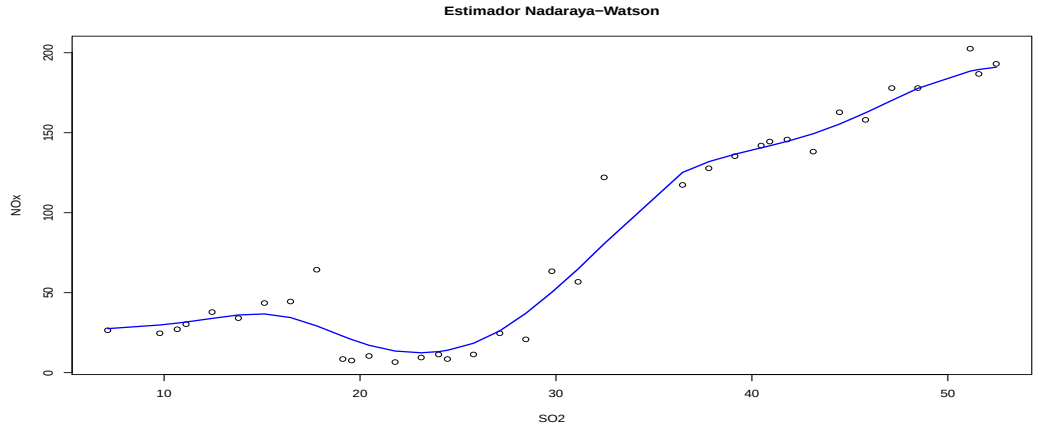

Figura 9: Función de validación cruzada para el estimador de Nadaraya-Watson

3.2.2. Estimador Local-lineal

La idea de este método es muy sencilla. En lugar de hacer un ajuste global por mínimos cuadrados de una recta podemos intentar buscar una recta que ajuste bien sólo en los puntos próximos a x. Dado h >0 podemos proponer un modelo lineal válido sólo en el entorno (x - h, x + h)

$$Y_i = \alpha(x) + \beta(x)X_i + e_i; siendo X_i \in (x-h, x+h) \tag{21}$$

El estimador lineal local en el punto x vendrá dado por

$$\overset{m}{}_{n,LL}(x) = a(x) + b(x)x \tag{22}$$

donde a(x), b(x) son los valores que minimizan la suma de cuadrados ponderada

$$\sum_{i=1}^{n}(Y_i - \alpha(x) - \beta(x)X_i)K_h(x - X_i) \tag{23}$$

donde donde K una funci´on de densidad unimodal y simétrica alrededor del cero que proporciona diferentes pesos a los errores del intervalo (x - h, x + h),dependiendo de su proximidad a x.

Al igual que ocurría con el estimador de Nadaraya-Watson, el estimador lineal-local también es un estimador lineal. Por tanto, la función de validación cruzada es bastante sencilla de calcular. **Teorema** El estimador local lineal se puede escribir de la forma

$$\overset{m}{}_{n,LL}(x) = \sum_{j=1}^{n} l_j(x)Y_j, donde l_j(x) = \frac{b_j(x)}{\sum_{k=1}^{n} b_k(x)} \tag{24}$$

con

$$b_j(x) = K(\frac{x_j - x}{h})(S_{n,2}(x) - (x - x_j)S_{n,1}(x)), \tag{25}$$

y

$$S_{n,r}(x) = \sum_{j=1}^{n} K(\frac{x_j - x}{h})(x - x_j)^r, r = 1, 2 \tag{26}$$

Notemos que hemos construído el estimador Local-lineal seleccionando la ventana por validación cruzada, h=2.38,(Figura 11)

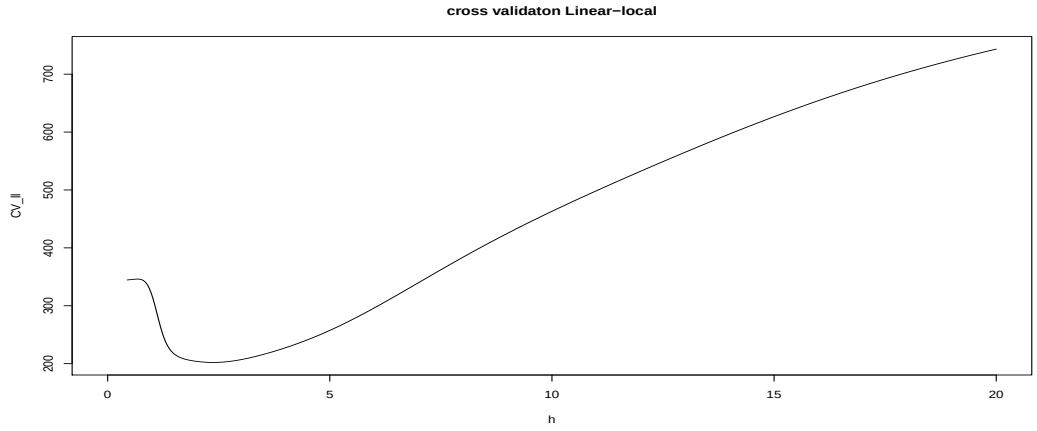

Figura 10: Función de validación cruzada para el estimador Local-lineal

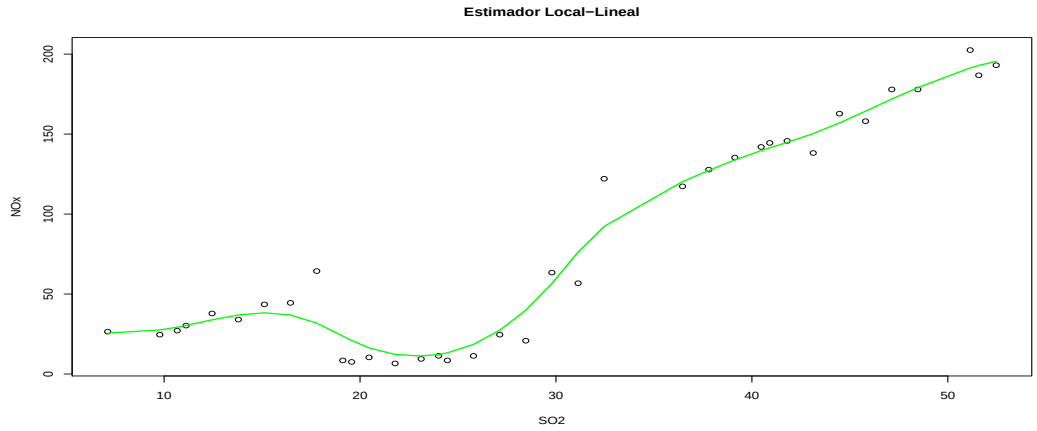

Figura 11: Estimación de la regresión dada por Local-lineal con ventana calculada mediante VC

3.3. Comparación de las estimaciones paramétricas y no-paramétricas de la regresión

En (Figura 12) se observa que las mejores estimaciones son las dadas por los estimadores Nadaraya-Watson y Local-lineal, ambos con el parámetro de suavizado escogido mediante validación cruzada.

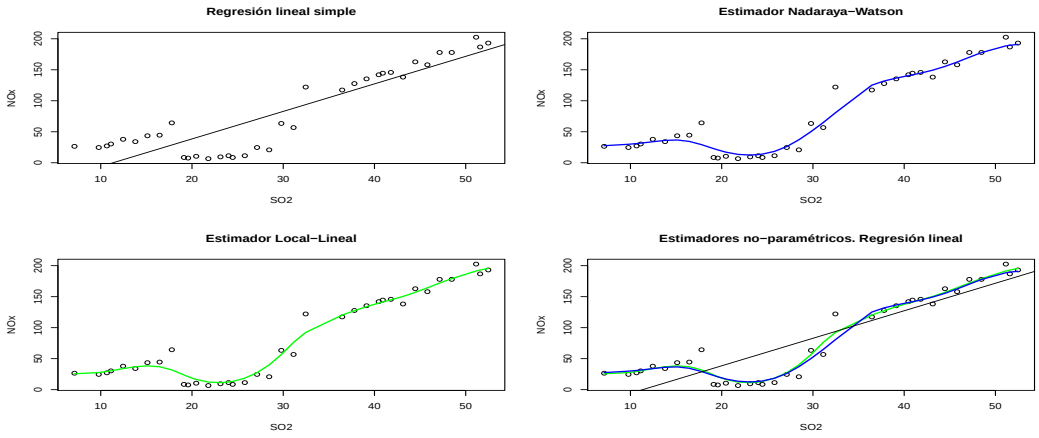

Figura 12: Estimaciones no-paramétricas: Nadaraya-Watson(azul), Local-lineal VC(verde). Estimación paramétrica: Regresión lineal simple(negro)

A la vista de las estimaciones proporcionadas por los resultados dados por los estimadores no paramétricos Nasaraya-Watson y Local-lineal observamos que hay dos datos atípicos (outliers) en la muestra. Por tanto estudiaremos como se comporta cada uno de los modelos estimados anteriormente. En primer lugar, el modelo regresión lineal simple (Figura 13) no presenta un cambio significativo al eliminar outliers. En cambio, los modelos dados por los estimadores Nadaraya-Watson (Figura **??**)y Local-lineal(Figura 15) se ajustan mejor a los datos. Esto puede ser debido a que los parámetros de suavizado calculados mediante validación cruzada son menores. En el caso de Nadaraya-Watson el parámetro de suavizado con la muestra anterior era de 2.7 y con la nueva muestra, es decir, sin outliers pasa a ser 1.2; con el estimador Lineal-local el paramétro de suavizado no se ve tan afectado, pues pasa a ser 2.14 cuando antes era de 2.38.

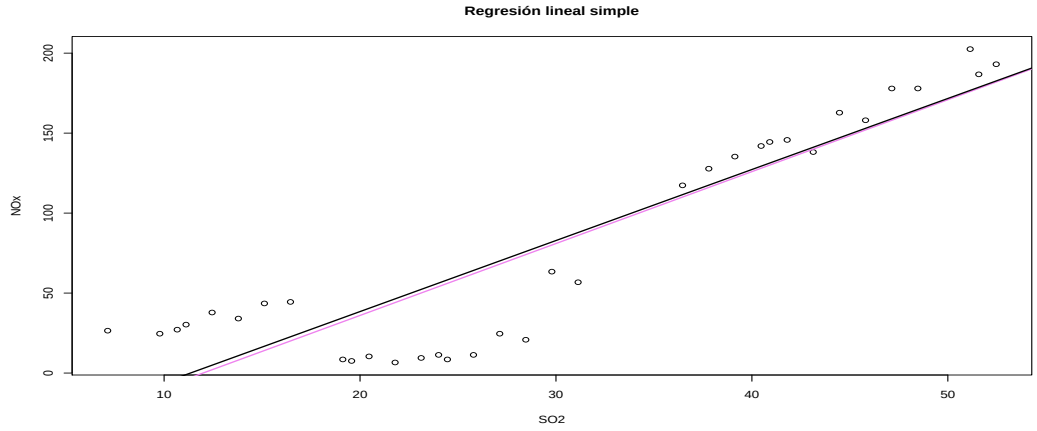

Figura 13: Estimación paramétrica: Modelo de regresión lineal simple (negro). Modelo de regresión lineal simple sin outliers (violeta)

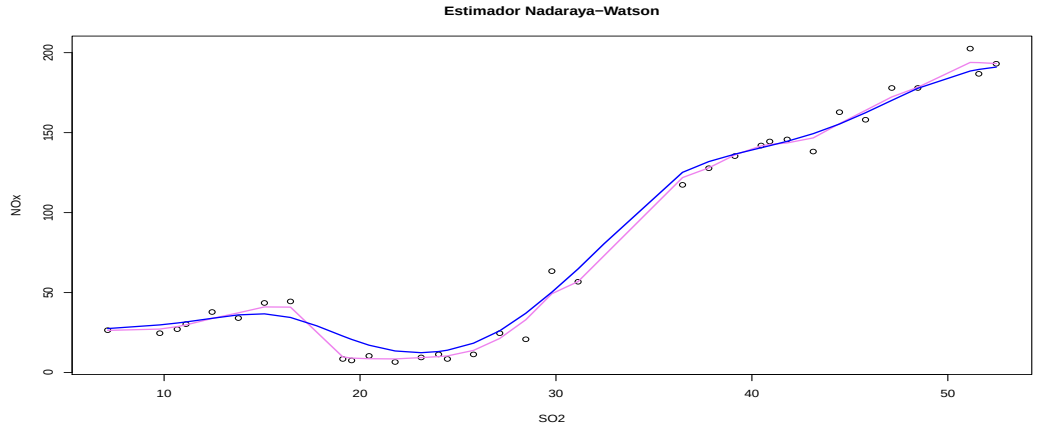

Figura 14: Estimación no-paramétrica: Estimación dada por Nadaraya$_W atson(azul) VS Estimacin dada por Nadaraya_W atson, sin outliers(violeta)$

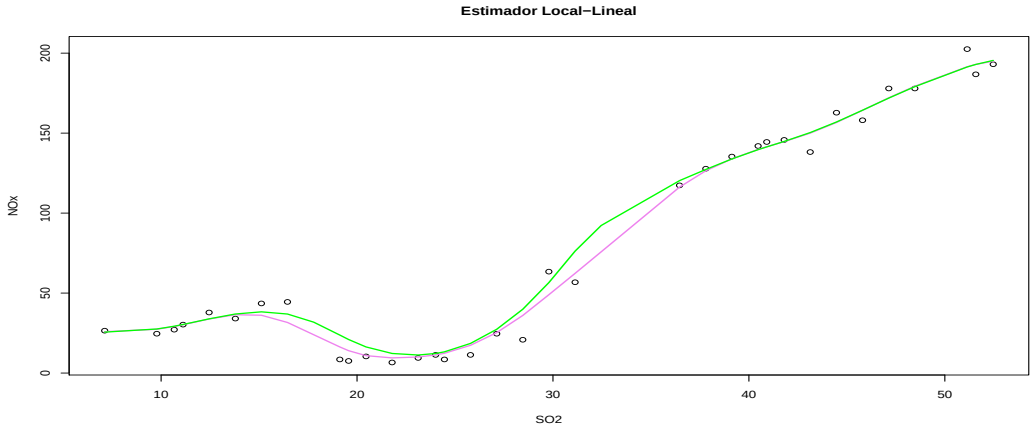

Figura 15: Estimación no-paramétrica: Estimación dada por Local-Lineal (verde) VS Estimación dada por Local-Lineal, sin outliers (violeta)

4. Interpretación de los modelos de regresión construídos como reglas de predicción

Hasta ahora, hemos trabajado con un problema de estimación dónde el objetivo es fijo, aunque es desconocido. Por tanto, la única fuente de variabilidad reside en los datos. A continuación, transformamos el problema de estimación anterior en un problema de predicción. En este nuevo problema existen dos fuentes de variabilidad los datos y el propio objetivo, pues este último es una variable aleatoria. por tanto, a la hora de estimar el error de una regla de predicción habrá que tener en cuenta las dos fuentes de variabilidad. A la hora de cuantificar el error de la regla de de predicción,$y = \eta(t, \vec{x}), \vec{x} = (t, x)$ se obtiene mediante los siguientes términos:

- **Función de pérdida** Cuantifica el error cometido al predecir NO_x mediante el modelo de regresión considerado, en general, el error cometido al predecir y_0 con $\eta(t, \bar{x})$ viene dado por

$$Q(y_0, \eta(t, \vec{x})) = (y_0 - \eta(t, \vec{x}))^2 \tag{27}$$

- **Error verdadero de la regla de predicción** Es el valor esperado de la función de pérdida respecto de NO_x con el modelo de regresión considerado, en general, es el valor esperado de la función de la función de pérdida respecto de (Y_0, T_0).

$$Err(\vec{x}; F) = E_F\left[(Y_0 - \eta(T_0, \vec{x}))^2\right] \tag{28}$$

- **Error aparente de la regla de predicción** Es la evaluación del error verdadero sobre los datos muestrales. Notemos que tiende a infraestimar el error cometido pues la regla de predicción ha sidoi reconstruída con dichos datos.

$$err(\vec{x}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \eta(t_i, \vec{x}))^2 \tag{29}$$

- **Optimismo de la regla de predicción** Es la diferencia entre el error verdadero y el aparente.

$$op(\vec{x}; F) = Err(\vec{x}, F) - err(\vec{x}) \tag{30}$$

- **Optimismo esperado de la regla de predicción** Es el valor esperado del optimismo respecto de los datos de SO_2. w(F)=E$_F\left[op(\vec{X}; F)\right]$ (30)

Notemos que en nuestro caso particular no conocemos F por tanto usaremos remuestreo boostrap uniforme para obtener una aproximación de las cantidades definidas anteriormente. En primer lugar obtenemos réplicas mediante el boostrap uniforme y con cada una de ellas obtenemos $\eta(t*, \vec{x}^{*b})$ por tanto la ecuación (27) se transfoma en $Q(Y*_0, \eta(t*_0, \vec{x}^{*b}))$. Luego, ya podemos obtener una aproximación para cada remuestra boostrap del error real (28) $Err(\vec{x}^{*b}; \hat{F}) = \frac{1}{n}\sum_{i=1}^{n} Q(y_i, \eta(t_i, \vec{x}^{*b}))$ del error aparente (29) $err(\vec{x}^{*b}) = \sum_{i=1}^{n} \frac{(x_j^{*b} == x_i)}{n} Q(y_i, \eta(t_i, \vec{x}^{*b}))$ y del optmismo de la regla de predicción (30) $op(\vec{x}^{*b}; \hat{F}) = \sum_{i=1}^{n} \frac{1}{n} - \frac{(x_j^{*b} == x_i)}{n} Q(y_i, \eta(t_i, \vec{x}^{*b}))$

4.1. Construcción de la regla de predicción mediante el modelo de regresión lineal simple

En primer lugar, definimos la regla de predicción como se había definido la recta de regresión lineal simple solo que en esta ocasión la variable $y = NO_x$ es desconocida, por lo tanto, nuestro objetivo es predecir los valores de NO_x a partir de los valores de SO_2. Estimamos el error real, el error aparente y el optimismo mediante boostrap(Tabla 3). Podemos decir que una buena estimación del error es el optimismo boostrap más el error aparente, 91.78288+884.4266=976.2095

$$y = -50{,}36 + 4{,}44SO_2 \tag{31}$$

A continuación representamos las rectas de regresión con cada una de las replicas Boostrap.

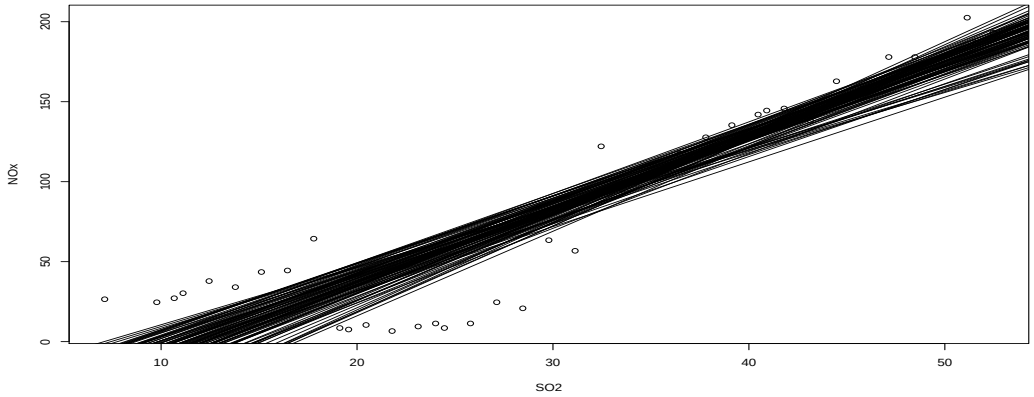

Figura 16: Función de validación cruzada para el estimador Local-lineal

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Media(B=100)	936.6984	844.9155	91.78288	884.4266

Cuadro 3: Valores del error real, error aparente y optimismo de la regla de predicción

Teniendo en cuenta que en la sección anterior habíamos detectado datos atípicos, vamos a eliminarlos y construír la regla de predicción sin ellos Al estimar el error real, el error aparente y el optimismo mediante boostrap(Tabla 4) observamos que no existe una diferencia significativa respecto al construir la regla de predicción con datos atípicos. Podemos decir que una buena estimación del error es el optimismo boostrap más el error aparente, 109.7548+872.7528=982.5076

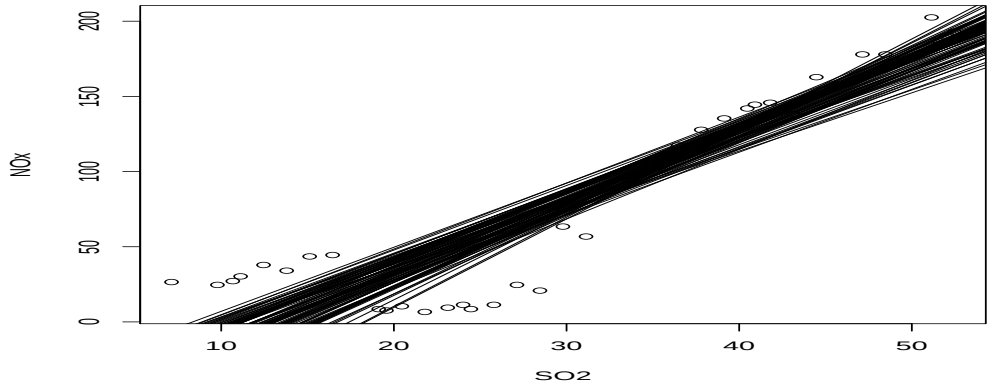

Figura 17: Estimación del modelo lineal simple con cada una de las réplicas boostrap

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Media(B=100)	937.5123	827.7575	109.7548	872.7528

Cuadro 4: Valores del error real, error aparente y optimismo de la regla de predicción

4.2. Construcción de la regla de predicción mediante el modelo de regresión dado por el estimador de Nadaraya-Watson

Ahora usamos la estimación del modelo de regresión dada por Nadaraya-Watson(Figura 9)y obtenemos los resultados (Cuadro 5).Podemos decir que una buena estimación del error es el optimismo boostrap más el error aparente, 111.0946 + 133.0673= 244.1619

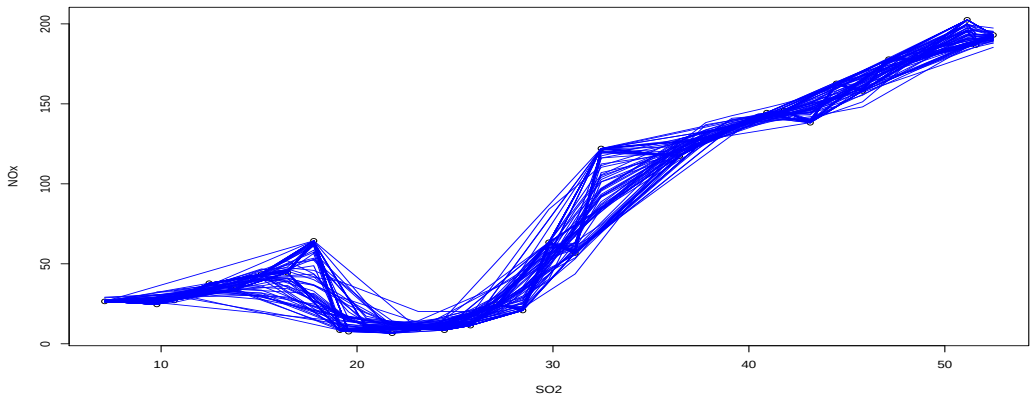

Figura 18: Construcción del estimador del estimador de Nadaraya-Watson con cada una de las réplicas Boostrap

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Media(B=100)	142.1942	31.09966	111.0946	133.0673

Cuadro 5: Valores del error real, error aparente y optimismo de la regla de predicción

A continuación, usamos la estimación del modelo de regresión dada por Nadaraya-Watson pero eliminando los datos atípicos (Figura 14)y obtenemos los resultados (Cuadro 6)que son notablemente mejores que los obtenidos teniendo en cuenta los datos atípicos. Además, una buena estimación del error sería 49.85111+22.93453 = 72.78564

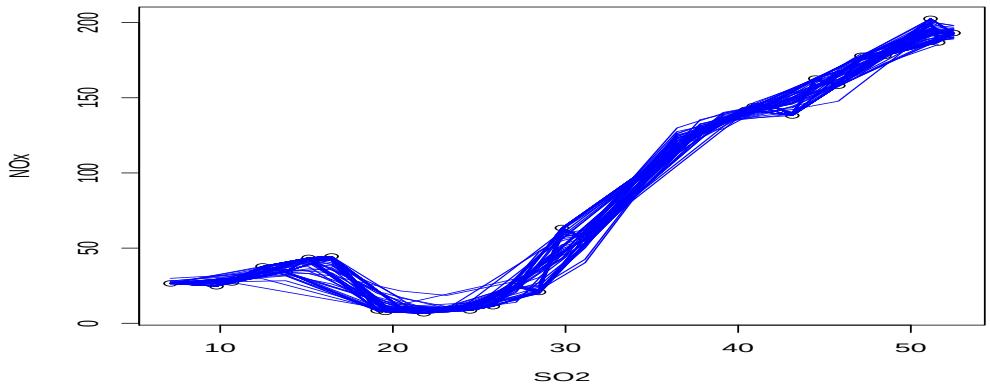

Figura 19: Construcción del estimador del estimador de Nadaraya-Watson con cada una de las réplicas Boostrap

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Media(B=100)	59.74708	9.895966	49.85111	22.93453

Cuadro 6: Valores del error real, error aparente y optimismo de la regla de predicción

4.3. Construcción de la regla de predicción mediante el modelo de regresión dado por el estimador Local-lineal

Finalmente usamos la estimación del modelo de regresión dada por Local-lineal(Figura **??**) y obtenemos los resultados (Cuadro 7)Podemos decir que una buena estimación del error es el optimismo boostrap más el error aparente, 114.4468 +107.8261 = 222.2729

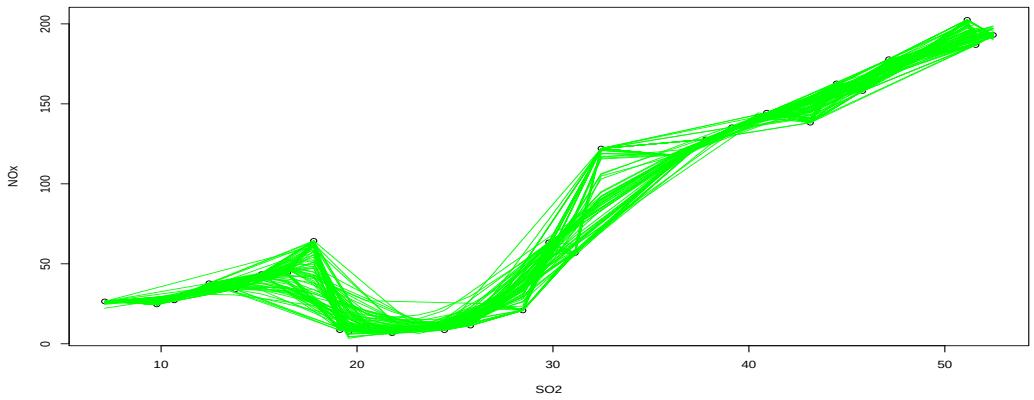

Figura 20: Construcción del estimador Local-lineal con cada una de las réplicas Boostrap

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Media(B=100)	145.8479	31.40112	114.4468	107.8261

Cuadro 7: Valores del error real, error aparente y optimismo de la regla de predicción

Finalmente usamos la estimación del modelo de regresión dada por Local-lineal(Figura 15) y obtenemos los resultados (Cuadro 8) que determinan que este estimador, al igual que Nadaraya-Watson mejora considerablemente al eliminar los datos atípicos pues en este caso, una buena estimación del error es 51.38684+38.86235= 90.24919

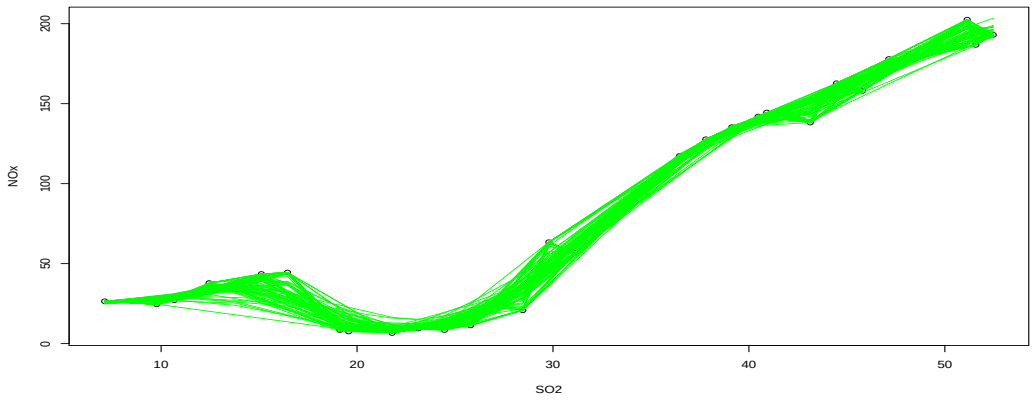

Figura 21: Construcción del estimador Local-lineal con cada una de las réplicas Boostrap (sin outliers)

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Media(B=100)	69.03245	31.40112	51.38684	38.86235

Cuadro 8: Valores del error real, error aparente y optimismo de la regla de predicción (sin outliers)

4.4. Comparación de las tres reglas de predicción con y sin outliers

Teniendo en cuenta los outlers, en (Tabla 9) podemos observar que las mejores reglas de predicción son las dadas por las estimaciones no-parámericas de los modelos de regresión. Aúnque nos decantaremos por la estimación dada por Nadaraya-Watson por tener el memor error real estimado mediante la suma del error aparente y el optimismo Boostrap (Tabla 10). Al elimirnar los outliers obtenemos que las reglas de predicción dadas por las estimaciones no paramétricas siguen siendo mejores (Tabla 11). Además tambien se observa como al eliminar los datos atípicos, éstas mejoran notablemnte los resutados anteriores, así como da una mejor estimación del error real (Tabla 12)

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Regresión lineal	936.6984	844.9155	91.78288	884.4266
Nadaraya-Watson	142.1942	31.09966	111.0946	133.0673
Local-lineal	145.8479	31.40112	114.4468	107.8261

Cuadro 9: Valores del error real, error aparente y optimismo de la regla de predicción

n=36	ESTIMACIÓN ERROR REAL
Regresión lineal	976.2095
Nadaraya-Watson	244.1619
Local-lineal	222.2729

Cuadro 10: Valores del error real, error aparente y optimismo de la regla de predicción

n=36	Error real Boostrap	error aparente Boostrap	optimismo Boostrap	error aparente
Regresión lineal	937.5123	827.7575	109.7548	872.7528
Nadaraya-Watson	59.74708	9.895966	49.85111	22.93453
Local-lineal	69.03245	31.40112	51.38684	38.86235

Cuadro 11: Valores del error real, error aparente y optimismo de la regla de predicción

n=36	ESTIMACIÓN ERROR REAL
Regresión lineal	982.5076
Nadaraya-Watson	72.78564
Local-lineal	90.24919

Cuadro 12: Valores del error real, error aparente y optimismo de la regla de predicción

5. Implementación en R

```
setwd("C:/Users/leyenda/Desktop/master/Remuestreo/Trabajo")
datos<- read.delim("C:/Users/leyenda/Desktop/master/Remuestreo/Trabajo/Datos_G2.txt",
dec=".", sep="\t",fill=TRUE,na.strings="NA",header=TRUE)
attach(datos)
names(datos)
summary(datos)
dim(datos)
ind1<-which(SO2==-1)
ind2<-which(NOx==-1)
SO2<-SO2[-ind1]
NOx<-NOx[-ind2]
windows()
par(mfrow=c(1,2))
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2",ylab="NOx", main="Gráfico de dispersión")
abline(lm(NOx[order(SO2)]~SO2[order(SO2)]))
summary(lm(NOx[order(SO2)]~SO2[order(SO2)]))
plot(NOx[order(NOx)],SO2[order(NOx)],xlab="NOx",ylab="SO2", main="Gráfico de dispersión")
abline(lm(SO2[order(NOx)]~NOx[order(NOx)]))
summary(lm(SO2[order(NOx)]~NOx[order(NOx)]))
##Diseñamos muestra de modo que a cada dato de SO2 le corresponda un único dato de NOx.
orden_unicos<-unique(SO2)
long=length(orden_unicos)
indice=numeric(long)
for(i in 1:long){
indice[i]<-min(which(SO2==orden_unicos[i]))}
NOx<-NOx[indice]
SO2<-SO2[indice]
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2",ylab="NOx", main="Gráfico de dispersión")
```

```
########################################
# Estimación de las funciones de densidad #
########################################
windows()
hist(SO2,freq=FALSE)
lines(density(SO2,kernel="rectangular"),col=4,xlim=c(0,300),ylim=c(0.0,0.5))
lines(density(SO2),lwd=2,xlim=c(0,300),ylim=c(0.0,0.5),col=3)
legend("topright",legend=c("Histograma","Histograma movil",
"Estimacion de Parzen-Rosemblat"),lwd=c(1,1,2),col=c(1,4,3))
shapiro.test(SO2)
windows()
hist(NOx,freq=FALSE)
lines(density(NOx,kernel="rectangular"),col=4,xlim=c(0,300),ylim=c(0.0,0.5))
lines(density(NOx),lwd=2,xlim=c(0,300),ylim=c(0.0,0.5),col=3)
legend("topright",legend=c("Histograma","Histograma movil",
"Estimacion de Parzen-Rosemblat"),lwd=c(1,1,2),col=c(1,4,3))
shapiro.test(NOx)

#####################################
# Modelos de regresión              #
#####################################

# Modelo de regresión lineal simple #
#####################################

## Cálculo de los coeficientes del modelo de regresion
regresion<-lm(NOx[order(SO2)]~SO2[order(SO2)])
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2",ylab="NOx")
abline(regresion)
#plot(regresion)
summary(regresion)
attach(regresion)
names(regresion)
```

```
# Estimación de la regresión  mediante Nadaraya-Watson #
########################################################

##########################################
#       Funciones auxilares              #
##########################################
K<-function(mas){
#--Núcleo gausiano
return(1/sqrt(2*pi)*exp(-mas^2/2))}
S<-function(t,mas,bandwidth,exponent){
#-- t vector de puntos dónde S será evaluada.
#-- mas valores de la variable independiente.
#-- bandwidth ventana.
#-- r exponent.
   return(diag(t(K(outer(mas,t,"-")/bandwidth))%*%(outer(mas,t,"-")^exponent)))}

##########################################################
#     Nadaraya-Watson estimator                          #
##########################################################
nw<-function(t,mas,y,bandwidth){
#-- Esta función devuelve la estimación calculada por el estimador no-paramétrico
#- de regresión Nayadara-Watson.
#-- t vector de puntos dónde evaluamos.
#-- mas valores de la variable independiente.
#-- y valores de la variable independiente.
#-- bandwidth es la ventana.
    return(apply(as.matrix(bandwidth),1,function(x)(K(outer(t,mas,"-")/x)/
    apply(K(outer(t,mas,"-")/x),1,sum))%*%y))}
```

```
#########################################################
#     Local-lineal estimator                            #
#########################################################
EstLL<-function(t,mas,y,bandwidth){
#-- Esta función devuelve la estimación calculada por el estimador
#- no-paramétrico de regresión Local-lineal.
#-- t vector de puntos dónde evaluamos.
#-- mas valores de la variable independiente.
#-- y valores de la variable independiente.
#-- bandwidth es la ventana.
return(apply(as.matrix(bandwidth),1,function(h){(t(K(outer(mas,t,"-")/h))*
(S(t,mas,h,2)-t(outer(mas,t,"-"))*S(t,mas,h,1))/apply(t(K(outer(mas,t,"-")/h))*
(S(t,mas,h,2)-t(outer(mas,t,"-"))*S(t,mas,h,1)),1,sum ))%*%y}))}

####################################
##                                ##
##   Validación cruzada           ##
##                                ##
####################################
cv_nw<-function(mas,y,bandwidth){
#-- Es la función de validación cruzada aplicada al estimador
#- no-paramétrico de la regresión Nadaraya-Watson.
    return(apply(as.matrix(bandwidth),1,function(x){1/length(mas) *
    sum( ((y - nw(mas,mas,y,x)) / (1 - (K(0)/
    apply( K(outer(mas,mas,"-")/x) ,1,sum) ) ) )^2 )}))}
cv_ll<-function(mas,y,bandwidth){
#--Es la función de validación cruzada aplicada al estimador
no-paramétrico de la regresión Local-lineal.
return(apply(as.matrix(bandwidth),1,function(h){1/length(mas)*
sum(((as.numeric(y -EstLL(mas,mas,y,h)))/(1-( K(0)*S(mas,mas,h,2)/
apply(t(K(outer(mas,mas,"-")/h))*(S(mas,mas,h,2)
-t(outer(mas,mas,"-"))*S(mas,mas,h,1)),1,sum))))^2)}))}

##- Secuencia de ventanas
h<-seq(0,20,by=0.02)
##--Se guardan los valores de la función de validación cruzada aplicada
a los estimadores no-paramétricos:Nadaraya-Watson y Local-lineal.
CV_nw<-numeric()
CV_nw<-cv_nw(SO2[order(SO2)],NOx[order(SO2)],h)
CV_ll<-numeric()
CV_ll<-cv_ll(SO2[order(SO2)],NOx[order(SO2)],h)
```

```
##-- Representación gráfica de la función de validación cruzada aplicada
#-a los estimadores no-paramétricos: Nadaraya-Watson y Local-lineal.
windows()
plot(CV_nw~h,type="l",main="cross validation Nadaraya-Watson")
windows()
plot(CV_ll~h,type="l",main="cross validaton Linear-local")


##-- Estimador de la regresión Nadaraya-Watson con la ventana de validación cruzada.
nadaraya_watson<-nw(SO2[order(SO2)],SO2[order(SO2)],NOx[order(SO2)],
h[which(CV_nw==min(CV_nw,na.rm=T))])
h[which(CV_nw==min(CV_nw,na.rm=T))] # 2.7
##-- Estimador de la regresión Local-lineal con la ventana de validación cruzada.
local_lineal<-EstLL(SO2[order(SO2)],SO2[order(SO2)],NOx[order(SO2)],
h[which(CV_ll==min(CV_ll,na.rm=T))])
h[which(CV_ll==min(CV_ll,na.rm=T))] # 2.38


#####################################################################
#           Representación gráfica                                  #
#####################################################################
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2", ylab="NOx",
main="Estimador Nadaraya-Watson")
lines(nadaraya_watson~SO2[order(SO2)],col="blue",lwd=2)
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2", ylab="NOx",
main="Estimador Local-Lineal")
lines(local_lineal~SO2[order(SO2)],col="green",lwd=2)
windows()
par(mfrow=c(2,2))
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2", ylab="NOx",
main="Regresión lineal simple")
abline(lm(NOx[order(SO2)]~SO2[order(SO2)]))
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2", ylab="NOx",
main="Estimador Nadaraya-Watson")
lines(nadaraya_watson~SO2[order(SO2)],col="blue",lwd=2)
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2", ylab="NOx",
main="Estimador Local-Lineal")
lines(local_lineal~SO2[order(SO2)],lwd=2,col="green")
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2", ylab="NOx",
```

```
main="Estimadores no-paramétricos. Regresión lineal")
lines(local_lineal~SO2[order(SO2)],lwd=2,col="green")
lines(nadaraya_watson~SO2[order(SO2)],col="blue",lwd=2)
abline(regresion)

##############################
#  ELIMINADO DATOS ATÍPICOS   #
##############################
# Modelo de regresión lineal simple

## Cálculo de los coeficientes del modelo de regresion
regresion_out<-lm(NOx[order(SO2)][-c(9,22)]~SO2[order(SO2)][-c(9,22)])
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2",ylab="NOx")
abline(regresion_out)
#plot(regresion)
summary(regresion_out)
attach(regresion_out)
names(regresion_out)

# Estimación de la regresión  mediante Nadaraya-Watson #
########################################################

##- Secuencia de ventanas
h<-seq(0,20,by=0.02)
##--Se guardan los valores de la función de validación cruzada aplicada
a los estimadores no-paramétricos:Nadaraya-Watson y Local-lineal.
CV_nw_out<-numeric()
CV_nw_out<-cv_nw(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],h)
CV_ll_out<-numeric()
CV_ll_out<-cv_ll(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],h)

##-- Representación gráfica de la función de validación cruzada aplicada a los
#- estimadores no-paramétricos: Nadaraya-Watson y Local-lineal.
windows()
plot(CV_nw_out~h,type="l",main="cross validation Nadaraya-Watson")
windows()
plot(CV_ll_out~h,type="l",main="cross validaton Linear-local")

##-- Estimador de la regresión Nadaraya-Watson con la ventana de validación cruzada.
nadaraya_watson_out<-nw(SO2[order(SO2)][-c(9,22)],SO2[order(SO2)][-c(9,22)],
NOx[order(SO2)][-c(9,22)],h[which(CV_nw_out==min(CV_nw_out,na.rm=T))])
```

```
h[which(CV_nw_out==min(CV_nw_out,na.rm=T))] # 1.22
##-- Estimador de la regresión Local-lineal con la ventana de validación cruzada.
local_lineal_out<-EstLL(SO2[order(SO2)][-c(9,22)],SO2[order(SO2)][-c(9,22)],
NOx[order(SO2)][-c(9,22)],h[which(CV_ll_out==min(CV_ll_out,na.rm=T))])
h[which(CV_ll_out==min(CV_ll_out,na.rm=T))] # 2.14


#####################################################################
#           Representación gráfica                                  #
#####################################################################
windows()
plot(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],xlab="SO2",
ylab="NOx",main="Estimador Nadaraya-Watson")
lines(nadaraya_watson_out~SO2[order(SO2)][-c(9,22)],col="violet",lwd=2)
lines(nadaraya_watson~SO2[order(SO2)],col="blue",lwd=2)
windows()
plot(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],xlab="SO2",
ylab="NOx",main="Estimador Local-Lineal")
lines(local_lineal_out~SO2[order(SO2)][-c(9,22)],col="violet",lwd=2)
lines(local_lineal~SO2[order(SO2)],lwd=2,col="green")
windows()
plot(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],xlab="SO2",
 ylab="NOx",main="Regresión lineal simple")
abline(regresion_out,col="violet",lwd=2)
abline(regresion,lwd=2)
```

```
#############################################################################
#WILD BOOTSTRAP. REGRESION LINEAL (Y=a+bx: Modelo de diseño fijo y heterocedástico). #
#Estimación de la densidad con teoría clásica, bootstraps uniforme y Wild bootstrap. #
#############################################################################
x<-SO2
y<-NOx
Varx<-var(x)*(length(x)-1)/length(x)
n<-length(SO2)
#TEORÍA CLÁSICA
a0<-mean(y)-mean(x)*(cov(x,y)/Varx)
b0<-cov(x,y)/Varx
windows()
plot(x,y)
abline(a=a0,b=b0)
u<-y-a0-b0*x
e<-u^2
Vare<-(n/(n-2))*mean(e)
#BOOTSTRAP UNIFORME Y WILD BOOTSTRAP
muestra<-numeric(n)
wmuestra<-numeric(n)
muestra=u-mean(u)
yboot<-numeric(n)
ywboot<-numeric(n)
B=1000
aboot<-numeric(B)
bboot<-numeric(B)
awboot<-numeric(B)
bwboot<-numeric(B)
for (k in 1:B){
l<-sample(1:n,replace=TRUE)
yboot=a0+b0*SO2+muestra[l]
aboot[k]<-mean(yboot)-mean(x)*(cov(x,yboot)/Varx)
bboot[k]<-cov(x,yboot)/Varx
for (i in 1:n){
r<-sample(c(u[i]*(1-sqrt(5))/2,u[i]*(1+sqrt(5))/2),replace=TRUE,
prob=c((5+sqrt(5))/10,1-(5+sqrt(5))/10))
wmuestra[i]=r[1]}
ywboot=a0+b0*x+wmuestra
awboot[k]<-mean(ywboot)-mean(x)*(cov(x,ywboot)/Varx)
bwboot[k]<-cov(x,ywboot)/Varx}
```

```
za<-seq(-100,20,length(n))
windows()
par(mfrow=c(1,2))
plot(za,dnorm(za,mean=a0,sd=sqrt((Vare/n)*(1+(mean(x)^2)/Varx))),col=2,xlab="",
ylab="",type="l",main="Densidad de ordenada en el origen")
lines(density(aboot),col=3, ylim=c(0,0.04))
lines(density(awboot),col=4)
zb<-seq(3,6,by=0.01)
plot(zb,dnorm(zb,mean=b0,sd=(sqrt(Vare/(n*Varx)))),col=2,xlab="",
ylab="", ylim=c(0,1.5),type="l",main="Densidad de la pendiente")
lines(density(bwboot),col=4)
lines(density(bboot),col=3)
```

```
#############################################################
# REGLA DE PREDICCIÓN: Modelo de regresión lineal simple #
#############################################################
B=100
Errbu<-numeric(B)
errbu<-numeric(B)
opbbu<-numeric(B)
t<-SO2
y<-NOx
mean((y-a0-b0*t)^2)
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2",ylab="NOx")

for(b in 1:B){
l<-sample(1:n,replace=TRUE)
tbu<-t[l]
ybu<-y[l]
cuantos<-numeric(n)
for(i in 1:n){
cuantos[i]<-length(which(tbu==t[i]& ybu==y[i]))}

Vartbu<-var(tbu)*(length(tbu)-1)/length(tbu)
a0bu<-mean(ybu)-mean(tbu)*(cov(tbu,ybu)/Vartbu)
b0bu<-cov(tbu,ybu)/Vartbu
abline(a=a0bu,b=b0bu)
ubu<-y-a0bu-b0bu*t
ebu<-ubu^2
Errbu[b]<-mean(ebu)
ubbu<-ybu-a0bu-b0bu*tbu
ebbu<-ubbu^2
errbu[b]<-mean(ebbu)
#errbu[b]=sum((cuantos/n)*ebbu)
opbbu[b]<-Errbu[b]-errbu[b]
}
op_Boostrap<-mean(opbbu)
err_Boostrap<-mean(errbu)
Err_Boostrap<-mean(Errbu)
op_Boostrap
err_Boostrap
Err_Boostrap
```

```
############################################################
# REGLA DE PREDICCIÓN: Modelo de regresión Nadaraya-Watson #
############################################################
B=100
Errbu<-numeric(B)
errbu<-numeric(B)
opbbu<-numeric(B)
t<-SO2
y<-NOx
n=length(t)
mean((y[order(t)]-as.numeric(nadaraya_watson))^2)
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2",ylab="NOx")
for(b in 1:B){
l<-sample(1:n,replace=TRUE)
tbu<-t[l]
ybu<-y[l]
CV_nwbu<-cv_nw(tbu[order(tbu)],ybu[order(tbu)],h)
est_bu0<-nw(t[order(t)],tbu[order(t)],ybu[order(t)],h[which(CV_nwbu==min(CV_nwbu,na.rm=T)
est_bu<-nw(tbu[order(tbu)],tbu[order(tbu)],ybu[order(tbu)],h[which(CV_nwbu==min(CV_nwbu,n
lines(est_bu~tbu[order(tbu)],col="blue")
ubu<-y[order(t)]-as.numeric(est_bu0)
ebu<-ubu^2
Errbu[b]<-mean(ebu)
ubbu<-ybu[order(tbu)]-as.numeric(est_bu)
ebbu<-ubbu^2
errbu[b]<-mean(ebbu)
opbbu[b]<-Errbu[b]-errbu[b]
}
op_Boostrap_NAD<-mean(opbbu)
err_Boostrap_NAD<-mean(errbu)
Err_Boostrap_NAD<-mean(Errbu)
op_Boostrap_NAD
err_Boostrap_NAD
Err_Boostrap_NAD
```

```
#############################################################
# REGLA DE PREDICCIÓN: Modelo de regresión Local-lineal    #
#############################################################
B=100
Errbu<-numeric(B)
errbu<-numeric(B)
opbbu<-numeric(B)
t<-SO2
y<-NOx
n=length(SO2)
mean((y[order(t)]-as.numeric(local_lineal))^2)
windows()
plot(SO2[order(SO2)],NOx[order(SO2)],xlab="SO2",ylab="NOx")
for(b in 1:B){
l<-sample(1:n,replace=TRUE)
tbu<-t[l]
ybu<-y[l]
CV_llbu<-cv_ll(tbu[order(tbu)],ybu[order(tbu)],h)
est_ll_bu0<-EstLL(t[order(t)],tbu[order(t)],ybu[order(t)],h[which(CV_llbu==min(CV_llbu,na
est_ll_bu<-EstLL(tbu[order(tbu)],tbu[order(tbu)],ybu[order(tbu)],h[which(CV_llbu==min(CV_
lines(est_ll_bu~tbu[order(tbu)],col="green")
ubu<-y[order(t)]-as.numeric(est_ll_bu0)
ebu<-ubu^2
Errbu[b]<-mean(ebu)
ubbu<-ybu[order(tbu)]-as.numeric(est_ll_bu)
ebbu<-ubbu^2
errbu[b]<-mean(ebbu)
opbbu[b]<-Errbu[b]-errbu[b]
}
op_Boostrap_LL<-mean(opbbu)
err_Boostrap_LL<-mean(errbu)
Err_Boostrap_LL<-mean(Errbu)
op_Boostrap_LL
err_Boostrap_LL
Err_Boostrap_LL
```

```
############################################################
# REGLA DE PREDICCIÓN: Modelo de regresión lineal simple #
############################################################
B=100
Errbu<-numeric(B)
errbu<-numeric(B)
opbbu<-numeric(B)
outliers<-c(which(SO2==SO2[order(SO2)][9]),which(SO2==SO2[order(SO2)][22]))
t<-SO2[-outliers]
y<-NOx[-outliers]
n=length(t)
a0_out=regresion_out$coef[1]
b0_out=regresion_out$coef[2]
mean((y-a0-b0*t)^2)
windows()
plot(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],xlab="SO2",ylab="NOx")
for(b in 1:B){
l<-sample(1:n,replace=TRUE)
tbu<-t[l]
ybu<-y[l]
cuantos<-numeric(n)
for(i in 1:n){
cuantos[i]<-length(which(tbu==t[i]& ybu==y[i]))}
Vartbu<-var(tbu)*(length(tbu)-1)/length(tbu)
a0bu<-mean(ybu)-mean(tbu)*(cov(tbu,ybu)/Vartbu)
b0bu<-cov(tbu,ybu)/Vartbu
abline(a=a0bu,b=b0bu)
ubu<-y-a0bu-b0bu*t
ebu<-ubu^2
Errbu[b]<-mean(ebu)
ubbu<-ybu-a0bu-b0bu*tbu
ebbu<-ubbu^2
errbu[b]<-mean(ebbu)
#errbu[b]=sum((cuantos/n)*ebbu)
opbbu[b]<-Errbu[b]-errbu[b]}
op_Boostrap_out<-mean(opbbu)
err_Boostrap_out<-mean(errbu)
Err_Boostrap_out<-mean(Errbu)
op_Boostrap_out
err_Boostrap_out
Err_Boostrap_out
```

```
############################################################
# REGLA DE PREDICCIÓN: Modelo de regresión Nadaraya-Watson #
############################################################
B=100
Errbu<-numeric(B)
errbu<-numeric(B)
opbbu<-numeric(B)
outliers<-c(which(SO2==SO2[order(SO2)][9]),which(SO2==SO2[order(SO2)][22]))
t<-SO2[-outliers]
y<-NOx[-outliers]
n=length(t)
mean((y[order(t)]-as.numeric(nadaraya_watson_out))^2)
windows()
plot(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],xlab="SO2",ylab="NOx")
for(b in 1:B){
l<-sample(1:n,replace=TRUE)
tbu<-t[l]
ybu<-y[l]
CV_nwbu<-cv_nw(tbu[order(tbu)],ybu[order(tbu)],h)
est_bu0<-nw(t[order(t)],tbu[order(t)],ybu[order(t)],h[which(CV_nwbu==min(CV_nwbu,na.rm=T)
est_bu<-nw(tbu[order(tbu)],tbu[order(tbu)],ybu[order(tbu)],h[which(CV_nwbu==min(CV_nwbu,n
lines(est_bu~tbu[order(tbu)],col="blue")
ubu<-y[order(t)]-as.numeric(est_bu0)
ebu<-ubu^2
Errbu[b]<-mean(ebu)
ubbu<-ybu[order(tbu)]-as.numeric(est_bu)
ebbu<-ubbu^2
errbu[b]<-mean(ebbu)
opbbu[b]<-Errbu[b]-errbu[b]}
op_Boostrap_NAD_out<-mean(opbbu)
err_Boostrap_NAD_out<-mean(errbu)
Err_Boostrap_NAD_out<-mean(Errbu)
op_Boostrap_NAD_out
err_Boostrap_NAD_out
Err_Boostrap_NAD_out
```

```
#############################################################
# REGLA DE PREDICCIÓN: Modelo de regresión Local-lineal  #
#############################################################
B=100
Errbu<-numeric(B)
errbu<-numeric(B)
opbbu<-numeric(B)
outliers<-c(which(SO2==SO2[order(SO2)][9]),which(SO2==SO2[order(SO2)][22]))
t<-SO2[-outliers]
y<-NOx[-outliers]
n=length(t)
mean((y[order(t)]-as.numeric(local_lineal_out))^2)
windows()
plot(SO2[order(SO2)][-c(9,22)],NOx[order(SO2)][-c(9,22)],xlab="SO2",ylab="NOx")
for(b in 1:B){
l<-sample(1:n,replace=TRUE)
tbu<-t[l]
ybu<-y[l]
CV_llbu<-cv_ll(tbu[order(tbu)],ybu[order(tbu)],h)
est_ll_bu0<-EstLL(t[order(t)],tbu[order(t)],ybu[order(t)],h[which(CV_llbu==min(CV_llbu,na
est_ll_bu<-EstLL(tbu[order(tbu)],tbu[order(tbu)],ybu[order(tbu)],h[which(CV_llbu==min(CV_
lines(est_ll_bu~tbu[order(tbu)],col="green")
ubu<-y[order(t)]-as.numeric(est_ll_bu0)
ebu<-ubu^2
Errbu[b]<-mean(ebu)
ubbu<-ybu[order(tbu)]-as.numeric(est_ll_bu)
ebbu<-ubbu^2
errbu[b]<-mean(ebbu)
opbbu[b]<-Errbu[b]-errbu[b]
}
op_Boostrap_LL_out<-mean(opbbu)
err_Boostrap_LL_out<-mean(errbu)
Err_Boostrap_LL_out<-mean(Errbu)
op_Boostrap_LL_out
err_Boostrap_LL_out
Err_Boostrap_LL_out
```