DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP3 de Métodos Numéricos

4 de mayo de 2024 Métodos Numéricos

Grupo: 21

Estudiante	LU	Correo electrónico
Lucas Mas Roca	122/20	lmasroca@gmail.com
Juan Ignacio Ponce	420/21	juaniponce0@gmail.com
Juan Pablo Anachure	099/16	janachure@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Introducción

En este trabajo práctico, usaremos lo visto en la materia para implementar el procedimiento de Regresión Localmente Pesada, o loess, visto en los papers de Cleveland de 1979 y 1988, nos centraremos más en este último. Utilizaremos la versión multivariada de este procedimiento, suavizando una variable dependiente en función de varias variables independientes. Este proceso se aplica de forma local, con lo cual solo los puntos cercanos a un punto x influyen en el proceso de suavizar ese punto.

Para aplicar este método tendremos varias suposiciones, las cuales debemos verificar que se cumplan (y en caso de que sea necesario tomar medidas correctivas si alguna de esas suposiciones no se cumple). El objetivo de este trabajo práctico es implementar este método en Python y además recrear y comprender la experimentación realizada en el paper de Cleveland de 1988. Intentaremos realizar un procedimiento parecido y conseguir gráficos similares a los del paper a partir de nuestra implementación del método utilizando el dataset provisto por la cátedra. Adicionalmente, experimentaremos cambiando parámetros como el tamaño del vecindario y el grado, además experimentaremos usando datos sintéticos.

2. Desarrollo

Para la implementación del procedimiento loess utilizamos Python, importando paquetes como NumPy, Pandas y SciPy, entre otros. Primero debemos importar la base de datos, debemos separar la variable dependiente de las variables independientes, en este caso la variable dependiente será el ozono, el resto (radiación solar, viento y temperatura) serán las variables independientes. Llamaremos y al vector de $\mathbb{R}^n$ que contiene las observaciones (las y_i) de la variable dependiente, mientras que X será la matriz que tiene como filas a los vectores x_1, x_2 y x_3 cada uno de estos vectores de $\mathbb{R}^n$ contiene las observaciones (las x_{j_i}) de las respectivas variables independientes (en este caso son 3 para los datos del paper).

Ahora debemos estandarizar X (estandarizamos cada una de sus filas) lo haremos de la misma forma que se hace en el paper, simplemente dividiendo cada variable independiente por su varianza (más sobre esto en la sección de comentarios adicionales). Definimos $p(x, y)$ la distancia euclídea entre x e y dos vectores de $\mathbb{R}^m$ para un m genérico, mientras que $W(u)$ es la función tricúbica de la misma forma que en el paper.

Adicionalmente, definimos $X_V(X, x_k, q)$ como el vecindario de tamaño q de x_k y adicionalmente definimos $d_k = d(X_V(X, x_k, q), x_k)$ como la distancia entre $x_k = (x_{1_k}, x_{2_k}, x_{3_k})$ y el elemento más lejano dentro del vecindario de tamaño q de este elemento, para resolver esto debemos conseguir el vecindario de x_k usando la función $neighbors$ (una función que toma X, x_k y q y devuelve la matriz X_V que contiene los q elementos de X más cercanos, en distancia euclídea, a x_k, en otras palabras, su vecindario) y luego usar la función d (una función que toma una matriz que contiene el vecindario de x_k y el vector x_k y devuelve la distancia entre x_k y el elemento con la mayor distancia a x_k dentro del vecindario) para conseguir la mayor distancia dentro de su vecindario.

Utilizaremos todo esto para armar una matriz de pesos W (vamos a tener que armar n matrices W, una por cada x_k las llamaremos W_k), definimos la matriz W_k como una matriz diagonal con los elementos de su diagonal tales que

$$W_{k_{ii}} = W(\frac{p(x_k, x_i)}{d(neighbors(X, x_k, q), x_k)})$$

Además definimos la matriz A tal que tenga como columnas los polinomial features de X, en otras palabras si queremos aplicar el procedimiento loess linear la matriz A tendrá como columnas los vectores x_1, x_2, x_3 y un vector de unos (estos son los polinomial features de grado 1), mientras que

si queremos aplicar el método cuadrático A tendrá las mismas columnas que la matriz A del método lineal, pero se agregan las columnas que contienen los cuadrados y productos cruzados de los x_j (estos serán los polinomial features de grado 2). Podemos ver como se extiende esto para un grado genérico, siendo A una matriz de $\mathbb{R}^{nx(cantX_i^{grado}+1)}$ con $cantX_i$ la cantidad de variables independientes y $grado$ el grado de fitting que usaremos (por ejemplo si usamos fitting lineal $grado = 1$, para cuadrático $grado = 2$).

Finalmente, debemos llamar a la función $armarA(X, grado)$ luego de estandarizar X e y, tendremos ahora que usar $armarW(X, x_k, q)$ para armar cada una de las W_k. Una vez que conseguimos las W_k debemos resolver las siguientes n ecuaciones normales pesadas:

$$A^t W_k A\beta = A^t W_k y$$

Pero puede pasar que este sistema tenga infinitas soluciones (por ejemplo, si $q < cantX_i^{grado} + 1$) con lo cual no vamos a poder usar `np.linalg.solve`$(A^t$ @ W_k @ A, A^t @ W_k @ $y)$ ya que $A^t W_k A$ no será invertible, pero podemos usar `np.linalg.pinv`$(A^t$ @ W_k @ $A)$ para conseguir la pseudoinversa de esta matriz utilizando SVD, luego multiplicamos esta pseudoinversa por A^t @ W_k @ y para finalmente conseguir un vector incógnita β el cual será un vector de $\mathbb{R}^{cantX_i^{grado}+1}$ que contendrá los coeficientes que multiplican a cada uno de los polinomial features para ese x_k.

Una vez que tenemos los coeficientes debemos conseguir los valores fitteados de y, llamemos $\widehat{y}_k$ al valor fitteado de la k-ésima observación de la variable y. Tenemos entonces $\widehat{y} = (\beta_k^t$ @ $A^t)[k]$ para cada k, si recorremos todos los valores k y armamos un vector con los $\widehat{y}_k$ conseguimos $\widehat{y}$ cómo resultado final. Podemos ver como tendremos que resolver n ecuaciones normales (uno por cada x_k), en este caso serian 111 veces.

Una alternativa a esto es armar una función para predecir y luego usarla para predecir cada uno de nuestros X, la función *predict* ($\widehat{g}(x)$ en el paper) es similar a hacer fit, pero de un solo elemento (en otras palabras buscamos hacer fit pero solo para un x_k). Debemos estandarizar X y el elemento x que buscamos predecir (x puede estar o no estar en X, veremos más sobre esto más adelante), para estandarizar este último debemos dividir cada una de sus componentes por la varianza de cada variable independiente de X. Luego, debemos armar la matriz A, de la misma forma que antes, y armar una matriz W (en este caso, como x podría no estar en X, es posible que ninguno de los pesos de los x_i en X sea 1, pero esto no importa) que defina el vecindario de x dentro de X.

A continuación, calculamos β de la misma forma que lo hacíamos antes, pero en este caso solo resolvemos una vez cuadrados mínimos, teniendo ahora el β asociado a x, luego debemos conseguir $\overline{A}^t$ que es conseguir los polinomial features de x, de forma similar a como hicimos antes (dependiendo del caso es posible que sea necesario aplicar un reshape a x para que se pueda usar la función de polinomial features de sklearn, por ejemplo $x.reshape(1, -1)$). Finalmente, usamos $(\beta^t$ @ $\overline{A}^t)[0]$ para conseguir $\widehat{g}(x)$, la predicción de ozono para los valores x de las variables independientes (esta función además toma parámetros como $grado$ y q, además de recibir el X e y para predecir), este último [0] se debe a que el producto matricial resulta en una matriz de 1 dimensión, con lo cual podemos acceder al único elemento para devolver este número.

Usando esta función podemos recorrer todos los x_i en X y predecir cada uno, el resultado de poner todas esas predicciones en un vector será $\widehat{y}$, ya que $\widehat{g}(x_i) = \widehat{y}_i$ para todo x_i en X $(i = 1, ..., n)$, por lo visto en las secciones 2 y 4 del paper de 1988. Nuevamente, podemos ver que vamos a tener que llamar a esta función n veces, con lo cual seguimos teniendo que resolver cuadrados mínimos n veces.

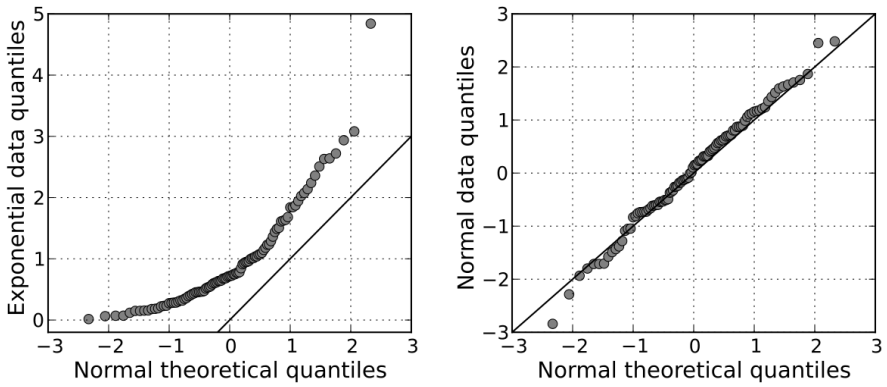
Esta función *predict* es la que utilizaremos para dibujar curvas en los gráficos, ya que la usaremos para predecir los valores de un *np.linspace*, generando de esta forma un gráfico.

2.1. Herramientas de Visualización

Existen 3 herramientas a tomar en cuenta:

- **QQ-Plot**: Un gráfico Q-Q ("Q"viene de cuantil) es un método que permite comparar nuestros datos observados con una distribución estadística, y poder darle un sentido a nuestros datos desde un punto de vista estadístico. De esta manera podremos ver si nuestros datos son totalmente aleatorios o si siguen alguna distribución teórica conocida. Además, puede usarse la misma idea para comparar las distribuciones inferidas directamente de dos conjuntos de observaciones, donde los tamaños de las muestras sean distintos.

  En nuestro caso, nos interesa comparar nuestros datos con la distribución gaussiana (distribución normal). Se ordenan los datos y se grafica el i-ésimo dato contra el correspondiente cuantil gaussiano, este nos ayuda a separar los datos de una distribución (en este caso normal) en grupos iguales. Los grupos los haremos de un elemento y el cálculo de estos cuantiles depende de si conocemos o no la distribución de nuestros datos, que en nuestro caso lo sabemos, ya que operaremos con media 0 y varianza 1 por lo que se pueden calcular desde un punto de vista paramétrico, entonces comparamos los percentiles empíricos de un conjunto de datos, con los percentiles teóricos de una Normal que se encuentran resumidas en una tabla, y por cada valor de los datos de la distribución obtenemos un valor z para los datos de la muestra.

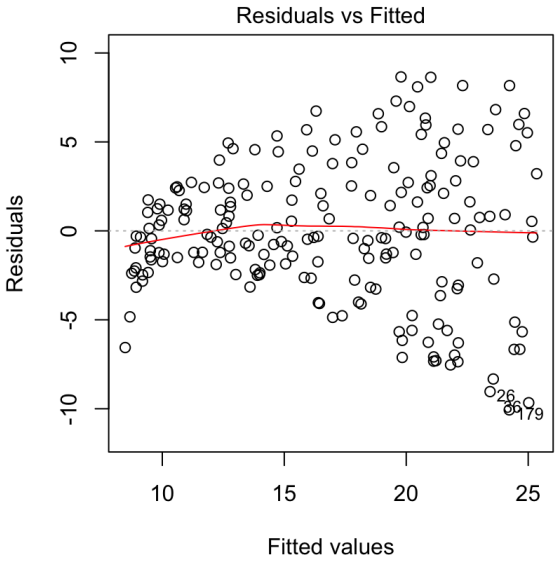


En la primera imagen tenemos un QQ-Plot que compara una distribución exponencial con parámetro $\lambda = 1$ contra datos generados por una distribución normal de media 0 y varianza 1, mientras que en el segundo gráfico podemos ver una comparación entre dos distribuciones normales. Acá se puede ver claramente que los datos sacados de una distribución exponencial no se parecen en nada a una distribución normal.

- **Diagnostic Plot Absolut Residuals vs Fitted:** Es un método gráfico, comúnmente orientado a los problemas de análisis de regresiones, útil para ilustrar si los residuos de las regresiones mantienen un patrón de distribución normal o no. Además, podemos medir si la varianza de los errores de la función (ε_i) depende del nivel g. Como alternativa, también es posible tomar el absoluto de los residuos como utilizaremos en secciones siguientes.

  Se coloca sobre el eje x los valores predichos, mientras que en el eje y la diferencia entre lo predicho y las observaciones de la variable dependiente, se puede observar qué tan grandes son los residuos y si estos tienen una distribución normal.

  Podemos ver en el siguiente gráfico cómo la dispersión de los residuos aumenta en valores fitteados más altos, indicando que la varianza de los errores tiene una dependencia del nivel de la variable dependiente.



- **Component Residual Plot:** Este gráfico nos permite estudiar la relación entre los residuos y cada una de las variables independientes del problema. Para cada uno de estos gráficos tomamos en el eje x cada una de las variables independientes que se tiene en el dataset, mientras que en el eje y se muestran los residuos del fit.

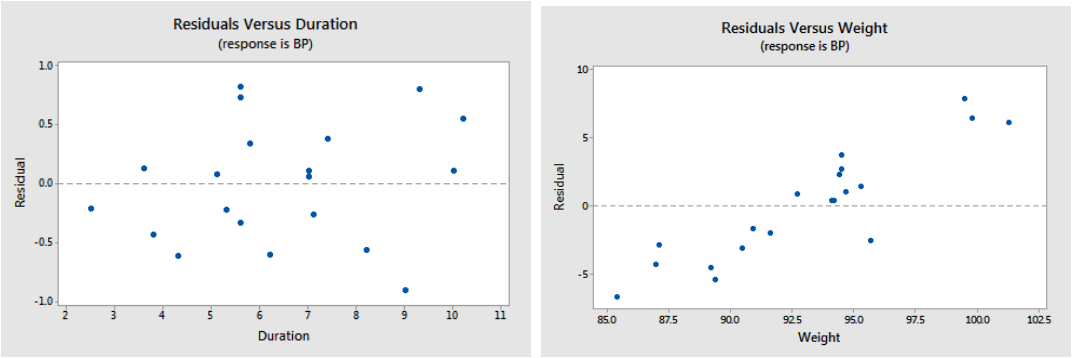


Figura 1: Component residual plot

Podemos ver en el gráfico de la derecha que hay una fuerte relación entre el peso y el residuo de fittear la presión sanguínea, mientras que en el gráfico de la izquierda no se ve una relación tan clara entre la duración de la hipertensión y el residuo de fittear la presión sanguínea.

2.2. Gráficos de diagnóstico

En esta subsección intentaremos recrear los gráficos de la sección 5 del paper de Cleveland de 1988. Entre estos se encuentran Pairplot, QQ-Plot, Residual vs Fitted, entre otros. Los datos que se estudian durante el trabajo están relacionados con variables meteorológicas, y buscamos realizar la regresión sobre los valores de ozono, por ello, antes de realizar la experimentación queremos estudiar los datos de entrada.

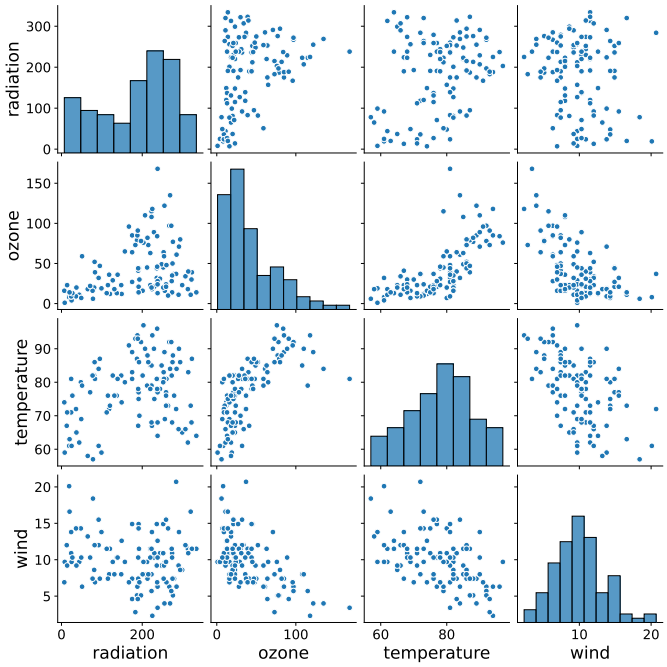


Figura 2: Pairplot de las variables independientes y el ozono

Como se observa en la figura 2, particularmente en la fila 2, se encuentra la relación del parámetro ozono contra el resto de las variables, notemos que a cuanto más crece la temperatura mayor crece las observaciones de ozono, algo inverso ocurre con la velocidad del viento, ya que cuando aumenta se reduce la cantidad de ozono, mientras que cuando la radiación aumenta no hay un claro aumento en la cantidad de ozono, es decir, a mayor radiación no se puede decir que hay un aumento en el ozono. Adicionalmente, podemos ver en el gráfico que no hay una correlación clara de las variables independientes entre sí, lo cual es deseable para este método.

Para el Pairplot solo necesitamos comparar los datos de las distintas variables entre sí. Esto se realiza de una manera muy sencilla con la herramienta de *seaborn* `pairplot`, el cual toma el dataframe y realiza un plot por cada combinación posible entre las distintas columnas del dataframe. Luego, en la diagonal se muestra la distribución marginal de los datos de cada columna.

Para el QQ-Plot (explicado en la subsección 2.1) calculamos primero los cuantiles de los datos que tenemos como residuo y luego hacemos lo mismo con una distribución normal de esperanza 0 y varianza 1, los cuales deben tener el mismo tamaño que los datos de la columna de residuo. Hay que asegurarse de que ambos cuantiles calculados estén en orden para luego poder graficarlo con un `scatterplot`. Luego, podemos ver qué tan parecido son nuestros datos a una distribución normal a partir de qué tan pegados a la línea diagonal imaginaria están. En este caso, podemos ver que los datos se asemejan a una distribución normal porque se acercan a la línea recta marcada en rojo.

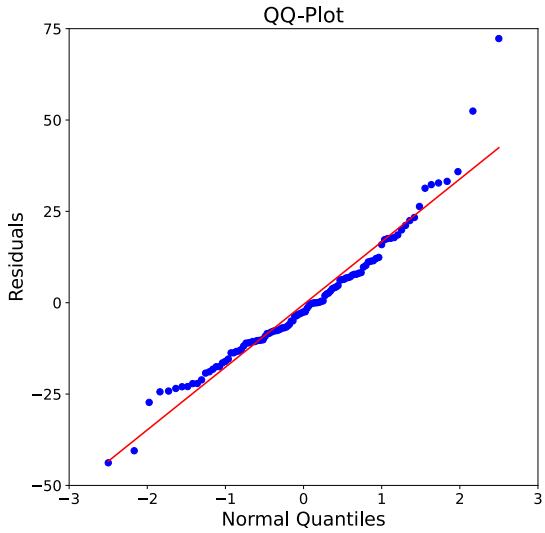


Figura 3: Pairplot de las variables independientes y el ozono

En el gráfico 3 se muestra como los errores están normalmente distribuidos, esto es deseable que ocurra, porque una de las asunciones que tiene el método de loess, es que los errores se mantengan normalmente distribuidos, además de que la varianza sea 1.

Para el segundo gráfico de la misma figura en el paper (Figura 4), no fue más que hacer otro scatterplot de los valores fiteados linealmente contra los residuos en su valor absoluto, esto quiere decir que fiteamos los valores del ozono con una cantidad de vecinos del 40 % (q=44) de la cantidad de puntos totales. Además, graficamos el loess del plot prediciendo el residuo absoluto para una serie de puntos en el eje x dentro del gráfico a partir de los datos fiteados con $f = 2/3$ (q=74), o sea que ahora la proporción de datos en el vecindario es de dos tercios la población total. Para esta última parte, utilizamos el eje x del gráfico, los valores fitteados, como X y el eje y del gráfico, sus residuos absolutos, como y y usamos eso con $q = 74$ para predecir todos los elementos de un linspace con el rango del eje x, luego graficamos ese linspace contra el resultado de la predicción y obtenemos la curva.

Para los plots de la Figura 5 que compara los residuos con las variables independientes, hicimos también un scatterplot como en el gráfico anterior, solo que no usamos los absolutos y dibujamos un scatter para cada una de las variables independientes por separado. El loess lo calculamos de la misma manera, calculamos la función $\widehat{g}(x)$ para una serie de puntos para las variables independientes.

Podemos ver que el primer gráfico parece tener una curva bastante lineal y centrada en el 0, lo cual es algo deseable en este método, ya que indica que los residuos parecen ser normales, al menos en relación a la radiación. Mientras tanto, podemos ver que en el segundo gráfico la curva ya no es tan lineal y esto es aún peor en el tercero, cuya curva parece más una parábola que una línea, por este motivo en el paper indica que estos parámetros no pasan los diagnósticos para estos datos. Por este motivo en el paper se decide cambiar de parámetros y repetir los experimentos.

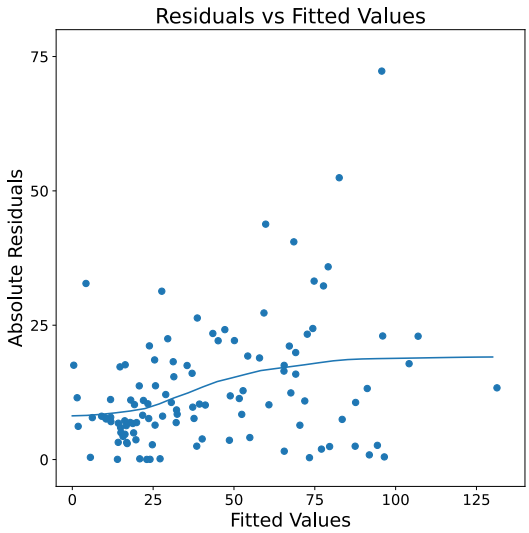


Figura 4: Gráfico de los valores absolutos de los residuos contra los valores 'fiteados'

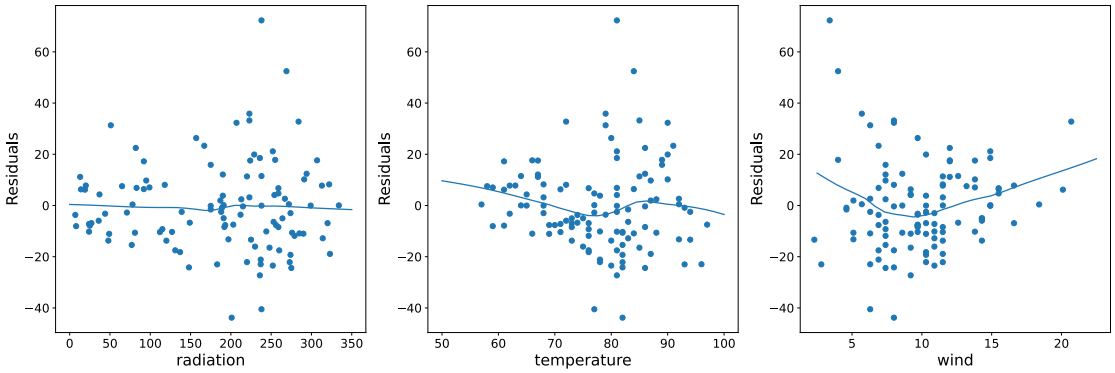


Figura 5: Gráficos de los residuos contra las variables independientes

Para las últimas tres figuras de la sección 5 del paper se realiza un three-variable conditioning plot, en el cual se fijan dos variables independientes y se compara la raíz cúbica del ozono con una variable independiente a la vez. Para esto, tuvimos que calcular el loess, pero cuadrático con una proporción de vecinos de $f = 0.8$ ($q = 89$), debemos hacer predict de x usando X y $\sqrt[3]{y}$ siendo x tal que dos de sus componentes están fijos y el componente restante es un linspace, debemos recorrer todo este linspace y predecirlo, para luego graficar el linspace contra su predicción para obtener una curva. Primero con temperatura, luego con radiación y por último viento (Figura 6 respectivamente). Con el objetivo de estudiar que tan consistente son los datos, uno podría evaluar el sesgo que tiene la predicción de

ozono para cada variable independiente en sí, como se puede ver en la primera imagen, vemos que se puede percibir una leve perdida de linealidad en los residuos cuando se estudian modificando las variables de *temperature* y *wind*, pero esto no ocurre contra la radiación solar.

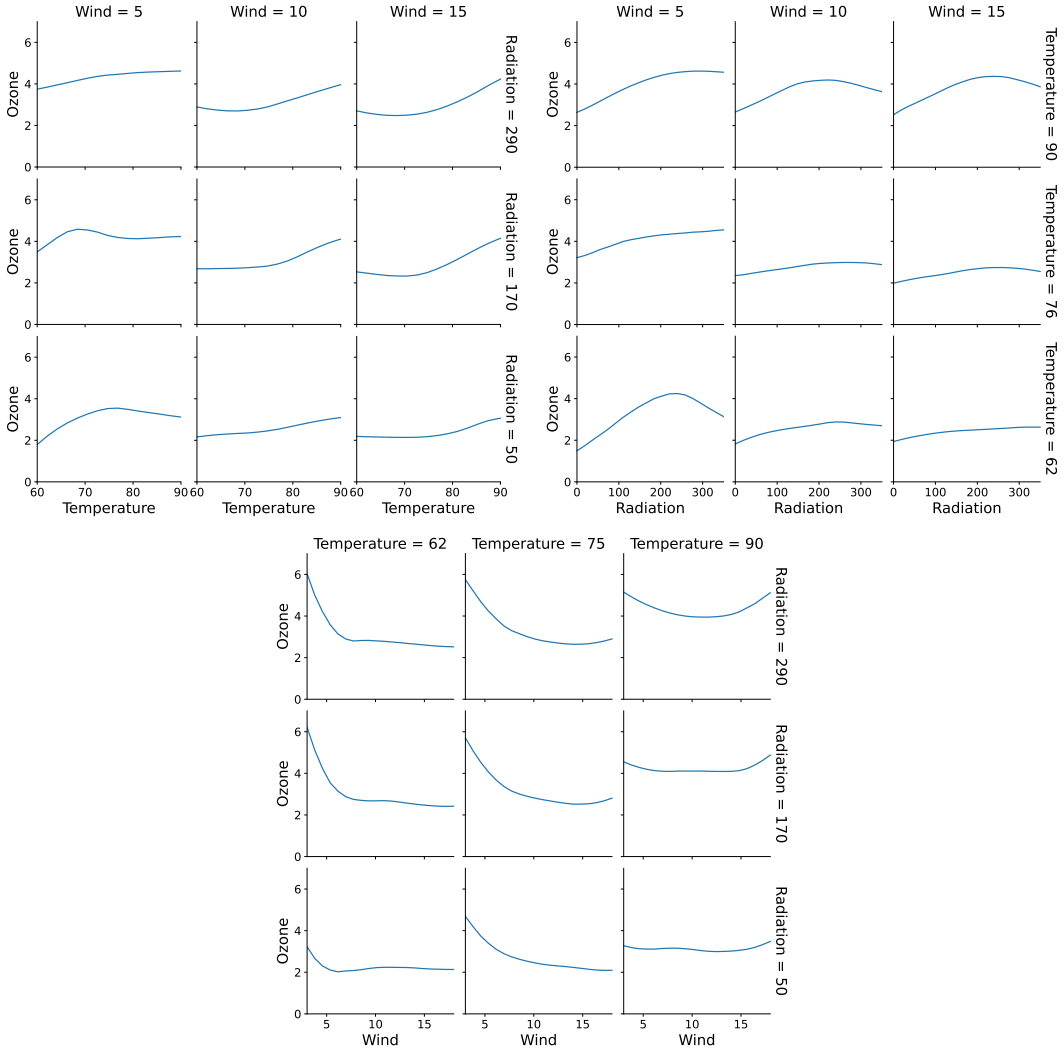


Figura 6: Gráfico multivariable de la raíz cúbica del ozono contra la variable independiente libre

3. Experimentación

Para la parte de experimentación, y a partir de los gráficos de diagnóstico implementados, buscaremos experimentar cómo varían estos mismos plots, con relación a sus parámetros (es decir, proporción y la cantidad de vecinos q, el grado de polinomio y función de distancia) y formularemos algunas hipótesis a partir de lo implementado.

Además de la experimentación de parámetros, buscaremos mostrar un análisis similar para un conjunto de datos sintéticos creados por nosotros. Los valores de los parámetros para los diferentes experimentos serán explicados en su respectiva subsección.

3.1. Cantidad de Vecinos

Buscamos experimentar sobre el aumento o decremento el tamaño del vecindario, para esto utilizamos una aproximación cuadrática, donde incrementamos gradualmente el tamaño de los vecinos y recreamos las figuras 4a y 4b del paper con la cantidad de vecinos probados.

Como hipótesis de este experimento, dado que las funciones que se intentan estimar tienen un término ε, el cual tiene una distribución normal de media 0 y varianza 1, y se encarga de agregar ruido a la función, es de esperar que al incrementar la cantidad de vecinos para cantidades pequeñas los residuos deben asemejarse más al ε. Mientras que a medida que aumenta significativamente la cantidad de vecinos los residuos se irán distando de este ε, esto porque si tomamos una mayor cantidad de vecinos existe el riesgo de fitear los datos que estamos evaluando con datos que disten demasiado a lo que buscamos ajustar, y esos puntos no deben ser muy relevantes para ajustar el dato actual (para valores de q muy altos empezamos a perder la idea de localidad del fit).

Luego, al incrementar o decrementar la cantidad de vecinos queremos corroborar que el suavizado sobre los puntos mejora cuando se incrementan mientras que el suavizado empeora cuando se reducen la cantidad de vecinos.

3.2. Ajuste lineal y cuadrático

Para este experimento pretendemos modificar el grado, es decir, realizar un cálculo lineal (grado = 1) o cuadrático (grado = 2) para fitear los valores del ozono. Recrearemos y analizaremos las figuras 4a y 5 del paper para una misma instancia de vecinos ($q = 44$). De esta manera, podremos ver cómo varía el ajuste lineal del cuadrático tanto para el QQ-Plot sin demostrar una gran influencia por el peso que genera una vecindad muy grande, y las variables independientes contra los residuos para ver cuál genera más residuo.

Nuestra hipótesis es que para el QQ-Plot encontraremos una mejor aproximación a una distribución normal para un ajuste cuadrático que lineal porque al haber más polinomial features, tiende a fitear mejor los datos lo cual también se habla en el paper.

Además, queremos ver cuan relacionadas están las variables independientes y los residuos, creemos que estarán más relacionadas (entonces la calidad del fit resulta inferior) para un ajuste lineal que cuadrático, ya que para el caso del cuadrático tendremos más parámetros para poder fitear mejor los datos.

3.3. Funciones de peso

Como mencionamos anteriormente en la sección 2, la función de pesos que se elabora en el paper de Cleveland [2] y la utilizada en la gran mayoría de este trabajo es la función tricúbica, pero es esperable que utilizando otra definición de pesos los resultados obtenidos difieran a los actuales. Es por ello, que resulta interesante estudiar otra forma de asignar estos pesos a los vecinos, inicialmente decidimos probar con la función B que utilizamos para $robust\_fit$ pero los resultados no parecían diferir si simplemente intercambiábamos la función W, probamos definir funciones iguales a la tricúbica, pero con distintos grados (ejemplos: 5, 7, etc.) pero los resultados siempre parecían idénticos. Optamos entonces por definir una función de peso distinta:

$$Z(x) = \begin{cases} \sqrt[3]{1 - \sqrt[3]{x}} & 0 \leq x < 1 \\ 0 & 0 > x \vee x \geq 1 \end{cases}$$

Podemos ver que la función Z cumple los requisitos marcados en el paper de 1979 [1] para la función de peso (la función da resultados positivos, no creciente y dará 0 para todo número mayor o igual a 1) con lo cual debería parecerse a la tricúbica hasta cierto punto. Sin embargo, podemos ver

con un gráfico como varían estas funciones:
Podemos ver como estas funciones presentan comportamientos opuestos cerca de los extremos (Z

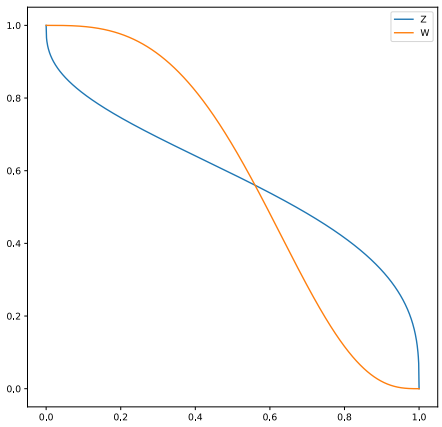


Figura 7: Funcion Z vs W en intervalo no nulo (entre 0 y 1)

empieza decreciendo más rápido que W, mientras que pasa lo opuesto cerca del final en el otro extremo)

Para este experimento manejamos la hipótesis de que si tomamos curvas que tengan una pendiente suficientemente pronunciada, Z dará mejores resultados, es decir fittea mejor los datos, que W. Como Z tiende a perder el peso más rápido que W para los puntos más cercanos del vecindario estos puntos tienden a perder importancia rápidamente, lo cual parecería útil a la hora de fittear funciones que cambian rápidamente. Esto nos lleva a creer que la función Z debe ajustar mejor que la W para funciones de este estilo.

3.4. Datos sintéticos

Para esta parte de la experimentación, se quiere probar el modelo con diferentes datos, elaborados manualmente, para ello generamos tres features de manera aleatoria, 2 de ellas tendrán una distribución normal y la restante uniforme. Para elaborar cada y_i, vamos a armar de manera similar a como fue presentado en el paper de Cleveland [1] pero extendiendo la idea a tres x_i, de igual forma se toman tres coeficientes de manera también aleatoria que acompañen a cada x_i. Finalmente, la fórmula que genera cada y_i queda definida de la siguiente manera:

$$y_i = C_1 * x_1^2 + C_2 * x_2 - C_3 * x_3 + \varepsilon_i \tag{1}$$

Cada ε son errores o ruido que tiene una distribución normal, con esperanza 0 y varianza 1. Al generar los datos de esta forma, es posible ver que se cumplen las hipótesis para aplicar Loess, en la figura 8 se observa que utilizando esta ecuación los residuos se ajustan correctamente a los datos de cada una de las variables x_i.

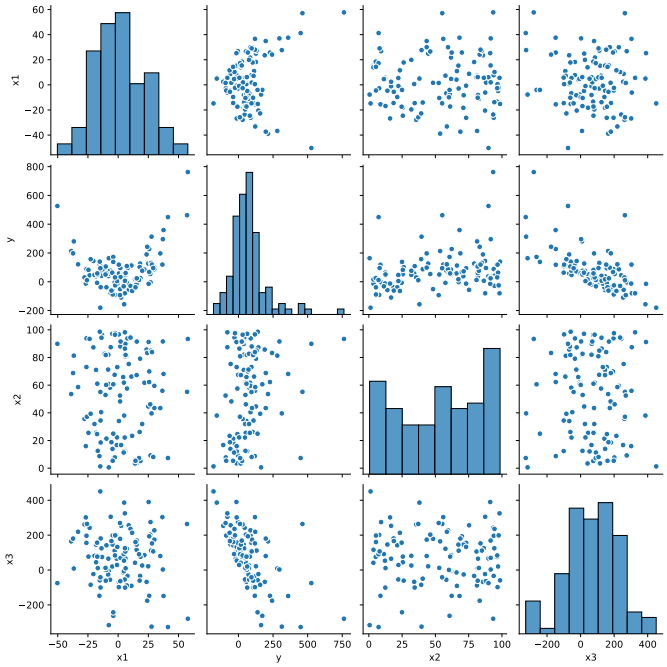


Figura 8: Pariplot de las variables independientes y la variable dependiente

4. Resultados

4.1. Cantidad de Vecinos

Para los primeros valores de la experimentación pensamos que sean alejados entre sí para poder distinguir bien los resultados. Sin embargo, durante la experimentación obtuvimos 3 instancias bien distinguibles entre sí, estas se pueden ver en la Figura 9.

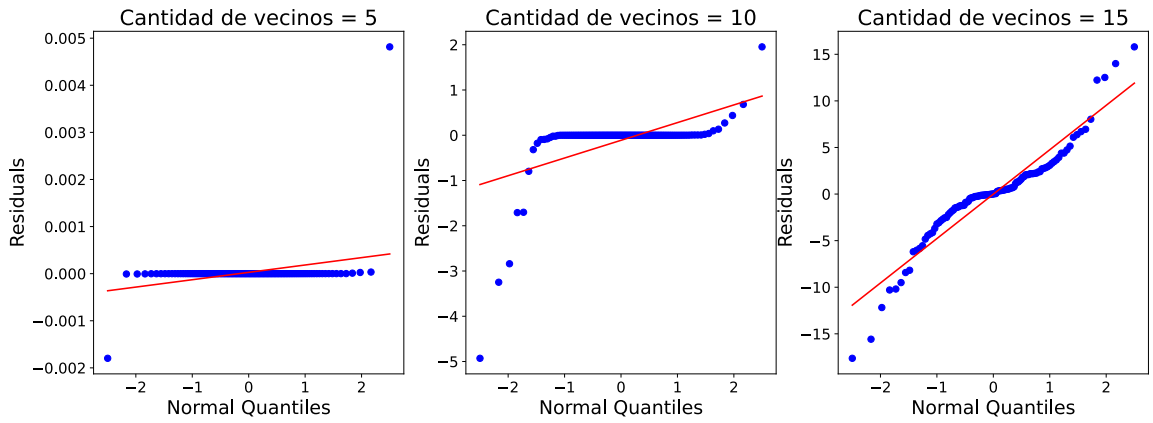


Figura 9: Gráfico de QQplots para distintas cantidades de vecinos

De izquierda a derecha se tomaron 5, 10 y 15 vecinos para el vecindario. Como, podemos observar, obtuvimos que para valores muy chicos, es decir de 1 a 10, la cantidad de vecinos parecía ser muy poca como para encontrar una diferencia notable en los residuos, ya que por lo general eran todos muy cercanos a 0. Luego, a medida que se toma más vecinos en consideración, se empieza a parecer cada vez más a la distribución buscada, y además se empiezan a dispersar más los datos de los residuos.

Luego, para los residuos absolutos contra los valores fiteados, mostrado en la Figura 10 podemos apreciar mejor cómo a medida que aumenta la cantidad de vecinos, se va notando un claro aumento de los residuos en su valor absoluto, para $q = 5$, los residuos no superan el 0.005, mientras que ya para $q = 15$, los valores ya alcanzan el 17. Esto ocurre ya que tenemos un tamaño de vecindario muy chico como para construir un buen fit, con lo cual los valores fitteados son muy parecidos a los valores originales de y (para $q = 5$ y $q = 10$, pero especialmente para $q = 5$).

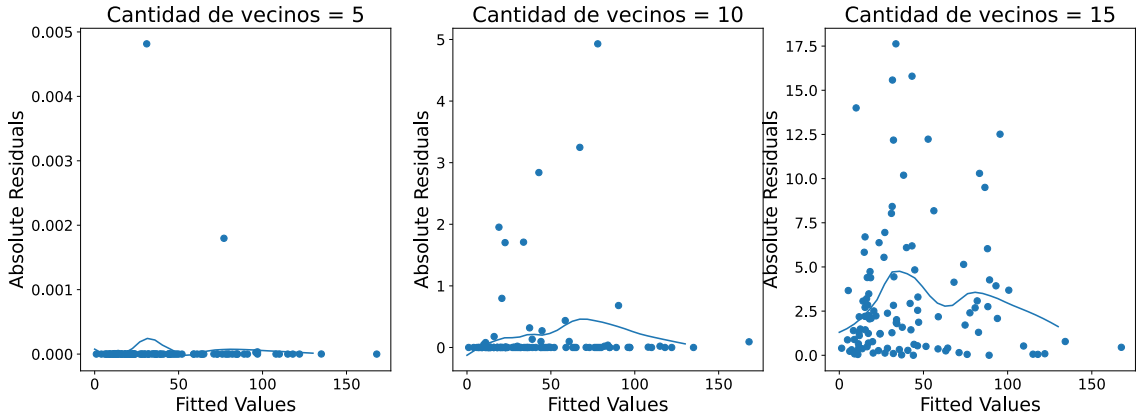


Figura 10: Gráfico de valores fiteados contra el absoluto de los residuos para distintas cantidades de vecinos

Veamos otro experimento posible donde las cantidades de vecinos usadas están más separadas para poder ver como aumentar la cantidad de vecinos para valores más altos tiene resultados distintos. Podemos ver que pasa en un ejemplo simple, usaremos fitting lineal univariado sobre un conjunto de datos sintéticos y veremos como cambian los residuos e intentaremos visualizar la suavidad del fit:

Para esto armamos una función (para este ejemplo tomamos $f(x) = \frac{x^2}{50}$) y tomamos 100 puntos de una distribución uniforme dentro del rango que nos interese estudiar de esta función (tomamos desde 0 hasta 50), luego le agregamos ruido sumando 100 observaciones de una distribución normal con media 0 y varianza 1. De esta forma construimos un conjunto de observaciones y además tenemos la función original, lo cual puede servir para ver la calidad de nuestro fit.

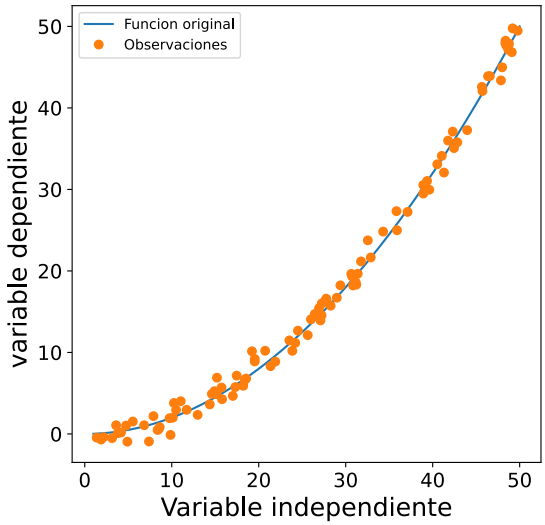


Figura 11: Función original y observaciones

Queremos ver los datos (observaciones) y su función original, por este motivo vemos la función como una línea y los datos observados como puntos (estos datos salen de agregarle ruido a puntos aleatorios de la función), luego probamos fittear con distintos valores de q y comparamos los resultados (vamos a comparar los gráficos de component-residuals y los gráficos de las superficies fitteadas para ver cuan suave es cada una, con superficie en este caso nos referimos a la curva sobre la cual se encuentran los puntos fitteados).

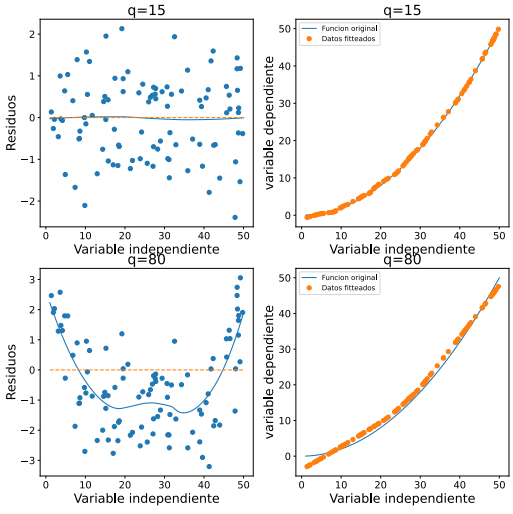


Figura 12: A la izquierda residuos vs variable independiente, a la derecha comparación entre función y puntos fitteados, tomando $q = 15$ y $q = 80$

Podemos ver que al usar más vecinos la superficie fitteada es más suave, comprobando la hipótesis que habíamos planteado, pero a su vez se acerca menos a la curva de la función original. Además, podemos ver en el gráfico de residuos que al usar 80 vecinos hay una dependencia más clara entre los residuos y la variable independiente, notemos también que con 15 vecinos tendremos residuos que se

parecen a una distribución normal (lo cual esperábamos), esto nos lleva a creer que usar 80 vecinos en este caso no es adecuado para fittear linealmente los datos. Vale la pena aclarar que para grados más altos probablemente necesitemos más puntos para conseguir un mejor fit ya que tenemos más parámetros.

4.2. Ajuste lineal y cuadrático

Para estos resultados terminamos obteniendo un gráfico similar para los QQ-Plots ya que ambos se asemejan bastante a una distribución normal. Sin embargo, podemos ver como los residuos del fit lineal son mayores a los del fit cuadrático, como suponíamos que pasaría.

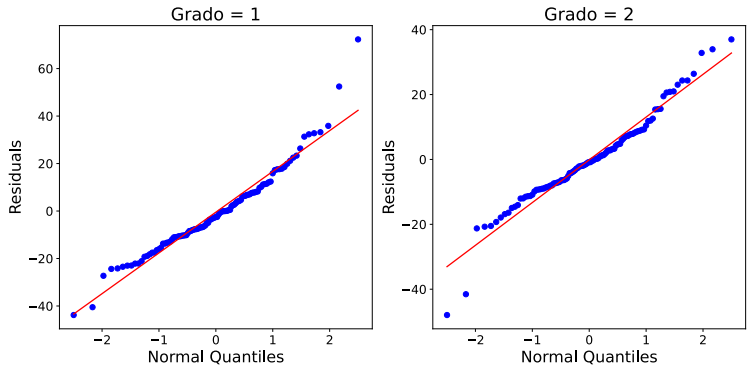


Figura 13: QQ-Plots para q=44 y diferentes grados de ajuste

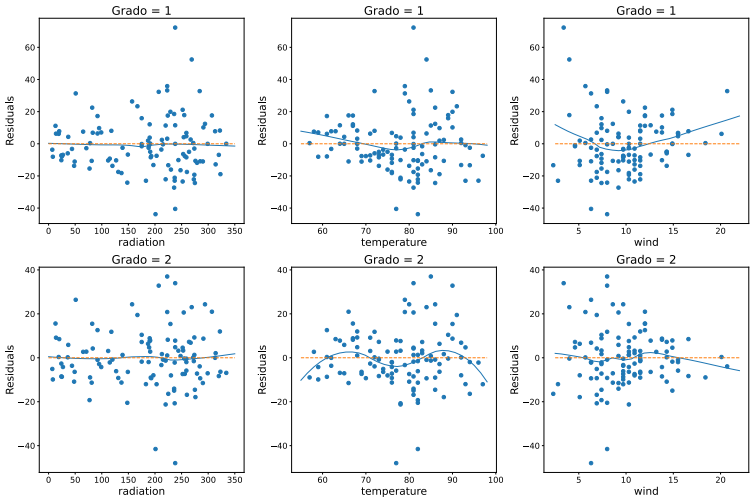


Figura 14: Gráfico de residuos contra las variables independientes para distintos ajustes, fijando $q = 44$

Luego, como se observa en el segundo gráfico (Figura 14) los residuos presentan una escala diferente, es decir que utilizando grado 2 podemos obtener residuos más pequeños que utilizando grado = 1, pero dado que no estamos variando la cantidad de vecinos es posible que utilizando un vecindario de mayor tamaño se observen resultados diferentes. Para observar que ocurre cuando variamos la cantidad de vecinos en este caso tomamos el doble de los vecinos utilizados, es decir q = 88, de esta manera como se ve en la Figura 15 los errores aumentaron pero lo hacen para grado 1 como para grado 2, pero todavía manteniendo el hecho que utilizar grado 2 genera menos residuos que grado 1.

Finalmente, queremos verificar si estamos fiteando mejor los datos utilizando grado 2 o grado 1, ya estudiamos que ocurre entre los residuos pero como se puede ver en las figuras 14 y 15 tanto para radiación como viento utilizando grado 2 es suficiente para fittear los puntos y podemos observar normalidad (especialmente para el viento al comparar grado 1 y grado 2) en los gráficos de variables independientes vs residuos, pero solamente viendo la temperatura siempre aparecen curvaturas cuando la temperatura está cerca de los 65 y 90, pero usando grado 2 y $q = 88$ cuando la temperatura vale 65 ya se observa una curva más parecida a una recta debido al aumento en la normalidad de los residuos, indicando que nuestro fit está mejorando.

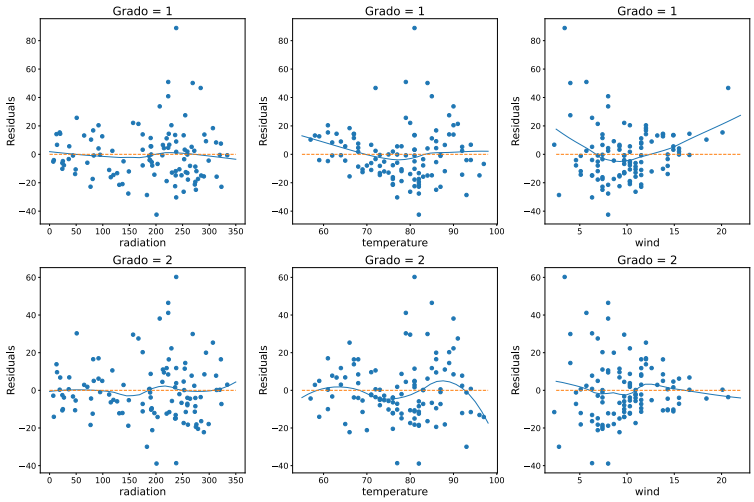


Figura 15: Gráfico de residuos contra las variables independientes para distintos ajustes, fijando $q = 88$

4.3. Funciones de peso:

Para este experimento la hipótesis era que si comparábamos los fits realizados con la función Z contra los realizados con la función W para funciones con pendiente pronunciada veríamos que la función Z produce mejores fits. Sin embargo, durante la experimentación preliminar notamos que los resultados del fit eran más similares de lo esperado entre Z y W incluso para funciones distintas con distintas pendientes. Por este motivo decidimos incluir el ejemplo donde encontramos más diferencia entre los resultados, ya que este resulta más relevante.

Para este experimento usamos datos sintéticos, planteamos una función $f(x)$ (en este caso tomamos $f(x) = ln(151 + x)$) y tomamos 100 puntos de una distribución uniforme (llamémoslos x_i) en el rango que nos interesa estudiar de la función (desde -150 hasta 150) y luego tomamos $f(x_i)$ para cada i y le sumamos observaciones de una distribución normal de esperanza 0 y varianza 1. Para este experimento usamos varias funciones distintas con distintas pendientes y probamos fitear con distintos parámetros.

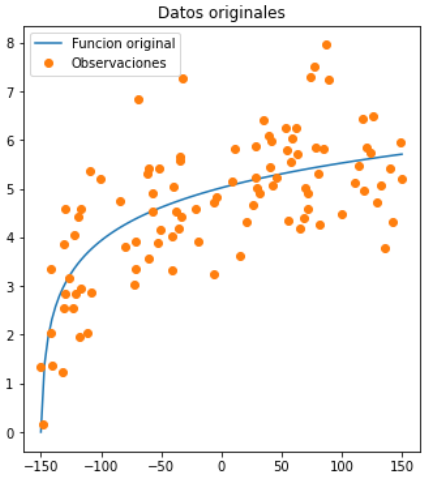


Figura 16: Función original y observaciones

En la figura 17 se observa la función que mayor diferencia encontramos, de todas las probadas, como se puede ver las diferencias más notorias se encuentran en los valores entre el 0 y 100 para las comparaciones entre funciones y puntos fitteados. En este caso podemos ver como el fit de la función Z parece oscilar menos y estar más cerca a la curva de la función original en comparación con el fit realizado por la función W, esto parece indicar que Z es la función que mejor fitea los datos en este caso, pero aun así los fits realizados no difieren tanto de los fits realizados por W. Podemos decir que la hipótesis queda negada en este caso ya que esperábamos observar una mejora (o al menos diferencia) de forma consistente en ciertos casos, pero esto no fue con lo que nos encontramos en la mayoría de los casos, ya que la diferencia de resultados obtenidos usando Z y W difirieron menos de lo esperado. A pesar de esto, decidimos incluir el ejemplo donde notamos la mayor diferencia entre Z y W de todas las funciones y parámetros con los que experimentamos, si bien en este caso parece que Z consigue un fit ligeramente mejor que W, esto no se cumple siempre.

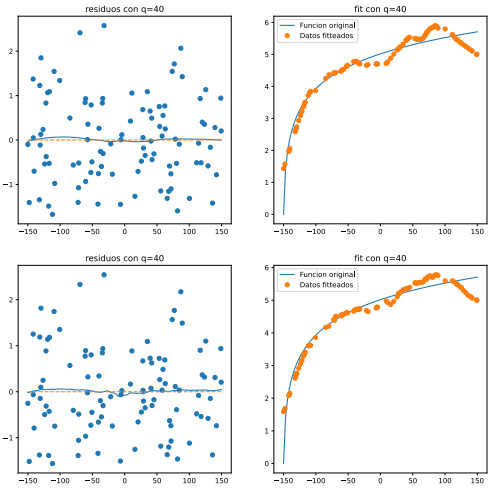


Figura 17: A la izquierda residuos vs variable independiente, a la derecha comparación entre función y puntos fitteados usando grado 2 y $q = 40$, arriba utilizando W y abajo Z

4.4. Datos sintéticos

Como se puede ver en las figuras 18 se puede observar una clara relación de normalidad comparando los residuos del fit en relación a las variables independientes, especialmente para x_3.

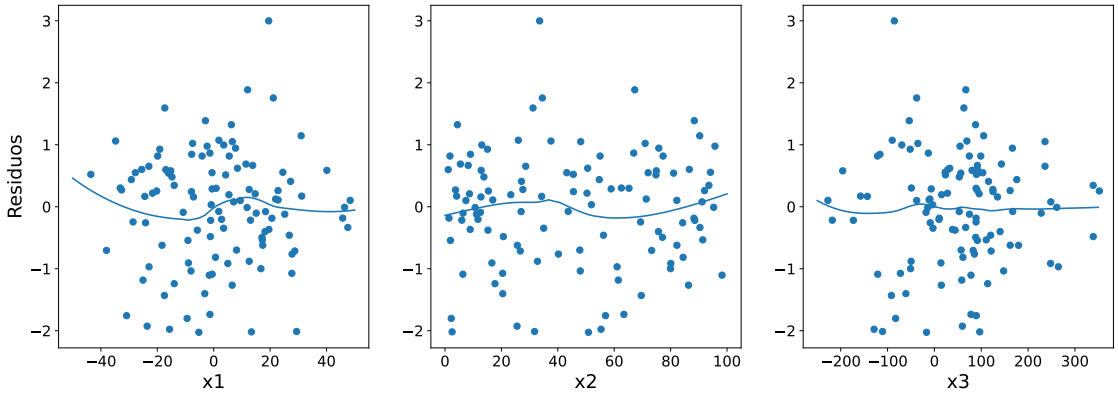


Figura 18: Gráfico de los residuos contra las variables independientes usando q=44 y $grado$=2

A raíz de estos resultados, consideramos que es necesario probar datos elegidos con una fórmula aún más compleja para intentar poner el modelo a prueba, es por ello que se decidió armar otra ecuación aparte de la presentada en 1, la idea es que la ecuación nueva aplique funciones a los previos x_1, x_2 y x_3:

$$y_i = sen(x_1) + 5 * (x_2^5) - cos(x_3) + \varepsilon_i \tag{2}$$

Esperamos que esta fórmula para los grados 2 y 3 no sea capaz de describir lo que está ocurriendo con cada una de las variables. Como se ve en la figura 19, la variable que menos ajustada se ve es la x_2 mientras que el resto de las variables parecen tener normalidad aceptable, dado que en x_1 solamente aparecieron curvaturas en los puntos -40, 0 y 40, mientras que para x_3 solamente se distorsionaron aquellos puntos alrededor del 100. Sin embargo, es importante notar que la magnitud de los residuos es realmente grande ya que el eje de los residuos (eje y) tiene una escala de $1e8$, esto nos indica que utilizar estos parámetros no concluyen con un buen fit ya que los residuos se encuentran en una escala que supera ampliamente la de y ya que todos estos datos tienen un módulo menor a 300.

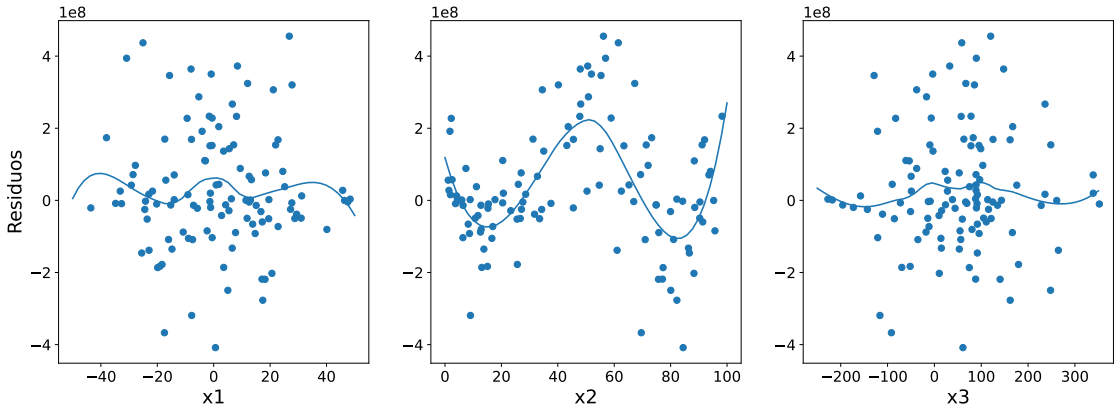


Figura 19: Tomando q = 44 y grado=3

Finalmente, como se observa en la figura 20 tomando grado 5 es lo mejor que logramos ajustar cada una de las variables, notemos que la mayoría de los residuos se encuentran cercanos a la recta

donde residuo = 0, esto puede estar causado porque quizás es necesario tomar mayor cantidad de vecinos para continuar ajustando los residuos. El problema en este caso es que ya no quedan muchos puntos como para seguir subiendo el q. Además, estamos intentando de conseguir una cantidad de parámetros que supera la cantidad de puntos que utilizamos para fittear ($cantX_i^{grado} + 1 > q$), y también los residuos terminan re escalados a valores significativamente más pequeños a diferencia de los grados 2 y 3.

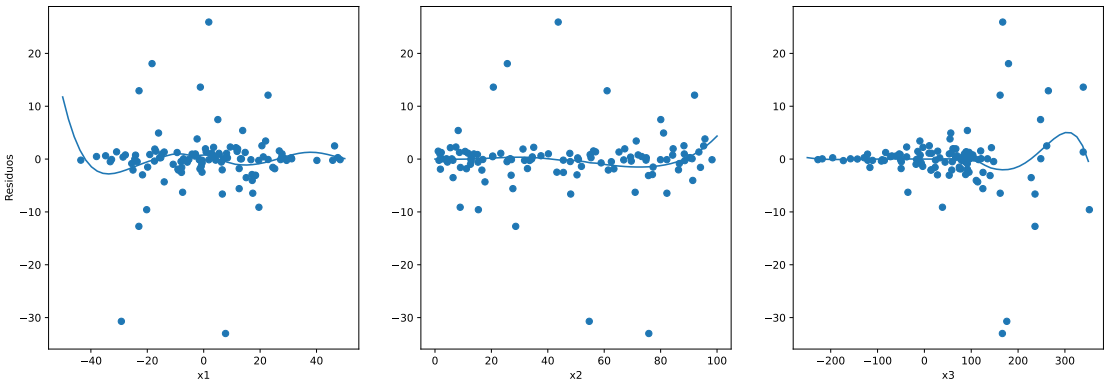


Figura 20: Tomando q = 100 y grado=5

5. Conclusión

Con los resultados obtenidos durante la etapa de experimentación, podemos ver la importancia de respetar y verificar las hipótesis iniciales, además logramos ver como cambiar los parámetros afecta a los resultados obtenidos y como cambiar estos parámetros puede hacer la diferencia entre pasar la etapa de diagnóstico y no pasarla. Elaboramos experimentos que corroboran la influencia entre la cantidad de vecinos y el grado que se toma para fittear los puntos y como a medida que crecen o decrecen que impacto tiene en los residuos. Estudiamos que diferentes funciones de peso pueden generar resultados similares, y no necesariamente al variar funciones de peso deberían ajustar mucho mejor los datos en algún contexto específico. Experimentamos con data sintética y vimos como este método puede ser mejor de lo que esperábamos consiguiendo resultados decentes para funciones complejas si ajustamos los parámetros y probamos distintos valores.

Se pudo replicar con éxito los gráficos y conjeturas obtenidas de los papers [1] y [2], los cuales fueron de gran ayuda para entender e implementar el modelo y lograr las modificaciones necesarias para la experimentación.

Pudimos ver también como el método puede no ser tan bueno si se intenta fittear funciones más complejas con pocos puntos (último experimento de datos sintéticos). Como posible trabajo futuro, se podría intentar de fittear funciones más complejas utilizando más puntos. Otros posibles trabajos futuros serían experimentar con la función $robust\_fit$ (ver sección 6.2) y también probar el ajuste que podemos lograr con mayor cantidad de datos, de esta manera se podría evaluar al modelo sobre vecinos más lejanos utilizando las funciones ya conocidas, también experimentar utilizando funciones de peso más complejas donde los datos no se ajusten de manera muy similar a la tricúbica. Adicionalmente, se podría experimentar utilizando distintos métodos de estandarización/normalización.

6. Comentarios Adicionales

En esta sección, hablaremos de algunas cosas mencionables, cosas que nos parecieron interesantes o simplemente cosas para tener en cuenta si se desea continuar con la experimentación.

6.1. Estandarización

En cuanto a la estandarización, notamos que si estandarizábamos los datos (de la forma que indica el paper), podemos replicar exactamente los mismos gráficos. Sin embargo, algunos gráficos quedaban distintos a los del paper, decidimos probar sin estandarizar, de esta forma vimos que algunos gráficos dejaban de coincidir, pero empezaban a coincidir otros.

Para replicar de forma exacta todos los gráficos del paper, estandarizamos X para las figuras 6, 7 y 8 del paper. Mientras que para las figuras 4 y 5 del paper no estandarizamos (para la figura 3 del paper esto no importa, ya que el gráfico se realiza sobre los datos iniciales), de esta forma conseguimos que todos los gráficos sean exactamente iguales a los del paper.

Para poder hacer esto de forma más prolija y cómoda, decidimos incluir un booleano en la función fit y predict que nos permita decidir si se aplicara loess sobre los datos estandarizados o no estandarizados.

6.2. Robustness

Para el punto obligatorio del trabajo práctico que indicaba que debíamos experimentar con otra función de distancia (que no sea la tricúbica), nuestra idea inicial era implementar la idea de robustness del paper univariado de 1979, esto usaba la función B:

$$B(x) = \begin{Bmatrix} (1-x^2)^2 & |x| < 1 \\ 0 & |x| \geq 1 \end{Bmatrix} \tag{3}$$

La idea era aplicar loess fit, y luego volver a aplicarlo, pero ajustando los pesos en función de los residuos, este proceso se repetía t veces (el paper recomendaba $t = 2$, ya que seguir aumentando el t era más costoso y no conseguía mejorar mucho más los resultados). Para calcular los pesos se usaban las mismas W_i que antes (usaremos $armarW$ igual a como lo hacíamos antes) multiplicándola por otra matriz diagonal de pesos, WB la cual calcula pesos basándonos en los residuos de la siguiente forma:

$$WB_{ii} = B(\frac{e_i}{6s})$$

Siendo $e_i = y_i - \widehat{y}_i$ el i-ésimo residuo del $(k-1)$-ésimo fit y s la mediana de los $|e_i|$. Luego el peso de la i-ésima ecuación normal pesada para calcular el k-ésimo fit está dado por $\widetilde{W}_i = W_i * WB$, el resto del proceso de loess es igual al que se hacía antes, nada más que debemos repetir este proceso para cada fit. Este proceso se repite t veces, fiteando nuevamente en cada iteración, con el objetivo de darle menos peso a los puntos que más residuos tienen, ya que los tomaremos como datos menos relevantes. Esto último es lo que le da el nombre al método, ya que debería ser más robusto y resistente frente a dichos casos.

Sin bien conseguimos implementar el método, no conseguimos probarlo demasiado, en parte por falta de tiempo y, por otro lado, no teníamos contra que comparar los resultados, con lo cual decidimos no incluirlo en la experimentación, si bien lo entregamos en el código (función $robust\_fit$ o función fit con argumento opcional t, la última llama a la primera en caso de que $t > 0$).

Referencias

[1] William S. Cleveland,Robust Locally Weighted Regression and Smoothing Scatterplots 1979.

[2] William S. Cleveland and Susan J. Devlin, Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting, 1988.