Notas Curso de Estadística II

Maikol Solís Chacón y Luis Barboza Chinchilla

Actualizado el 09 mayo, 2022

Índice general

Capítulo 1

Introducción

Estas son las notas de clase del curso CA0403: Estadística Actuarial II para el primer semestre del 2022.

Capítulo 2

Estimación no-paramétrica de densidades

2.1. Histograma

El histograma es una de las estructuras básicas en estadística y es una herramienta descriptiva que permite visualizar la distribución de los datos sin tener conocimiento previo de los mismos. En esta sección definiremos el histograma más como un estadístico que como una herramienta de visualización de datos.

2.1.1. Construcción Estadística

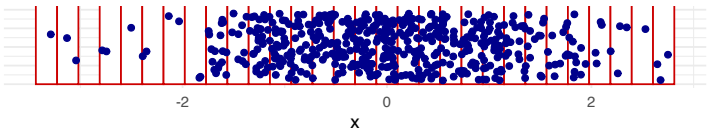
Suponga que $X_1, X_2, \ldots, X_n$ es una muestra independiente que proviene de una distribución desconocida f. En este caso no asumiremos que f tenga alguna forma particular, que permita definirla de manera paramétrica como en el curso anterior.

Construcción:

- Seleccione un origen x_0 y divida la linea real en *segmentos*.

$$B_j = [x_0 + (j-1)h, x_0 + jh), \quad j \in \mathbb{Z}$$

- Cuente cuántas observaciones caen en el segmento B_j. Denótelo como n_j.



- Divida el número de observaciones en B_j por el tamaño de muestra n y el ancho de banda h de cada caja.

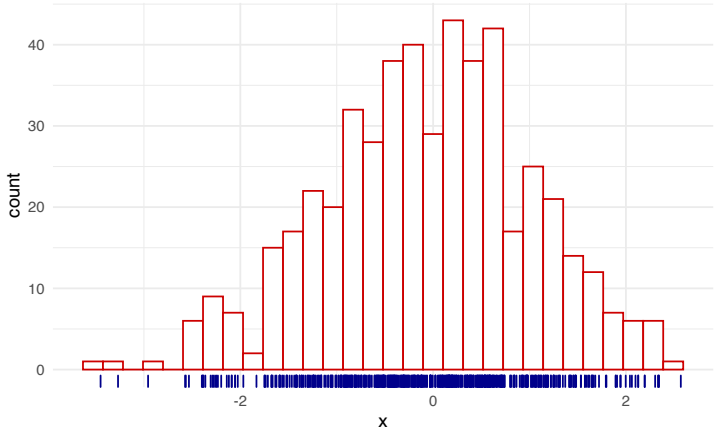
$$f_j = \frac{n_j}{nh}$$

  De esta forma si se suma las áreas definidas por el histograma da un total de 1.

- Cuente la frecuencia por el tamaño de muestra n y el ancho de banda h.

$$f_j = \frac{n_j}{nh}$$

- Dibuje el histograma.



Formalmente el histograma es el

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} \sum_{j} I(X_i \in B_j) I(x \in B_j),$$

donde I es la indicadora.

2.1.2. Construcción probabilística

Denote $m_j = jh - h/2$ el centro del segmento,

$$\begin{aligned}\mathbb{P}\left(X \in \left[m_j - \frac{h}{2}, m_j + \frac{h}{2}\right)\right) &= \int_{m_j - \frac{h}{2}}^{m_j + \frac{h}{2}} f(u)du \\ &\approx f(m_j)h\end{aligned}$$

Otra forma de aproximarlo es:

$$\mathbb{P}\left(X \in \left[m_j - \frac{h}{2}, m_j + \frac{h}{2}\right)\right) \approx \frac{1}{n}\#\left\{X \in \left[m_j - \frac{h}{2}, m_j + \frac{h}{2}\right)\right\}$$

Acomodando un poco la expresión

$$\hat{f}_h(m_j) = \frac{1}{nh}\#\left\{X \in \left[m_j - \frac{h}{2}, m_j + \frac{h}{2}\right)\right\}$$

2.1.3. Propiedades estadísticas

Note que el estimador de histograma $\hat{f}_h$ tiende a ser más suave conforme aumenta el ancho de banda h.

2.1.4. Propiedades estadísticas

Suponga que $x_0 = 0$ y que $x \in B_j$ es un punto fijo, entonces el estimador evaluado en x es:

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} I(X_i \in B_j)$$

2.1.5. Sesgo

Para calcular el sesgo primero calculamos:

$$\begin{aligned}\mathbb{E}\left[\hat{f}_h(x)\right] &= \frac{1}{nh}\sum_{i=1}^{n}\mathbb{E}\left[I(X_i \in B_j)\right] \\ &= \frac{1}{nh}n\mathbb{E}\left[I(X_i \in B_j)\right]\end{aligned}$$

donde $I(X_i \in B_j)$ es una variable Bernoulli con valor esperado:

$$\mathbb{E}\left[I(X_i \in B_j)\right] = \mathbb{P}\left(I(X_i \in B_j) = 1\right) = \int_{(j-1)h}^{jh} f(u)du.$$

Entonces,

$$\mathbb{E}\left[f_h(x)\right] = \frac{1}{h}\int_{(j-1)h}^{jh} f(u)du$$

y por lo tanto el sesgo de $\hat{f}_h(x)$ es:

$$Sesgo(\hat{f}_h(x)) = \frac{1}{h}\int_{(j-1)h}^{jh} f(u)du - f(x)$$

Esto se puede aproximar usando Taylor alrededor del centro $m_j = jh - h/2$ de B_j de modo que $f(u) - f(x) \approx f'(m_j)(u - x)$.

$$Sesgo(\hat{f}_h(x)) = \frac{1}{h}\int_{(j-1)h}^{jh}[f(u) - f(x)]du \approx f'(m_j)(m_j - x)$$

Entonces se puede concluir que:

- $\hat{f}_h(x)$ es un estimador sesgado de $f(x)$.
- El sesgo tiende a ser cero cerca del punto medio de B_j.
- El sesgo es creciente con respecto a la pendiente de la verdadera densidad evaluada en el punto medio m_j.

2.1.6. Varianza

Dado que todos los X_i son i.i.d., entonces

$$\begin{aligned}\text{Var}\left(\hat{f}_h(x)\right) &= \text{Var}\left(\frac{1}{nh}\sum_{i=1}^{n} I(X_i \in B_j)\right) \\ &= \frac{1}{n^2h^2} n\text{Var}\left(I(X_i \in B_j)\right)\end{aligned}$$

La variable I es una bernoulli con parametro $\int_{(j-1)h}^{h} f(u)du$ por lo tanto su varianza es el

$$\text{Var}\left(\hat{f}_h(x)\right) = \frac{1}{nh^2}\left(\int_{(j-1)h}^{h} f(u)du\right)\left(1 - \int_{(j-1)h}^{h} f(u)du\right)$$

Ejercicio 2.1. Usando un desarrollo de Taylor como en la parte anterior, pruebe que:

$$\text{Var}\left(\hat{f}_h(x)\right) \approx \frac{1}{nh} f(x)$$

Consecuencias:

- La varianza del estimador es proporcional a $f(x)$.
- La varianza decrece si el ancho de banda h crece.

2.1.7. Error cuadrático medio

El error cuadrático medio del histograma es el

$$\text{MSE}\left(\hat{f}_h(x)\right) = \text{E}\left[\left(\hat{f}_h(x) - f(x)\right)^2\right] = \text{Sesgo}^2\left(\hat{f}_h(x)\right) + \text{Var}\left(\hat{f}_h(x)\right).$$

Ejercicio 2.2. ¿Pueden probar la segunda igualdad de la expresión anterior?

Retomando los términos anteriores se puede comprobar que:

$$\text{MSE}\left(\hat{f}_h(x)\right) = \frac{1}{nh} f(x) + f' \left\{ \left(j - \frac{1}{2}\right) h \right\}^2 \left\{ \left(j - \frac{1}{2}\right) h - x \right\}^2 \tag{2.1}$$

$$+ o(h) + o\left(\frac{1}{nh}\right) \tag{2.2}$$

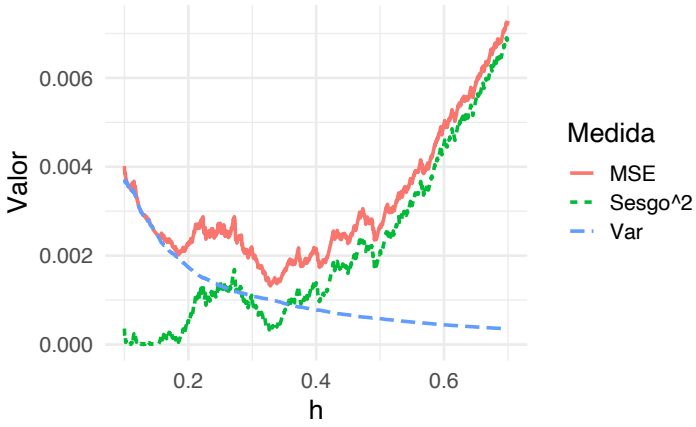
Nota: . Si $h \to 0$ y $nh \to \infty$ entonces $\text{MSE}\left(\hat{f}_h(x)\right) \to 0$. Es decir, conforme usamos más observaciones, pero el ancho de banda de banda no decrece tan rápido, entonces el error cuadrático medio converge a 0.

Como $\text{MSE}\left(\hat{f}_h(x)\right) \to 0$ (convergencia en $\mathbb{L}^2$) implica que $\hat{f}_h(x) \xrightarrow{\mathcal{P}} f(x)$, entonces $\hat{f}_h$ es consistente. Además según la fórmula (2.2), concluimos lo siguiente:

- Si $h \to 0$, la varianza crece (converge a ∞) y el sesgo decrece (converge a $f'(0)x^2$).
- Si $h \to \infty$, la varianza decrece (hacia 0) y el sesgo crece (hacia ∞)

Ejercicio 2.3. Si $f \sim N(0, 1)$, aproxime los componentes de sesgo, varianza y MSE, y grafíquelos para distintos valores de h.

Solución:



2.1.8. Error cuadrático medio integrado

Uno de los problemas con el $\text{MSE}\left(\hat{f}_h(x)\right)$ es que depende de x y de la función de densidad f (desconocida). Integrando con respecto a x el MSE se logra

resolver el primer problema:

$$\begin{aligned}\text{MISE}\left(\hat{f}_h\right) &= \text{E}\left[\int_{-\infty}^{\infty}\left\{\hat{f}_h(x) - f(x)\right\}^2 dx\right] \\ &= \int_{-\infty}^{\infty}\text{E}\left[\left\{\hat{f}_h(x) - f(x)\right\}^2\right] dx \\ &= \int_{-\infty}^{\infty}\text{MSE}(\hat{f}_h(x))\, dx\end{aligned}$$

Al MISE se le llama error cuadrático medio integrado. Además,

$$\begin{aligned}\text{MISE}(\hat{f}_h) &\approx \int_{-\infty}^{\infty}\frac{1}{nh}f(x)dx \\ &+ \int_{-\infty}^{\infty}\sum_j I(x \in B_j)\left\{\left(j-\frac{1}{2}\right)h - x\right\}^2\left[f'\left(\left\{j-\frac{1}{2}\right\}h\right)\right]^2 dx \\ &= \frac{1}{nh} + \sum_j\left[f'\left(\left\{j-\frac{1}{2}\right\}h\right)\right]^2\int_{B_j}\left\{\left(j-\frac{1}{2}\right)h - x\right\}^2 dx \\ &= \frac{1}{nh} + \frac{h^2}{12}\sum_j\left[f'\left(\left\{j-\frac{1}{2}\right\}h\right)\right]^2 \\ &\approx \frac{1}{nh} + \frac{h^2}{12}\int\{f'(x)\}^2 dx \\ &= \frac{1}{nh} + \frac{h^2}{12}\|f'\|_2^2\end{aligned}$$

la cual es una buena aproximación si $h \to 0$. A este último término se le llama MISE asintótico.

2.1.9. Ancho de banda óptimo para el histograma

El MISE tiene un comportamiento asintótico similar al observado en el MSE. La figura siguiente presenta el comportamiento de la varianza, sesgo y MISE para nuestro ejemplo anterior:

Un problema frecuente en los histogramas es que la mala elección del parámetro h causa que estos no capturen toda la estructura de los datos. Por ejemplo, en

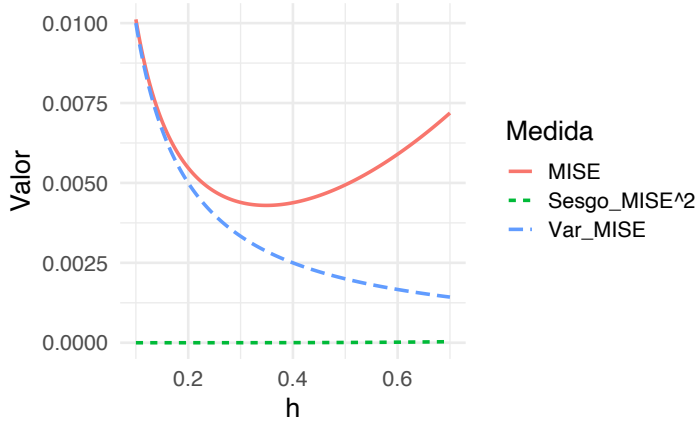
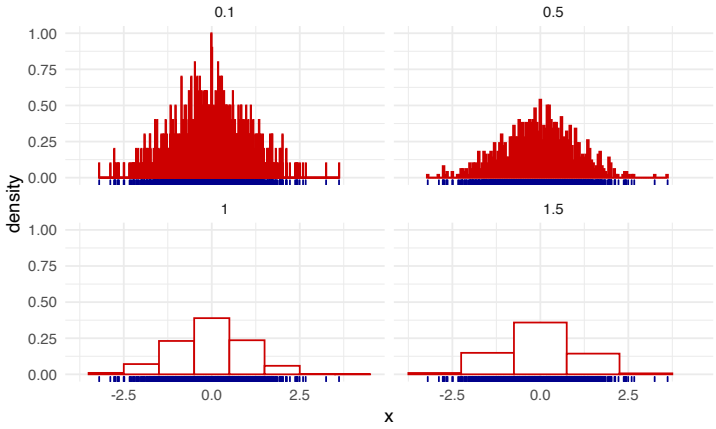


Figura 2.1:

el siguiente caso se muestra histogramas construídos a partir de 1000 números aleatorios según una $N(0,1)$, bajo 4 distintas escogencias de ancho de banda.



Un criterio más preciso para seleccionar el ancho de banda es a través de la minimización del MISE:

$$\frac{\partial \text{MISE}(f_h)}{\partial h} = -\frac{1}{nh^2} + \frac{1}{6}h\|f'\|_2^2 = 0$$

lo implica que

$$h_{opt} = \left(\frac{6}{n\|f'\|_2^2}\right)^{1/3} = O\left(n^{-1/3}\right).$$

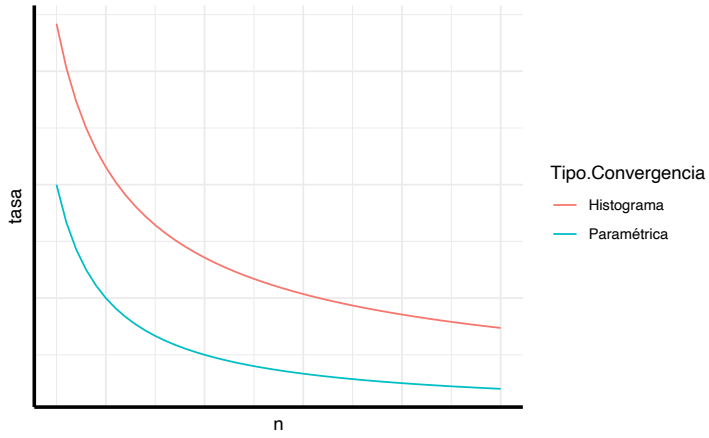
y por lo tanto

$$\text{MISE}(\hat{f}_h) = \frac{1}{n}\left(\frac{n\|f'\|_2^2}{6}\right)^{1/3}$$

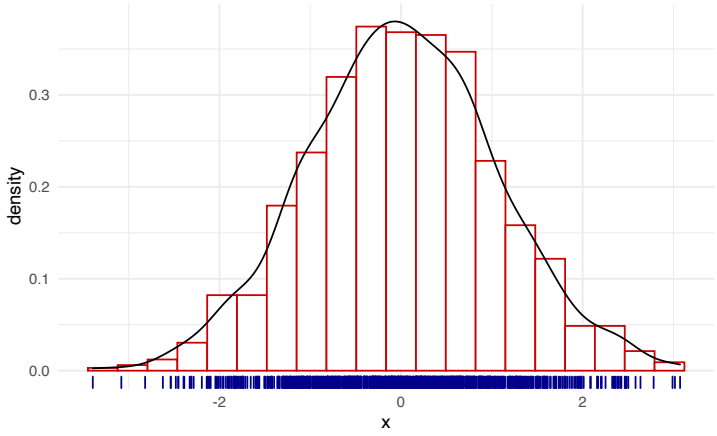
Nota: (Recuerde de Estadística I). Si $X_1, \ldots, X_n \sim f_\theta$ i.i.d, con $\text{Var}(X) = \sigma^2$ y media θ, recuerde que el estimador $\hat{\theta}$ de θ tiene la característica que

$$\text{MSE}(\theta) = \text{Var}(\hat{\theta}) + \text{Sesgo}^2(\hat{\theta}) = \frac{\sigma^2}{n}$$

Según la nota anterior la tasas de convergencia del histograma es más lenta que la de un estimador parámetrico considerando la misma cantidad de datos, tal y como se ilustra en el siguiente gráfico:



Finalmente, podemos encontrar el valor óptimo del ancho de banda ($h = 0.3285$) del conjunto de datos en el ejemplo anterior.



Ejercicio 2.4. Verifique que en el caso normal estándar: $h_{opt} \approx 3{,}5n^{-1/3}$.

2.2. Estimación de densidades basada en kernels.

2.2.1. Primera construcción

Sea $X_1, \ldots, X_n$ variables aleatorias i.i.d. con distribución f en $\mathbb{R}$. La distribución de f es $F(x) = \int_{-\infty}^{x} f(t)dt$.

La distribución empírica de F es:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x).$$

Por la ley de los grandes números tenemos que $F_n(x) \xrightarrow{c.s.} F(x)$ para todo x en $\mathbb{R}$, conforme $n \to \infty$. Entonces, $F_n(x)$ es un estimador consistente de $F(x)$ para todo x in $\mathbb{R}$.

Nota: . ¿Podríamos derivar F_n para encontrar el estimador $\hat{f}_n$?

La respuesta es si (más o menos).

Suponga que $h > 0$ tenemos la aproximación

$$f(x) \approx \frac{F(x+h) - F(x-h)}{2h}.$$

Remplazando F por su estimador F_n, defina

$$\hat{f}_n^R(x) = \frac{F_n(x+h) - F_n(x-h)}{2h},$$

donde $\hat{f}_n^R(x)$ es el estimador de *Rosenblatt* .

Podemos rescribirlo de la forma,

$$\hat{f}_n^R(x) = \frac{1}{2nh}\sum_{i=1}^{n} I(x-h < X_i \leq x+h) = \frac{1}{nh}\sum_{i=1}^{n} K_0\left(\frac{X_i - x}{h}\right)$$

con $K_0(u) = \frac{1}{2}I(-1 < u \leq 1)$, lo cual es equivalente al caso del histograma.

2.2.2. Otra construcción

Con el histograma construimos una serie de segmentos fijo B_j y contabamos el número de datos que estaban **contenidos en** B_j

Nota: . ¿Qué pasaría si cambiamos la palabra **contenidos** por **alrededor de "x"**?

Suponga que se tienen intervalos de longitud $2h$, es decir, intervalos de la forma $[x-h, x+h)$.

El estimador de histograma se escribe como

$$\hat{f}_h(x) = \frac{1}{2hn}\#\{X_i \in [x-h, x+h)\}.$$

Note que si definimos

$$K(u) = \frac{1}{2}I\left(|u| \leq 1\right)$$

con $u = \frac{x-x_i}{h}$, entonces parte del estimador de histograma se puede escribir como:

$$\frac{1}{2}\#\{X_i \in [x-h, x+h)\} = \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) = \sum_{i=1}^{n} \frac{1}{2} I\left(\left|\frac{x-x_i}{h}\right| \leq 1\right)$$

Finalmente se tendría que

$$\hat{f}_h\left(x\right) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$$

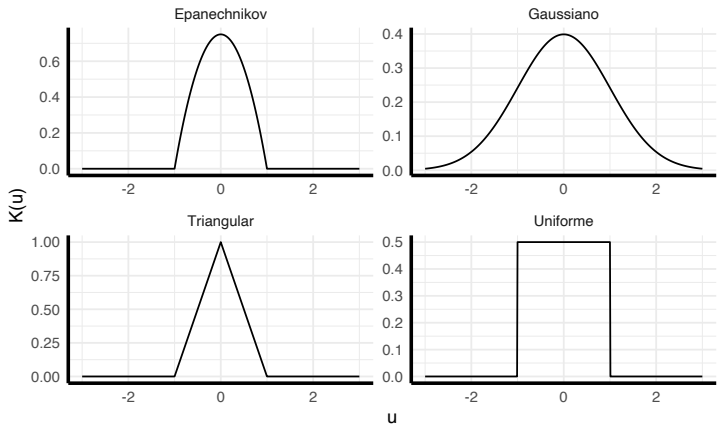
Nota: . ¿Qué pasaría si cambiaríamos la función K del histograma por una más general? Esto permitiría incluir la noción de "cercanía" de cada dato alrededor de x.

Esta función debería cumplir las siguientes características:

- $K(u) \geq 0$.
- $\int_{-\infty}^{\infty} K(u)du = 1$.
- $\int_{-\infty}^{\infty} uK(u)du = 0$.
- $\int_{-\infty}^{\infty} u^2K(u)du < \infty$.

Por ejemplo:

- **Uniforme:** $\frac{1}{2}I\left(|u| \leq 1\right)$.
- **Triangular:** $(1 - |u|)I\left(|u| \leq 1\right)$.
- **Epanechnikov:** $\frac{3}{4}(1 - u^2)I\left(|u| \leq 1\right)$.
- **Gaussian:** $\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right)$.



Entonces se tendría que la expresión general para un estimador por núcleos (kernel) de f:

$$\hat{f}_h\left(x\right) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i)$$

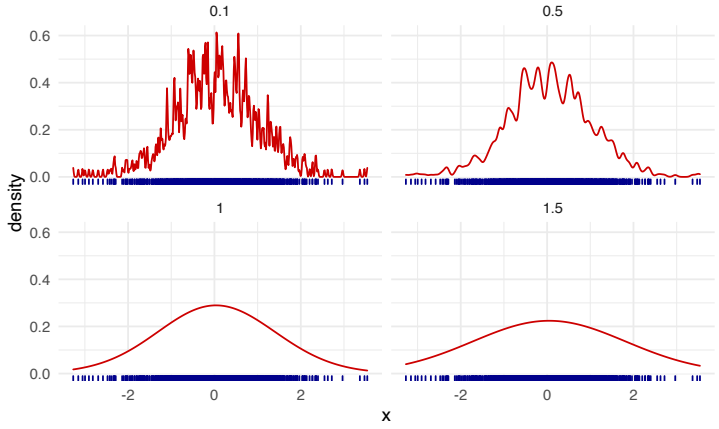
donde $x_1, \ldots, x_n$ es una muestra i.i.d. de f,

$$K_h(\cdot) = \frac{1}{h}K(\cdot/h).$$

y K es un kernel según las 4 propiedades anteriores.

Nota: . ¿Qué pasaría si modificamos el ancho de banda h para un mismo kernel?

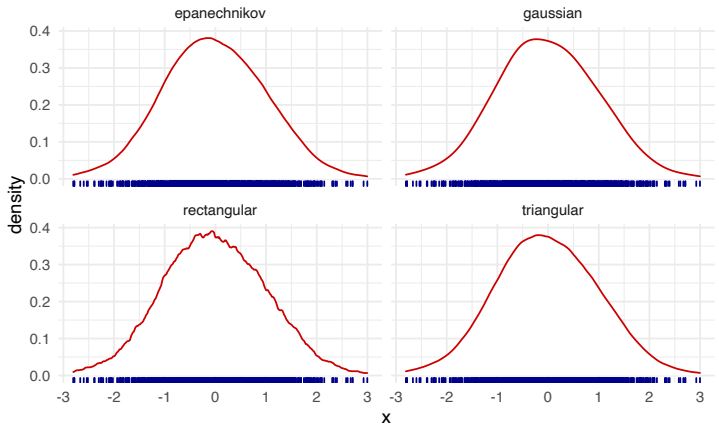
Nuevamente controlaríamos la suavidad del estimador a como se ilustra a continuación:



Inconveniente: no tenemos aún un criterio para un h óptimo.

Nota: . ¿Qué pasaría si modificamos el kernel para un mismo ancho de banda h?

Usando 1000 números aleatorios según una normal estándar, con un ancho de banda fijo ($h = 0{,}3$) podemos ver que no hay diferencias muy marcadas entre los estimadores por kernel:



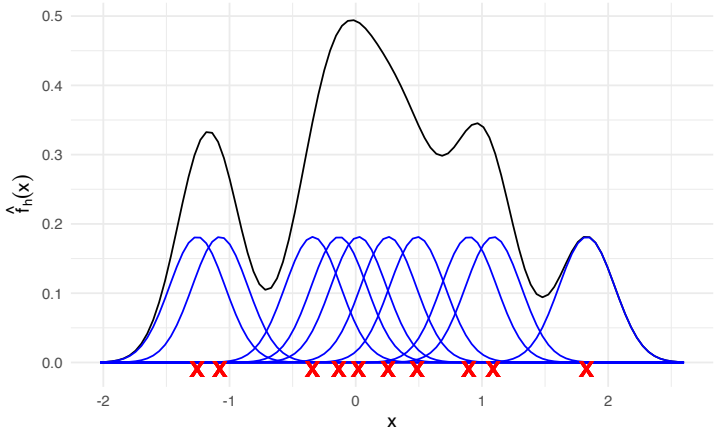
Recordemos nuevamente la fórmula

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

Cada sumando de esta expresión es una función de la variable x. Si la integramos se obtiene que

$$\frac{1}{nh}\int K\left(\frac{x - X_i}{h}\right)dx = \frac{1}{nh}\int K(u)\,hdu = \frac{1}{n}\int K(u)du = \frac{1}{n}$$

En el siguiente gráfico se generan 10 puntos aleatorios según una normal estándar (rojo) y se grafica cada uno de los 10 componentes del estimador de la densidad usando kernels gaussianos (azul). El estimador resultante aparece en color negro. Note que cada uno de los 10 componentes tiene la misma área bajo la curva, la cual en este caso es 0.1.



2.2.3. Propiedades Estadísticas

Al igual que en el caso de histograma, también aplica lo siguiente:

$$\begin{aligned}
\text{MSE}(\hat{f}_h(x)) &= \text{Var}(\hat{f}_h(x)) + \text{Sesgo}^2(\hat{f}_h(x)) \\
\text{MISE}(\hat{f}_h) &= \int \text{Var}(\hat{f}_h(x))dx + \int \text{Sesgo}^2(\hat{f}_h(x))dx
\end{aligned}$$

donde

$\text{Var}\left(\hat{f}_h(x)\right) = \mathbb{E}\left[\hat{f}_h(x) - \mathbb{E}\hat{f}_h(x)\right]^2$ and $\text{Sesgo}\left(\hat{f}_h(x)\right) = \mathbb{E}\left[\hat{f}_h(x)\right] - f(x)$.

En el caso de la varianza:

$$\begin{aligned}
\text{Var}(\hat{f}_h(x)) &= \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\right) \\
&= \frac{1}{n^2h^2}\sum_{i=1}^{n} \text{Var}\left(K\left(\frac{x - X_i}{h}\right)\right) \\
&= \frac{1}{nh^2}\text{Var}\left(K\left(\frac{x - X}{h}\right)\right) \\
&= \frac{1}{nh^2}\left\{\mathbb{E}\left[K^2\left(\frac{x - X}{h}\right)\right] - \left\{\mathbb{E}\left[K\left(\frac{x - X}{h}\right)\right]\right\}^2\right\}.
\end{aligned}$$

Usando que:

$$\begin{aligned}
\mathbb{E}\left[K^2\left(\frac{x - X}{h}\right)\right] &= \int K^2\left(\frac{x - s}{h}\right) f(s)ds \\
&= h\int K^2(u)\, f(uh + x)du \\
&= h\int K^2(u)\left\{f(x) + o(1)\right\}du \\
&= h\left\{\|K\|_2^2 f(x) + o(1)\right\}.
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left[K\left(\frac{x - X}{h}\right)\right] &= \int K\left(\frac{x - s}{h}\right) f(s)ds \\
&= h\int K(u)\, f(uh + x)du \\
&= h\int K(u)\left\{f(x) + o(1)\right\}du \\
&= h\left\{f(x) + o(1)\right\}.
\end{aligned}$$

Por lo tanto se obtiene que

$$\text{Var}\left(\hat{f}_h(x)\right) = \frac{1}{nh}\|K\|_2^2 f(x) + o\left(\frac{1}{nh}\right), \text{ si } nh \to \infty.$$

2.2.4. Sesgo

Para el sesgo tenemos

$$\begin{aligned}\text{Sesgo}\left(\hat{f}_h(x)\right) &= \mathbb{E}\left[\hat{f}_h(x)\right] - f(x) \\ &= \frac{1}{nh}\sum_{i=1}^{n}\mathrm{E}\left[K\left(\frac{x-X_i}{h}\right)\right] - f(x) \\ &= \frac{1}{h}\mathrm{E}\left[K\left(\frac{x-X_1}{h}\right)\right] - f(x) \\ &= \int \frac{1}{h}K\left(\frac{x-u}{h}\right)f(u)du - f(x)\end{aligned}$$

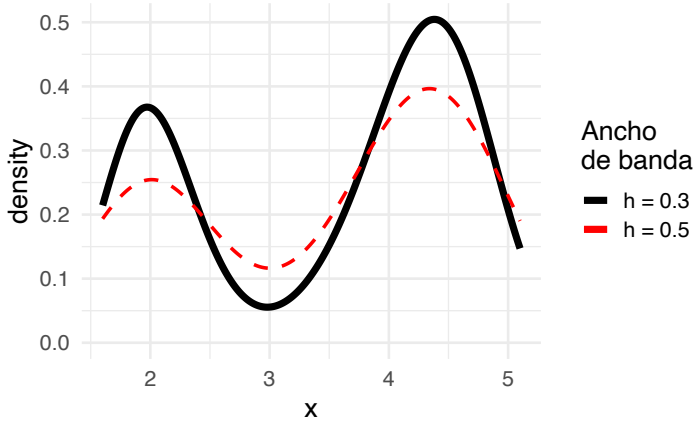
Ejercicio 2.5. Usando el cambio de variable $s = \frac{u-x}{h}$ y las propiedades del kernel pruebe que

$$\text{Sesgo}\left(\hat{f}_h(x)\right) = \frac{h^2}{2}f''\mu_2(K) + o(h^2), \text{ si } h \to 0$$

donde $\mu_2 = \int s^2K(s)ds$.

Nota: . En algunas pruebas más formales, se necesita además que f'' sea absolutamente continua y que $\int(f'''(x))dx < \infty$.}

En el siguiente gráfico se ilustra el estimador no paramétrico de la distribución de tiempos entre erupciones en la muy conocida tabla de datos *faithful*. El estimador se calcula bajo dos distintas escogencias de ancho de banda.



Nota: . Note como los cambios en el ancho de banda modifican la suavidad (sesgo) y el aplanamiento de la curva (varianza).

2.2.5. Error cuadrático medio y Error cuadrático medio integrado

El error cuadrático medio se escribe

$$\begin{aligned}\text{MSE}(\hat{f}_h(x)) &= \text{Sesgo}\left(\hat{f}_h(x)\right)^2 + \text{Var}\left(\hat{f}_h(x)\right) \\ &= \frac{h^4}{4}\left(\mu_2(K)f''(x)\right)^2 + \frac{1}{nh}\|K\|_2^2 f(x) + o(h^4) + o\left(\frac{1}{nh}\right).\end{aligned}$$

Y el error cuadrático medio integrado se escribe como,

$$\begin{aligned}\text{MISE}\left(\hat{f}_h\right) &= \int \text{MSE}\left(\hat{f}_h(x)\right) dx \\ &= \int \text{Sesgo}\left(\hat{f}_h(x)\right)^2 + \text{Var}\left(\hat{f}_h(x)\right) dx \\ &= \frac{h^4}{4}\mu_2^2(K)\left\|f''(x)\right\|_2^2 + \frac{1}{nh}\|K\|_2^2 + o(h^4) + o\left(\frac{1}{nh}\right).\end{aligned}$$

Al igual que en el caso del histograma, el estimador por kernels es un estimador consistente de f si $h \to 0$ y $nh \to \infty$. Además el MISE depende directamente de f''.

2.2.6. Ancho de banda óptimo

Minimizando el MISE con respecto a h obtenemos

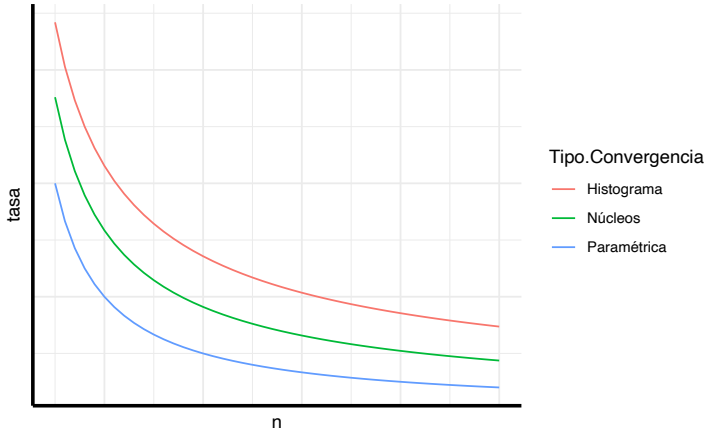
$$h_{opt} = \left(\frac{\|K\|_2^2}{\|f''\|_2^2 \left(\mu_2(K)\right)^2 n} \right)^{1/5} = O\left(n^{-1/5}\right).$$

Nota: . De forma práctica, h_{opt} no es un estimador útil de h porque depende de $\|f''\|_2^2$ que es desconocido. Más adelante veremos otra forma de encontrar este estimador.

Evaluando h_{opt} en el MISE tenemos que

$$\text{MISE}(\hat{f}_h) = \frac{5}{4} \left(\|K\|_2^2\right)^{4/5} \left(\|f''\|_2^2 \mu_2(K)\right)^{2/5} n^{-4/5} = O\left(n^{-4/5}\right).$$

y por lo tanto la tasa de convergencia del MISE a 0 es más rápida que para el caso del histograma:



Nota: . Como se comentó anteriormente, el principal inconveniente del ancho de banda:

$$h_{opt} = \left(\frac{\|K\|_2^2}{\|f''\|_2^2 \left(\mu_2(K)\right)^2 n} \right)^{1/5} = O\left(n^{-1/5}\right).$$

es que depende de f''.

A continuación se explica dos posibles métodos para determinar para aproximar el ancho de banda óptimo:

2.2.6.1. Referencia normal

Nota: . Este método es más efectivo si se conoce que la verdadera distribución es bastante suave, unimodal y simétrica. Más adelante veremos otro método para densidades más generales.

Asuma que f es normal distribuida y se utiliza un kernel K gausiano. Entonces se tiene que

$$\begin{aligned}\hat{h}_{rn} &= \left(\frac{\|K\|_2^2}{\|f''\|_2^2 \left(\mu_2(K)\right)^2 n}\right)^{1/5} = O\left(n^{-1/5}\right) \\ &= 1{,}06\hat{\sigma}n^{-1/5}.\end{aligned}$$

donde

$$\hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}^2)}$$

Ejercicio 2.6. Pruebe que la ecuación anterior es verdadera. Utilice el hecho de que:

$$\|f''\|_2^2 = \sigma^{-5}\int \phi''(x)^2 dx$$

donde ϕ es la función de densidad de una $N(0,1)$.

Nota: . El principal inconveniente de $\hat{h}_{rn}$ es su sensibilidad a los valores extremos:

Ejemplo 2.1. La varianza empírica de 1, 2, 3, 4, 5, es 2.5.

La varianza empírica de 1, 2, 3, 4, 5, 99, es 1538.

Para solucionar el problema anterior, se puede considerar una medida más robusta de variación, por ejemplo el rango intercuantil IQR:

$$\mathrm{IQR}^X = Q_3^X - Q_1^X$$

donde Q_1^X y Q_3^X son el primer y tercer cuartil de un conjunto de datos $X_1, \ldots, X_n$.

Con el supuesto que $X \sim \mathcal{N}(\mu, \sigma^2)$ entonces $Z = \dfrac{X-\mu}{\sigma} \sim \mathcal{N}(0,1)$ y entonces:

$$\begin{aligned}
\text{IQR} &= Q_3^X - Q_1^X \\
&= \left(\mu + \sigma Q_3^Z\right) - \left(\mu + \sigma Q_1^Z\right) \\
&= \sigma\left(Q_3^Z - Q_1^Z\right) \\
&\approx \sigma\left(0{,}67 - (-0{,}67)\right) \\
&= 1{,}34\sigma.
\end{aligned}$$

Por lo tanto $\hat{\sigma} = \dfrac{\widehat{\text{IQR}}^X}{1{,}34}$

Podemos sustituir la varianza empírica de la fórmula inicial y tenemos

$$\hat{h}_{rn} = 1{,}06\frac{\widehat{\text{IQR}}^X}{1{,}34}n^{-\frac{1}{5}} \approx 0{,}79\ \widehat{\text{IQR}}^X\ n^{-\frac{1}{5}}$$

Combinando ambos estimadores, podemos obtener,

$$\hat{h}_{rn} = 1{,}06\,\text{mín}\left\{\frac{\widehat{\text{IQR}}^X}{1{,}34}, \hat{\sigma}\right\} n^{-\frac{1}{5}}$$

pero esta aproximación es conveniente bajo el escenario de que la densidad f sea similar a una densidad normal.

2.2.6.2. Validación Cruzada

Defina el *error cuadrático integrado* como

$$\begin{aligned}
\text{ISE}(\hat{f}_h) &= \int \left(\hat{f}_h(x) - f(x)\right)^2 dx \\
&= \int \hat{f}_h^2(x)dx - 2\int \hat{f}_h(x)f(x)dx + \int f^2(x)dx.
\end{aligned}$$

Nota: . El MISE es el valor esperado del ISE.

Nuestro objetivo es minimizar el ISE con respecto a h.

Primero note que $\int f^2(x)dx$ NO DEPENDE de h. Podemos minimizar la expresión

$$\text{ISE}(\hat{f}_h) - \int f^2(x)dx = \int \hat{f}_h^2(x)dx - 2\int \hat{f}_h(x)f(x)dx$$

Vamos a resolver esto en dos pasos partes

Integral $\int \hat{f}_h(x)f(x)dx$

Integral $\int \hat{f}_h(x)f(x)dx$

El término $\int \hat{f}_h(x)f(x)dx$ es el valor esperado de $\text{E}\left[\hat{f}_h(X)\right]$. Su estimador empírico sería:

$$\widehat{\text{E}\left[\hat{f}_h(X)\right]} = \frac{1}{n}\sum_{i=1}^{n}\hat{f}_h(X_i) = \frac{1}{n^2h}\sum_{i=1}^{n}\sum_{j=1}^{n}K\left(\frac{X_j - X_i}{h}\right).$$

Nota: . El problema con esta expresión es que las observaciones que se usan para estimar la esperanza son las mismas que se usan para estimar $\hat{f}_h(x)$ (Se utilizan doble).

La solución es remover la $i^{\text{ésima}}$ observación de $\hat{f}_h$ para cada i.

Redefiniendo el estimador anterior tenemos una estimación de $\int \hat{f}_h(x)f(x)dx$ a través de:

$$\frac{1}{n}\sum_{i=1}^{n}\hat{f}_{h,-i}(X_i),$$

donde (estimador *leave-one-out*)

$$\hat{f}_{h,-i}(x) = \frac{1}{(n-1)h}\sum_{\substack{j=1\\ j\neq i}}^{n}K\left(\frac{x - X_j}{h}\right).$$

de esta forma nos aseguramos que las observaciones que se usan para calcular $\hat{f}_{h,-i}(x)$ son independientes de la observación que uno usa para definir el estimador de $E[\hat{f}_h(x)]$.

Siguiendo con el término $\int \hat{f}_h^2(x)dx$ note que este se puede reescribir como

$$\begin{aligned}\int \hat{f}_h^2(x)dx &= \int \left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)\right)^2 dx \\ &= \frac{1}{n^2h^2}\sum_{i=1}^{n}\sum_{i=1}^{n}\int K\left(\frac{x-X_i}{h}\right)K\left(\frac{x-X_j}{h}\right)dx \\ &= \frac{1}{n^2h}\sum_{i=1}^{n}\sum_{i=1}^{n}\int K(u)\,K\left(\frac{X_i-X_j}{h}-u\right)du \\ &= \frac{1}{n^2h}\sum_{i=1}^{n}\sum_{i=1}^{n} K*K\left(\frac{X_i-X_j}{h}\right).\end{aligned}$$

donde $K * K$ es la convolución de K consigo misma.

Finalmente tenemos la función,

Finalmente definimos la función objetivo del criterio de validación cruzada como:

$$\mathrm{CV}(h) = \frac{1}{n^2h}\sum_{i=1}^{n}\sum_{j=1}^{n} K*K\left(\frac{X_i-X_j}{h}\right) - \frac{2}{n(n-1)h}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} K\left(\frac{X_i-X_j}{h}\right).$$

Nota: . Note que $\mathrm{CV}(h)$ no depende de f o sus derivadas y además la función objetivo se adapta automáticamente a las características de la densidad f.

2.2.7. Intervalos de confianza para estimadores de densidad no paramétricos

Usando los resultados anteriores y asumiendo que $h = cn^{-\frac{1}{5}}$ entonces

$$n^{\frac{2}{5}}\left\{\hat{f}_h(x) - f(x)\right\} \xrightarrow{\mathcal{L}} \mathcal{N}\left(\underbrace{\frac{c^2}{2}f''\mu_2(K)}_{b_x}, \underbrace{\frac{1}{c}f(x)\,\|K\|_2^2}_{v_x}\right).$$

Si $z_{1-\frac{\alpha}{2}}$ es el cuantil $1-\frac{\alpha}{2}$ de una distribución normal estándar, entonces

$$\begin{aligned}1-\alpha &\approx \mathbb{P}\left(b_x - z_{1-\frac{\alpha}{2}} v_x \leq n^{2/5}\left\{\widehat{f}_h(x) - f(x)\right\} \leq b_x + z_{1-\frac{\alpha}{2}} v_x\right) \\ &= \mathbb{P}\left(\widehat{f}_h(x) - n^{-2/5}\left\{b_x + z_{1-\frac{\alpha}{2}} v_x\right\}\right. \\ &\qquad \left. \leq f(x) \leq \hat{f}_h(x) - n^{-2/5}\left\{b_x - z_{1-\frac{\alpha}{2}} v_x\right\}\right)\end{aligned}$$

Esta expresión nos dice que con una probabilidad de $1-\alpha$ se tiene que

$$\left[\hat{f}_h(x) - \frac{h^2}{2} f''(x)\mu_2(K) - z_{1-\frac{\alpha}{2}}\sqrt{\frac{f(x)\|K\|_2^2}{nh}}\right.$$
$$\left.\widehat{f}_h(x) - \frac{h^2}{2} f''(x)\mu_2(K) + z_{1-\frac{\alpha}{2}}\sqrt{\frac{f(x)\|K\|_2^2}{nh}}\right]$$

Al igual que en los casos anteriores, este invtervalo no es útil ya que depende de $f(x)$ y $f''(x)$.

Si h es pequeño relativamente a $n^{-\frac{1}{5}}$ entonces el segundo término $\frac{h^2}{2} f''(x)\mu_2(K)$ podría ser ignorado.

Si h es pequeño relativamente a $n^{-\frac{1}{5}}$ entonces el segundo término $\frac{h^2}{2} f''(x)\mu_2(K)$ podría ser ignorado.

Podemos reemplazar $f(x)$ por su estimador $\hat{f}_h(x)$. Entonces tendríamos una intervalo aplicable a nuestro caso:

$$\left[\hat{f}_h(x) - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{f}_h(x)\|K\|_2^2}{nh}}, \hat{f}_h(x) + z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{f}_h(x)\|\mathrm{K}\|_2^2}{nh}}\right]$$

Nota: . Este intervalo de confianza está definido para x fijo y no permite hacer inferencia sobre toda la función f. Una forma de determinar la banda de confianza de toda la función f es a través de la fórmula 3.52 en la página 62 de (Härdle y col. 2004).

2.3. Laboratorio

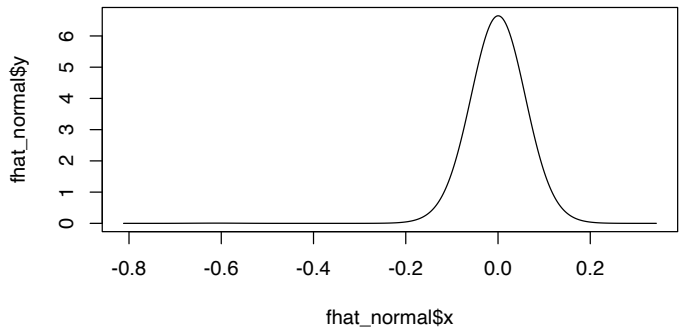
Comenzaremos con una librería bastante básica llamada `KernSmooth`.

```
x <- read.csv("data/stockres.txt")
x <- unlist(x)
```

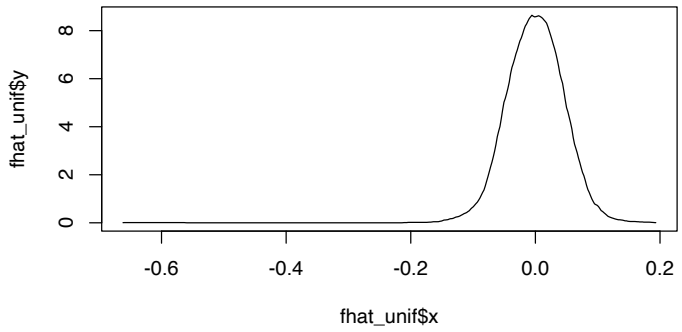
```
summary(x)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.6118200 -0.0204085 -0.0010632 -0.0004988  0.0215999  0.1432286
```

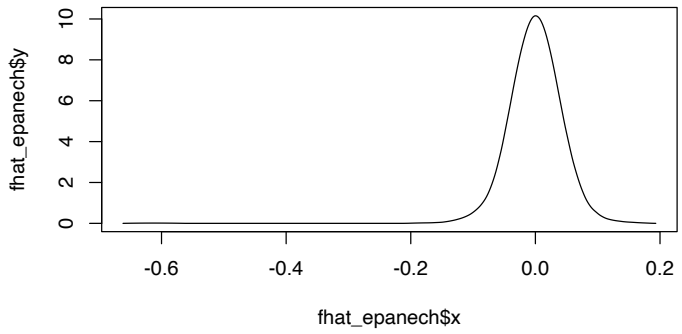
```
library(KernSmooth)

fhat_normal <- bkde(x, kernel = "normal", bandwidth = 0.05)
plot(fhat_normal, type = "l")
```



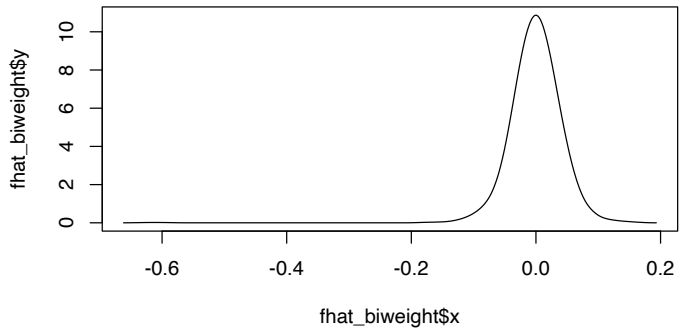

```
fhat_unif <- bkde(x, kernel = "box", bandwidth = 0.05)
plot(fhat_unif, type = "l")
```



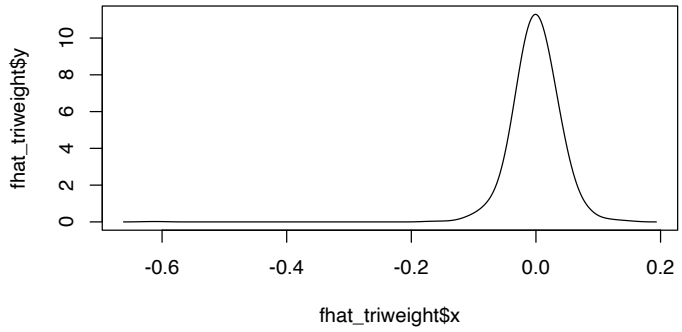

```
fhat_epanech <- bkde(x, kernel = "epanech", bandwidth = 0.05)
plot(fhat_epanech, type = "l")
```




```
fhat_biweight <- bkde(x, kernel = "biweight", bandwidth = 0.05)
plot(fhat_biweight, type = "l")
```



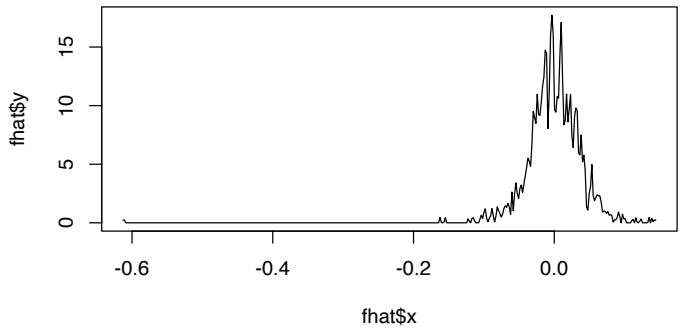

```
fhat_triweight <- bkde(x, kernel = "triweight", bandwidth = 0.05)
plot(fhat_triweight, type = "l")
```



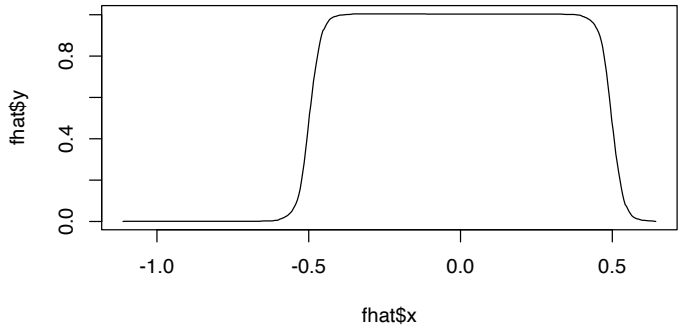
2.3.2. Efecto del ancho de banda en la estimación

** Kernel uniforme **

```
fhat <- bkde(x, kernel = "box", bandwidth = 0.001)
plot(fhat, type = "l")
```

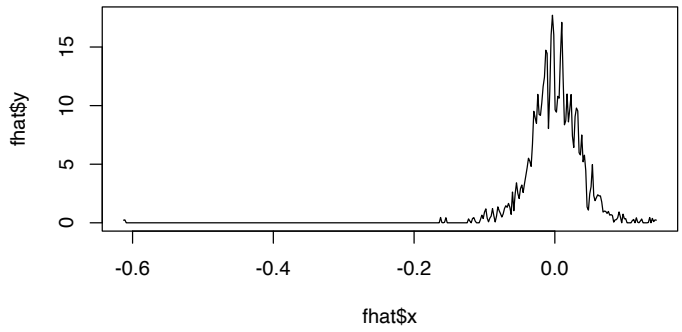



```
fhat <- bkde(x, kernel = "box", bandwidth = 0.5)
plot(fhat, type = "l")
```

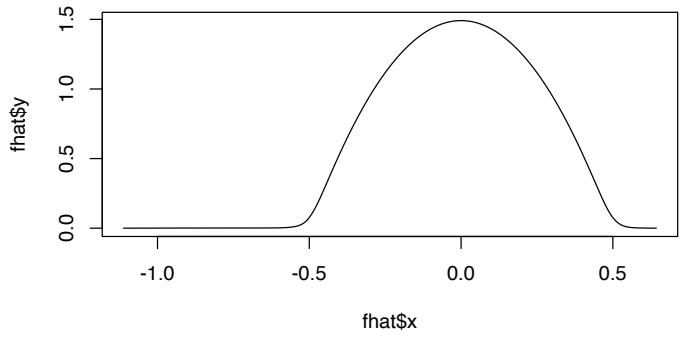


** Kernel Epanechnikov **

```
fhat <- bkde(x, kernel = "epa", bandwidth = 0.001)
plot(fhat, type = "l")
```




```
fhat <- bkde(x, kernel = "epa", bandwidth = 0.5)
plot(fhat, type = "l")
```




```
suppressMessages(library(tidyverse))
library(gganimate)

fani <- tibble()

for (b in seq(0.001, 0.02, length.out = 40)) {
    f <- bkde(x, kernel = "epa", bandwidth = b, gridsize = length(x))
    fani <- fani %>%
        bind_rows(tibble(xreal = sort(x), x = f$x,
            y = f$y, bw = b))
```

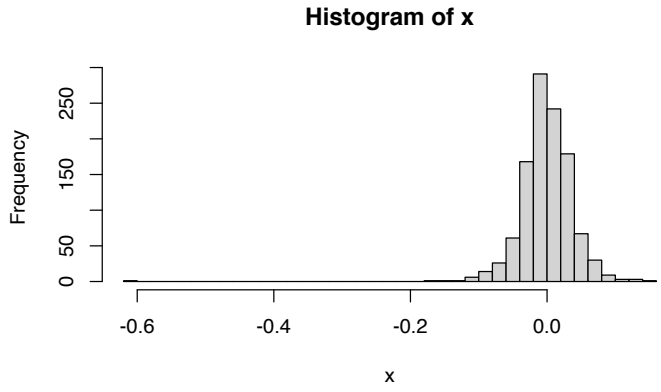
```
}

ggplot(data = fani) + geom_line(aes(x, y), color = "blue") +
    labs(title = paste0("Ancho de banda = {closest_state}")) +
    transition_states(bw) + view_follow() + theme_minimal(base_size = 20)

# anim_save('manual_figure/bandwidth-animation.gif')
```

Nota: .

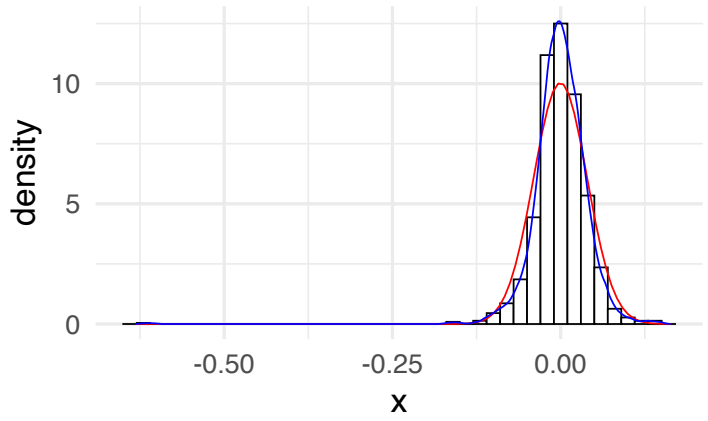
- Construya una variable llamada `u` que sea una secuencia de -0.15 a 0.15 con un paso de 0.01
- Asigne `x` a los datos `stockrel` y calcule su media y varianza.
- Usando la función `dnorm` construya los valores de la distribución de los datos usando la media y varianza calculada anteriormente. Asigne a esta variable `f\_param`.
- Defina un ancho de banda `h` en 0.02
- Construya un histograma para estos datos con ancho de banda `h`. Llame a esta variable `f\_hist`
- Usando el paquete `KernSmooth` y la función `bkde`, construya una función que calcule el estimador no paramétrico con un núcleo Epanechivok para un ancho de banda h. Llame a esta variable `f\_epa`.
- Dibuje en el mismo gráfico la estimación paramétrica y no paramétrica.

```
x <- read.csv("data/stockres.txt")
x <- unlist(x)
# Eliminar nombres de las columnas
names(x) <- NULL

u <- seq(-0.15, 0.15, by = 0.01)

mu <- mean(x)
sigma <- sd(x)

f_param <- dnorm(u, mean = mu, sd = sigma)

h <- 0.02

n_bins <- floor(diff(range(x))/h)
```

```
f_hist <- hist(x, breaks = n_bins)
```




```
f_epa <- as.data.frame(bkde(x, kernel = "epa", bandwidth = h))

x_df <- data.frame(x)

library(ggplot2)

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.02, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = f_epa,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```



2.3.3. Ancho de banda óptimo

Usemos la regla de la normal o también conocida como Silverman. **Primero recuerde que en este caso se asume que** $f(x)$ sigue una distribución normal. En este caso, lo que se obtiene es que

$$\begin{aligned}\|f''\|_2^2 &= \sigma^{-5}\int\{\phi''\}^2 dx \\ &= \sigma^{-5}\frac{3}{8\sqrt{\pi}} \approx 0{,}212\sigma^{-5}\end{aligned}$$

donde ϕ es la densidad de una normal estándar.

El estimador para σ es

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

Y usando el cálculo realizado anteriormente, se obtiene que

$$h_{normal} = \left(\frac{4s^5}{3n}\right)^{1/5} \approx 1{,}06sn^{-1/5}.$$

Un estimador más robusto es

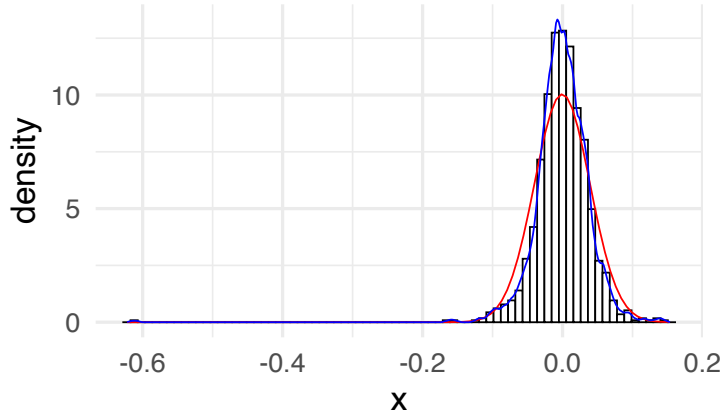
$$h_{normal} = 1{,}06\,\text{mín}\left\{s, \frac{IQR}{1{,}34}\right\}n^{-1/5}.$$

¿Por qué es $IQR/1{,}34$?

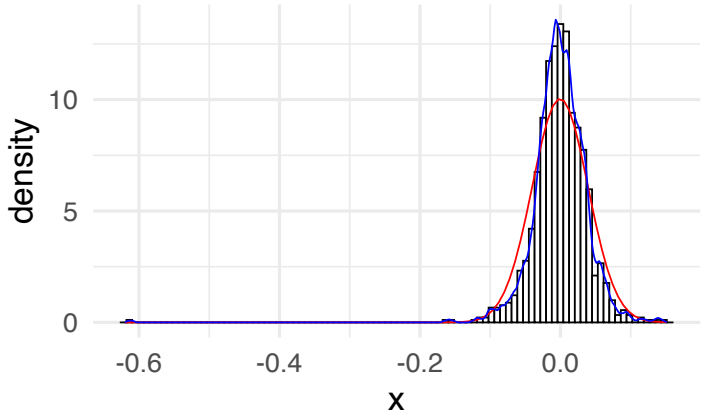
```
s <- sd(x)
n <- length(x)
```

```
h_normal <- 1.06 * s * n^(-1/5)

h <- h_normal
```

```
n_bins <- floor(diff(range(x))/h)
f_hist <- hist(x, breaks = n_bins, plot = FALSE)
f_epa <- as.data.frame(bkde(x, kernel = "epa", bandwidth = h))

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = h, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = f_epa,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```

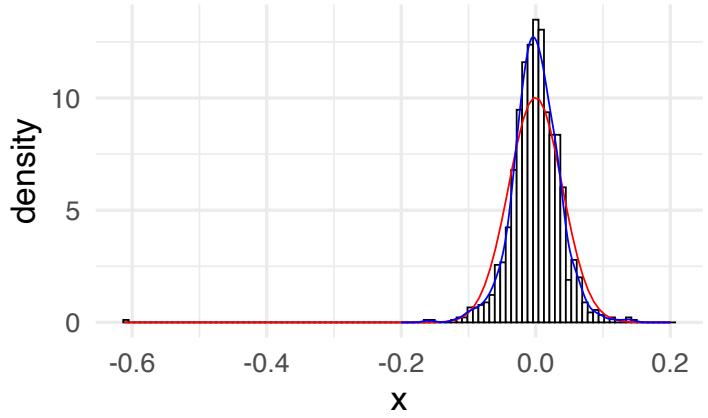



```
h_iqr <- 1.06 * min(s, IQR(x)/1.34) * n^(-1/5)

h <- h_iqr

n_bins <- floor(diff(range(x))/h)
f_hist <- hist(x, breaks = n_bins, plot = FALSE)
f_epa <- as.data.frame(bkde(x, kernel = "epa", bandwidth = h))

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = h, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = f_epa,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```



Una librería más especializada es `np` (non-parametric).

```
library(np)

x.eval <- seq(-0.2, 0.2, length.out = 200)

h_normal_np <- npudensbw(dat = x, bwmethod = "normal-reference")

dens.ksum <- npksum(txdat = x, exdat = x.eval, bws = h_normal_np$bw)$ksum/(n *
    h_normal_np$bw[1])

dens.ksum.df <- data.frame(x = x.eval, y = dens.ksum)

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = h_normal_np$bw, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = dens.ksum.df,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```



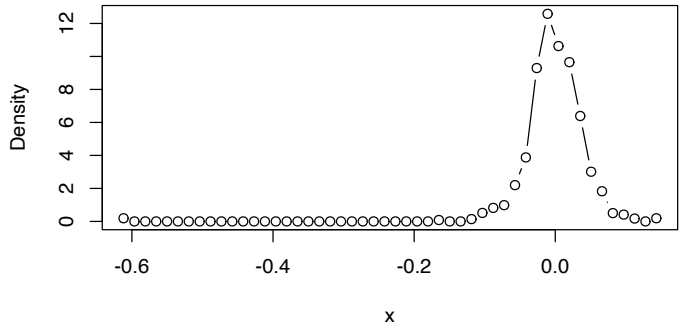
2.3.4. Validación cruzada

La forma que vimos en clase es la de validación cruzada por mínimos cuadrados"least-square cross validation'' la cual se puede ejecutar con este comando.

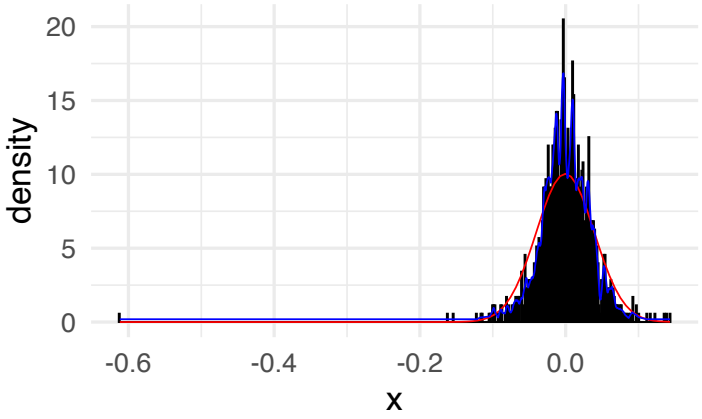
```
h_cv_np_ls <- npudensbw(dat = x, bwmethod = "cv.ls",
    ckertype = "epa", ckerorder = 2)
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1
```

```
dens.np <- npudens(h_cv_np_ls)

plot(dens.np, type = "b")
```



```
dens.np.df <- data.frame(x = dens.np$eval[, 1], y = dens.np$dens)

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = h_cv_np_ls$bw, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = dens.np.df,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```



2.3.5. Temas adicionales

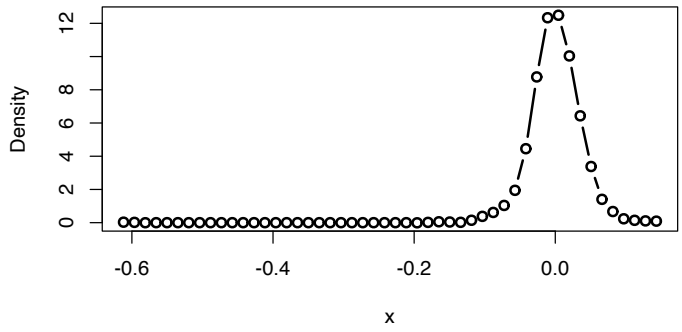
** Reducción del sesgo ** Como lo mencionamos en el texto, una forma de mejorar el sesgo en la estimación es suponer que la función de densidad es más veces diferenciable.

Esto se logra asumiendo que el Kernel es más veces diferenciable.

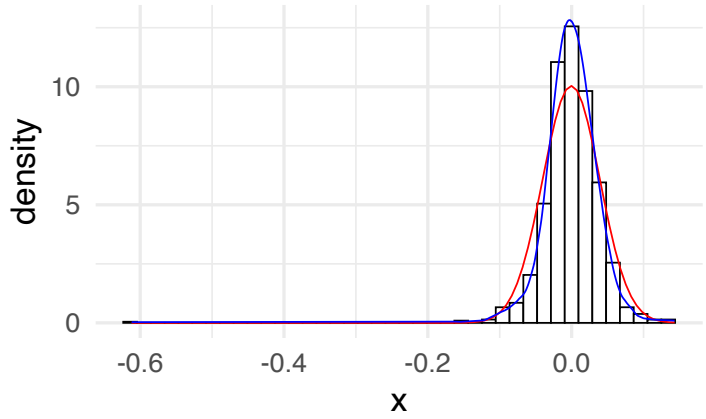
```
h_cv_np_ls <- npudensbw(dat = x, bwmethod = "cv.ls",
    ckertype = "epa", ckerorder = 4)
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 /Multis
```

```
dens.np <- npudens(h_cv_np_ls)

plot(dens.np, type = "b", lwd = 2)
```




```
dens.np.df <- data.frame(x = dens.np$eval[, 1], y = dens.np$dens)

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = h_cv_np_ls$bw, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = dens.np.df,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```

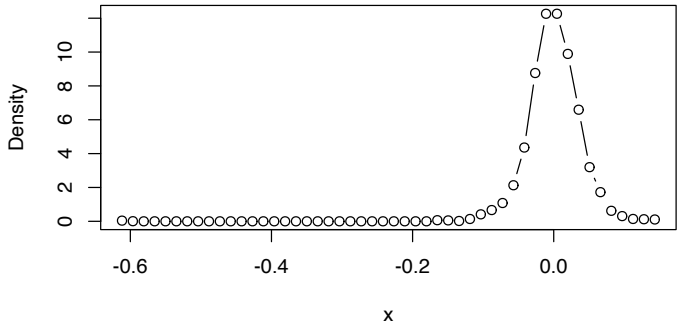


Otra forma de estimar el ancho de banda Otra forma de estimar ancho de bandas óptimos es usando máxima verosimilitud. Les dejo de tarea revisar la sección 1.1 del artículo de (Hall 1987) para entender su estructura.

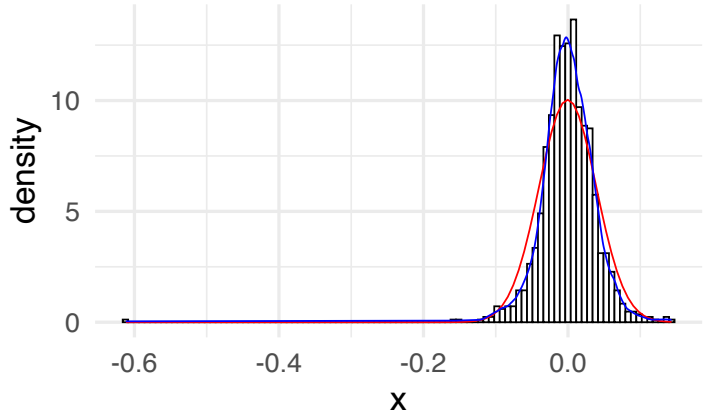
```
h_cv_np_ml <- npudensbw(dat = x, bwmethod = "cv.ml",
    ckertype = "epanechnikov")
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 /Multis
```

```
dens.np <- npudens(h_cv_np_ml)

plot(dens.np, type = "b")
```



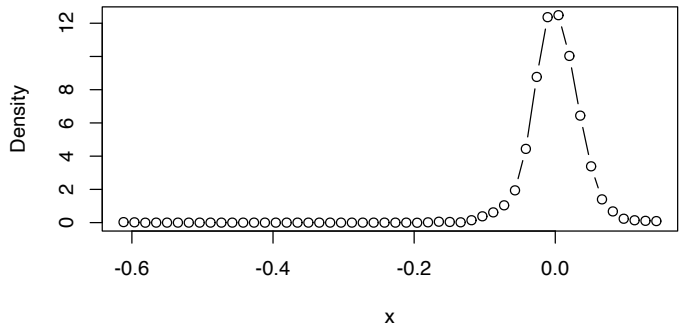

```
dens.np.df <- data.frame(x = dens.np$eval[, 1], y = dens.np$dens)

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = h_cv_np_ml$bw, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = dens.np.df,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```



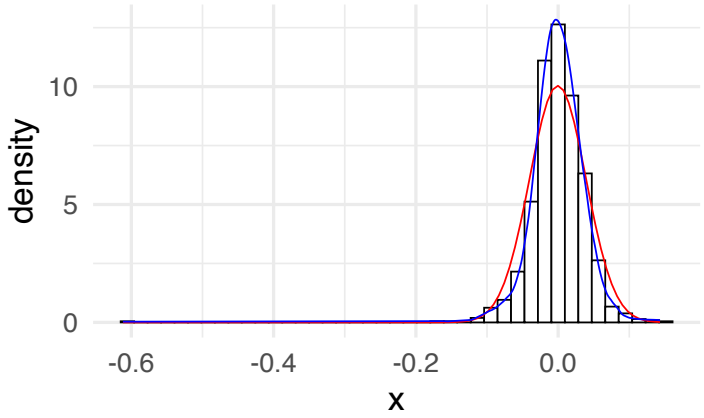
```
h_cv_np_ml <- npudensbw(dat = x, bwmethod = "cv.ml",
    ckertype = "epanechnikov", ckerorder = 4)
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1
```

```
dens.np <- npudens(h_cv_np_ml)

plot(dens.np, type = "b")
```




```
dens.np.df <- data.frame(x = dens.np$eval[, 1], y = dens.np$dens)

ggplot(x_df, aes(x)) + geom_histogram(aes(y = ..density..),
    binwidth = h_cv_np_ml$bw, col = "black", fill = "white") +
    stat_function(fun = dnorm, args = list(mean = mu,
        sd = sigma), color = "red") + geom_line(data = dens.np.df,
    aes(x, y), color = "blue") + theme_minimal(base_size = 20)
```




```
fani <- tibble()

for (b in seq(0.001, 0.05, length.out = 40)) {
    f <- npudens(tdat = x, ckertype = "epanechnikov",
        bandwidth.compute = FALSE, bws = b)
    fani <- fani %>%
        bind_rows(tibble(xreal = sort(x), x = f$eval$x,
            y = f$dens, bw = b))
}

ggplot(data = fani) + geom_line(aes(x, y), color = "blue") +
    labs(title = paste0("Ancho de banda = {closest_state}")) +
    theme_minimal(base_size = 20) + transition_states(bw) +
    view_follow()

# anim_save('manual_figure/bandwidth-animation-np.gif')
```

Ejercicio 2.7. Implementar el intervalo confianza visto en clase para estimadores de densidades por núcleos y visualizarlo de en ggplot.

Si se atreven: ¿Se podría hacer una versión animada de ese gráfico para visualizar el significado real de este el intervalo de confianza?

2.4. Ejercicios

Del libro de (Härdle y col. 2004) hagan los siguientes ejercicios

1. **Sección 2:** 1, 2, 3, 5, 7, 14
2. **Sección 3:** 4, 8, 10, 11, 16,

Capítulo 3

Jackknife y Bootstrap

Suponga que se quiere estimar un intervalo de confianza para la media μ desconocida de un conjunto de datos $X_1, \ldots, X_n$ que tiene distribución $\mathcal{N}(\mu, \sigma^2)$.

Primero se conoce que

$$\sqrt{n}\,(\hat{\mu} - \mu) \sim \mathcal{N}\left(0, \sigma^2\right),$$

y esto nos permite escribir el intervalo de confianza como

$$\left[\hat{\mu} - \hat{\sigma} z_{1-\frac{\alpha}{2}}, \hat{\mu} + \hat{\sigma} z_{1-\frac{\alpha}{2}}\right]$$

donde $z_{1-\frac{\alpha}{2}}$ es el cuantil $1 - \frac{\alpha}{2}$ de una normal estándar.

La expresión anterior es posible dado que la distribución de $\hat{\mu}$ es normal.

Nota: . ¿Qué pasaría si no conocemos la distribución de $\hat{\mu}$?

¿Cómo podemos encontrar ese intervalo de confianza?

3.1. Caso concreto

Suponga que tenemos la siguiente tabla de datos, que representa una muestra de tiempos y distancias de viajes en Atlanta.

Cargamos la base de la siguiente forma:

```
CommuteAtlanta <- read.csv2("data/CommuteAtlanta.csv")
```

City	Age	Distance	Time	Sex
Atlanta	19	10	15	M
Atlanta	55	45	60	M
Atlanta	48	12	45	M
Atlanta	45	4	10	F
Atlanta	48	15	30	F
Atlanta	43	33	60	M

Para este ejemplo tomaremos la variable `Time` que la llamaremos `x` para ser más breves. En este caso note que

```
x <- CommuteAtlanta$Time
```

La media es 29.11 y su varianza 429.2483968. Para efectos de lo que sigue, asignaremos la varianza a la variable T_n

```
Tn <- var(x)
```

A partir de estos dos valores, ¿Cuál sería un intervalo de confianza para la varianza?

Note que esta pregunta es difícil ya que no tenemos ningún tipo de información adicional para inferir la variación de la varianza T_n.

Las dos técnicas que veremos a continuación nos permitirán extraer *información adicional* de la muestra para inferir propiedades distribucionales de T_n.

Nota: . Para efectos de este capítulo, llamaremos $T_n = T(X_1, \ldots, X_n)$ al estadístico T formado por la muestra de los X_i's.

3.2. Jackknife

Esta técnica fue propuesta por (Quenouille 1949). Primero que todo se puede probar que existen estimadores que cumplen la siguiente propiedad:

$$\text{Sesgo}(T_n) = \frac{a}{n} + \frac{b}{n^2} + O\left(\frac{1}{n^3}\right) \tag{3.1}$$

para algún a and b.

Por ejemplo sea $\sigma^2 = \text{Var}(X_i)$ y sea $\widehat{\sigma}_n^2 = n^{-1}\sum_{i=1}^n (X_i - \bar{X})^2$. Entonces,

$$\mathbb{E}\left(\widehat{\sigma}_n^2\right) = \frac{n-1}{n}\sigma^2$$

por lo tanto

$$\text{Sesgo} = -\frac{\sigma^2}{n}$$

Por lo tanto en este caso $a = -\sigma^2$ y $b = 0$.

Defina $T_{(-i)}$ como el estimador T_n pero eliminando el i-ésimo elemento de la muestra.

Es claro que en este contexto, se tiene que

$$\text{Sesgo}\left(T_{(-i)}\right) = \frac{a}{n-1} + \frac{b}{(n-1)^2} + O\left(\frac{1}{(n-1)^3}\right) \tag{3.2}$$

Ejercicio 3.1. Una forma fácil de construir los $T_{(-i)}$ es primero replicando la matriz de datos múltiple veces usando el producto de kronecker

```
n <- length(x)
jackdf <- kronecker(matrix(1, 1, n), x)
```

15	15	15	15	15	15	15	15	15	15
60	60	60	60	60	60	60	60	60	60
45	45	45	45	45	45	45	45	45	45
10	10	10	10	10	10	10	10	10	10
30	30	30	30	30	30	30	30	30	30
60	60	60	60	60	60	60	60	60	60
45	45	45	45	45	45	45	45	45	45
10	10	10	10	10	10	10	10	10	10
25	25	25	25	25	25	25	25	25	25
15	15	15	15	15	15	15	15	15	15

Y luego se elimina la diagonal

```
diag(jackdf) <- NA
```

NA	15	15	15	15	15	15	15	15	15
60	NA	60	60	60	60	60	60	60	60
45	45	NA	45	45	45	45	45	45	45
10	10	10	NA	10	10	10	10	10	10
30	30	30	30	NA	30	30	30	30	30
60	60	60	60	60	NA	60	60	60	60
45	45	45	45	45	45	NA	45	45	45
10	10	10	10	10	10	10	NA	10	10
25	25	25	25	25	25	25	25	NA	25
15	15	15	15	15	15	15	15	15	NA

Cada columna contiene toda la muestra excepto el i-ésimo elemento. Solo basta estimar la media de cada columna:

```
T_i <- apply(jackdf, 2, var, na.rm = TRUE)
```

x
429.7098
428.1905
429.6023
429.3756
430.1087
428.1905
429.6023
429.3756
430.0764
429.7098

Definimos el estimador de sesgo *jackknife* de T_n como

$$b_{jack} = (n-1)(\overline{T}_n - T_n)$$

donde

$$\overline{T}_n = \frac{1}{n}\sum_{i=1}^{n} T_{(-i)}$$

y el estimador corregido por sesgo es: $T_{jack} = T_n - b_{jack}$. ::: {.exercise

#unnamed-chunk-74} En nuestro caso tendríamos lo siguiente: :::

```
(bjack <- (n - 1) * (mean(T_i) - Tn))
```

```
## [1] 0
```

Es decir, el sesgo aproximado (jackknife) del estimador T_n es 0.

Si se asume que T_n es un estimador del parámetro θ entonce se puede comprobar que b_{jack} cumple:

$$\begin{aligned}
\mathbb{E}\left(b_{\text{jack}}\right) &= (n-1)\left(\mathbb{E}\left[\overline{T}_n\right] - \mathbb{E}\left[T_n\right]\right) \\
&= (n-1)\left(\mathbb{E}\left[\overline{T}_n\right] - \theta + \theta - \mathbb{E}\left[T_n\right]\right) \\
&= (n-1)\left(\text{Sesgo}\left(\overline{T}_n\right) - \text{Sesgo}\left(T_n\right)\right) \\
&= (n-1)\left[\left(\frac{1}{n-1} - \frac{1}{n}\right)a + \left(\frac{1}{(n-1)^2} - \frac{1}{n^2}\right)b + O\left(\frac{1}{n^3}\right)\right] \\
&= \frac{a}{n} + \frac{(2n-1)b}{n^2(n-1)} + O\left(\frac{1}{n^2}\right) \\
&= \text{Sesgo}\left(T_n\right) + O\left(\frac{1}{n^2}\right)
\end{aligned}$$

Nota: . Es decir, en general, el estimador b_{jack} aproxima correctamente $\text{Sesgo}\left(T_n\right)$ hasta con un error del n^{-2}.

Podemos usar los T_i para generar muestras adicionales para estimar el parámetro θ a través del siguiente estimador:

$$\widetilde{T}_i = nT_n - (n-1)T_{(-i)}.$$

Nota: . A $\widetilde{T}_i$ se le llaman **pseudo-valor** y representa el aporte o peso que tiene la variable X_i para estimar T_n.

Ejercicio 3.2. Usado un cálculo similar para el b_{jack} pruebe que

$$\text{Sesgo}\left(T_{\text{jack}}\right) = -\frac{b}{n(n-1)} + O\left(\frac{1}{n^2}\right) = O\left(\frac{1}{n^2}\right).$$

¿Qué conclusión se obtiene de este cálculo?

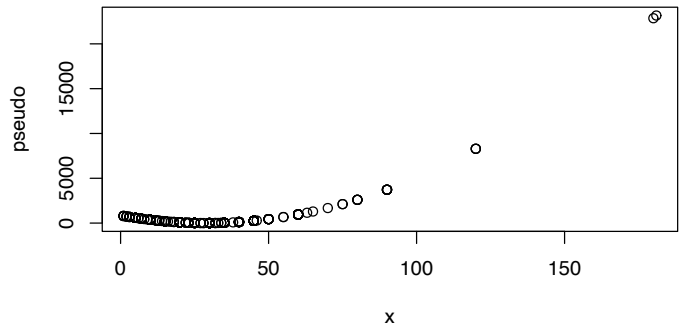
Ejercicio 3.3. Los pseudo-valores se estiman de forma directa como,

```
pseudo <- n * Tn - (n - 1) * T_i

pseudo[1:10]
```

```
##  [1] 199.02972209 957.16225222 252.64417993 365.79679037  -0.06666345
##  [6] 957.16225222 252.64417993 365.79679037  16.09799519 199.02972209
```

Lo importante acá es notar la asociación o correspondencia que tiene con los datos reales,

```
plot(x = x, y = pseudo)
```



Con estos pseudo-valores, es posible estimar la media y la varianza de T_n con los siguientes estimadores respectivos:

$$T_{\text{jack}} = \frac{1}{n}\sum_{i=1}^{n}\widetilde{T}_i$$

y

$$v_{jack} = \frac{\sum_{i=1}^{n}\left(\widetilde{T}_i - \frac{1}{n}\sum_{i=1}^{n}\widetilde{T}_i\right)^2}{n-1}.$$

Nota: . Sin embargo, se puede demostrar fácilmente que se pueden usar pseudovalores para construir una prueba normal de hipótesis.

Como los pseudovalores son idénticamente distribuidos entonces su promedio se ajusta de forma aproximada a una distribución normal a medida que el tamaño de la muestra aumenta. Por lo tanto, tenemos que

$$\frac{\sqrt{n}\left(T_{jack}-\theta\right)}{\sqrt{v_{jack}}} \to N(0,1).$$

```
(Tjack <- mean(pseudo))
```

```
## [1] 429.2484
```

```
(Vjack <- var(pseudo, na.rm = TRUE))
```

```
## [1] 2701991
```

```
(sdjack <- sqrt(Vjack))
```

```
## [1] 1643.774
```

```
(z <- qnorm(1 - 0.05/2))
```

```
## [1] 1.959964
```

```
c(Tjack - z * sdjack/sqrt(n), Tjack + z * sdjack/sqrt(n))
```

```
## [1] 285.1679 573.3289
```

3.3. Bootstrap

Este método es un poco más sencillo de implementar que Jackknife y es igualmente de eficaz. Este fue propuesto por Bradley Efron en (Efron 1979).

Primero recordemos que estamos estimando la variabilidad propia de un estadístico a partir de una muestra. Asuma que este estadístico tiene la forma $T_n = g\left(X_1, \ldots, X_n\right)$ donde g es cualquier función (media, varianza, quantiles, etc).

Supongamos que conocemos la distribución real de los X's, llamada $F(x)$ y asumamos que $T_n = \bar{X}_n$. Si uno quisiera estimar la varianza de T_n basta con hacer

$$\mathbb{V}_F\left(T_n\right) := \mathrm{Var}_F\left(T_n\right) = \frac{\sigma^2}{n} = \frac{\int x^2 dF(x) - \left(\int x dF(x)\right)^2}{n}$$

donde $\sigma^2 = \mathrm{Var}\left(X\right)$ y el subindice F es solo para indicar la dependencia con la distribución real.

Ahora dado que no tenemos la distribución real $F(x)$, una opción es utilizar el estimador empírico $\hat{F}_n$ como estimador plug-in en la formulación de la varianza de T_n.

De manera sencilla se puede resumir la técnica de bootstrap como una simulación iid de la distribución $\hat{F}_n$ de modo que se pueda conocer la varianza del estadístico T_n.

En simples pasos la técnica es

1. Seleccione $X_1^*, \ldots, X_n^* \sim \widehat{F}_n$
2. Estime $T_n^* = g\left(X_1^*, \ldots, X_n^*\right)$
3. Repita los Pasos 1 y 2, B veces para obtener $T_{n,1}^*, \ldots, T_{n,B}^*$
4. Estime
$$v_{\mathrm{boot}} = \frac{1}{B}\sum_{b=1}^{B}\left(T_{n,b}^* - \frac{1}{B}\sum_{r=1}^{B} T_{n,r}^*\right)^2$$

Por la ley de los grandes números tenemos que

$$v_{\mathrm{boot}} \xrightarrow{\text{a.s.}} \mathbb{V}_{\widehat{F}_n}\left(T_n\right), \quad \text{si} \quad B \to \infty. \tag{3.3}$$

además llamaremos,

$$\widehat{\mathrm{se}}_{\mathrm{boot}} = \sqrt{v_{\mathrm{boot}}}$$

En pocas palabras lo que tenemos es que

$$\begin{aligned} \text{Mundo Real: } F \quad &\Longrightarrow X_1, \ldots, X_n \Longrightarrow \quad T_n = g\left(X_1, \ldots, X_n\right) \\ \text{Mundo Bootstrap: } \widehat{F}_n \quad &\Longrightarrow X_1^*, \ldots, X_n^* \Longrightarrow \quad T_n^* = g\left(X_1^*, \ldots, X_n^*\right) \end{aligned}$$

En términos de convergencia lo que se tiene es que

$$\mathrm{Var}_F\left(T_n\right) \overbrace{\approx}^{O(1/\sqrt{n})} \mathrm{Var}_{\widehat{F}_n}\left(T_n\right) \overbrace{\approx}^{O(1/\sqrt{B})} v_{boot}$$

producto de la ley de grandes números en ambos casos.

Nota: . ¿Cómo extraemos una muestra de $\hat{F}_n$?

Recuerden que $\hat{F}_n$ asigna la probabilidad de $\frac{1}{n}$ a cada valor usado para construirla.

Por lo tanto, todos los puntos originales $X_1, \ldots, X_n$ tienen probabilidad $\frac{1}{n}$ de ser escogidos, que resulta ser equivalente a un muestreo con remplazo n-veces.

Así que basta cambiar el punto 1. del algoritmo mencionando anteriormente con

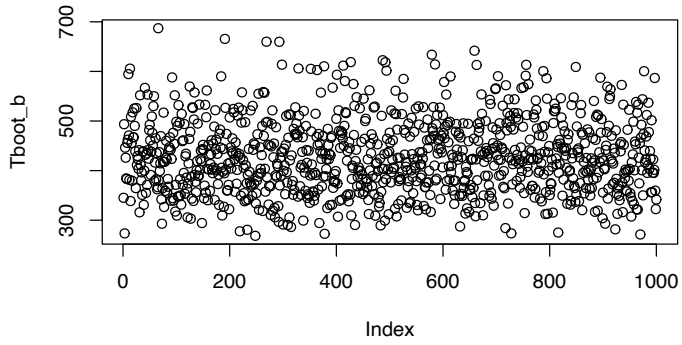
1. Seleccione una muestra con remplazo $X_1^*, \ldots, X_n^*$ de $X_1, \ldots, X_n$.

Ejercicio 3.4. En este ejemplo podemos tomar $B = 1000$ y construir esa cantidad de veces nuestro estimador de varianza:

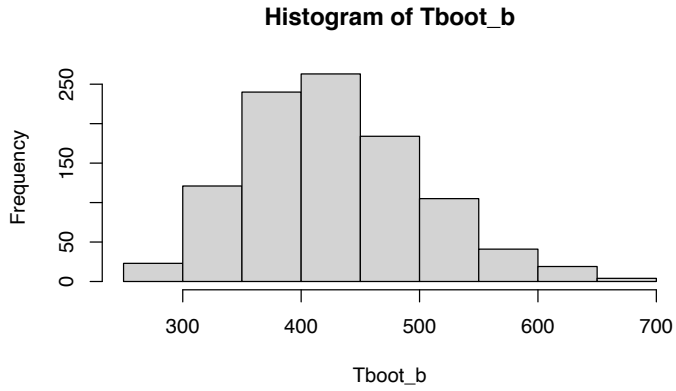
```
B <- 1000
Tboot_b <- NULL

for (b in 1:B) {
    xb <- sample(x, size = n, replace = TRUE)
    Tboot_b[b] <- var(xb)
}

Tboot_b[1:10]
```

```
##  [1] 345.1819 493.5279 273.3998 446.3071 426.0340 384.2662 383.2132 455.8139
##  [9] 462.3363 594.5774
```

```
plot(Tboot_b)
```




```
hist(Tboot_b)
```



Por supuesto podemos encontrar los estadísticos usuales para esta nueva muestra

```
(Tboot <- mean(Tboot_b))
```

```
## [1] 428.066
```

```
(Vboot <- var(Tboot_b))
```

```
## [1] 5504.701
```

```
(sdboot <- sqrt(Vboot))
```

```
## [1] 74.19367
```

Nota: . Si $\hat{\theta}$ es un estimador de θ (bajo cualquier método) entonces podemos sustituir el paso 1 en el algoritmo de Bootstrap por lo siguiente:

1. Seleccione $X_1^*, \ldots, X_n^* \sim F_{\hat{\theta}}$

A este algoritmo modificado le llamamos Bootstrap paramétrico.

3.3.1. Intervalos de confianza

3.3.1.1. Intervalo Normal

Este es el más sencillo y se escribe como

$$T_n \pm z_{\alpha/2}\widehat{\mathrm{Se}}_{\mathrm{boot}} \tag{3.4}$$

Nota: . Este intervalo solo funciona si la distribución de T_n es normal.

El cálculo de este intervalo es

```
c(Tn - z * sdboot, Tn + z * sdboot)
```

```
## [1] 283.8315 574.6653
```

3.3.1.2. Intervalo pivotal

Sea $\theta = T(F)$ y $\widehat{\theta}_n = T\left(\widehat{F}_n\right)$ y defina la cantidad pivotal $R_n = \widehat{\theta}_n - \theta$.

Sea $H(r)$ la función de distribución del pivote:

$$H(r) = \mathbb{P}_F\left(R_n \leq r\right).$$

Además considere $C_n^\star = (a, b)$ donde

$$a = \widehat{\theta}_n - H^{-1}\left(1 - \frac{\alpha}{2}\right) \quad \text{y} \quad b = \widehat{\theta}_n - H^{-1}\left(\frac{\alpha}{2}\right).$$

Se sigue que

$$\begin{aligned}\mathbb{P}(a \leq \theta \leq b) &= \mathbb{P}\left(\widehat{\theta}_n - b \leq R_n \leq \widehat{\theta}_n - a\right) \\ &= H\left(\widehat{\theta}_n - a\right) - H\left(\widehat{\theta}_n - b\right) \\ &= H\left(H^{-1}\left(1 - \frac{\alpha}{2}\right)\right) - H\left(H^{-1}\left(\frac{\alpha}{2}\right)\right) \\ &= 1 - \frac{\alpha}{2} - \frac{\alpha}{2} = 1 - \alpha\end{aligned}$$

Nota: . $C_n^\star = (a, b)$ es un intervalo de confianza al $(1-\alpha)\,\%$.

El problema es que este intervalo depende de H desconocido.

Para resolver este problema, se puede construir una versión *bootstrap* de H usando lo que sabemos hasta ahora:

$$\widehat{H}(r) = \frac{1}{B}\sum_{b=1}^{B} I\left(R_{n,b}^* \leq r\right)$$

donde $R_{n,b}^* = \widehat{\theta}_{n,b}^* - \widehat{\theta}_n$.

Sea r_β^* el cuantil muestral de tamaño β de $\left(R_{n,1}^*, \ldots, R_{n,B}^*\right)$ y sea θ_β^* el cuantil muestral de tamaño β de $\left(\theta_{n,1}^*, \ldots, \theta_{n,B}^*\right)$.

Nota: . Según la notación anterior se cumple que:

$$r_\beta^* = \theta_\beta^* - \widehat{\theta}_n$$

A partir de loa estadísticos anteriores se puede construir un intervalo de confianza aproximado $C_n = (\widehat{a}, \widehat{b})$ al $(1-\alpha)\,\%$ donde:

$$\begin{aligned}\widehat{a} &= \widehat{\theta}_n - \widehat{H}^{-1}\left(1 - \frac{\alpha}{2}\right) = \widehat{\theta}_n - r_{1-\alpha/2}^* = \widehat{\theta}_n - \theta_{1-\alpha/2}^* + \widehat{\theta}_n = 2\widehat{\theta}_n - \theta_{1-\alpha/2}^* \\ \widehat{b} &= \widehat{\theta}_n - \widehat{H}^{-1}\left(\frac{\alpha}{2}\right) = \widehat{\theta}_n - r_{\alpha/2}^* = \widehat{\theta}_n - \theta_{\alpha/2}^* + \widehat{\theta}_n = 2\widehat{\theta}_n - \theta_{\alpha/2}^*\end{aligned}$$

Nota: . El intervalo de confianza pivotal de tamaño $1-\alpha$ es

$$C_n = \left(2\widehat{\theta}_n - \widehat{\theta}_{((1-\alpha/2)B)}^*, 2\widehat{\theta}_n - \widehat{\theta}_{((\alpha/2)B)}^*\right)$$

El intervalo anterior para un nivel de 95 % se estima de la siguiente forma

```
c(2 * Tn - quantile(Tboot_b, 1 - 0.05/2), 2 * Tn -
    quantile(Tboot_b, 0.05/2))
```

```
##    97.5%     2.5%
## 267.1250 552.9294
```

3.3.1.3. Intervalo pivotal studentizado

Una versión mejorada del intervalo pivotal sería a través de la normalización de los estimadores de T_n:

$$Z_n = \frac{T_n - \theta}{\widehat{\text{se}}_{\text{boot}}}.$$

Como θ es desconocido, entonces la versión a estimar es

$$Z_{n,b}^* = \frac{T_{n,b}^* - T_n}{\widehat{\text{se}}_b^*}$$

donde $\widehat{\text{se}}_b^*$ es un estimador del error estándar de $T_{n,b}^*$ no de T_n.

Nota: . Para calcular $Z_{n,b}^*$ requerimos estimar la varianza de $T_{n,b}^*$ para cada b.

Con esto se puede obtener cantidades $Z_{n,1}^*, \ldots, Z_{n,B}^*$ que debería ser próximos a Z_n. (Bootstrap de los estadísticos normalizados)

Sea z_α^* el α-cuantil de $Z_{n,1}^*, \ldots, Z_{n,B}^*$, entonces $\mathbb{P}\left(Z_n \leq z_\alpha^*\right) \approx \alpha$.

Define el intervalo

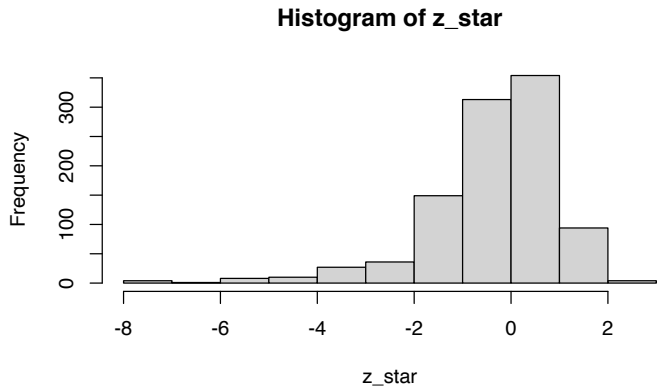
$$C_n = \left(T_n - z_{1-\alpha/2}^* \widehat{\text{se}}_{\text{boot}}, T_n - z_{\alpha/2}^* \widehat{\text{se}}_{\text{boot}}\right)$$

Justificado por el siguiente cálculo:

$$\begin{aligned}
\mathbb{P}\left(\theta \in C_n\right) &= \mathbb{P}\left(T_n - z_{1-\alpha/2}^* \widehat{\text{Se}}_{\text{boot}} \leq \theta \leq T_n - z_{\alpha/2}^* \widehat{\text{Se}}_{\text{boot}}\right) \\
&= \mathbb{P}\left(z_{\alpha/2}^* \leq \frac{T_n - \theta}{\text{se}_{\text{boot}}} \leq z_{1-\alpha/2}^*\right) \\
&= \mathbb{P}\left(z_{\alpha/2}^* \leq Z_n \leq z_{1-\alpha/2}^*\right) \\
&\approx 1 - \alpha
\end{aligned}$$

Note que para este caso tenemos que hacer bootstrap para cada estimador bootstrap calculado.

```
B <- 1000
Tboot_b <- NULL
Tboot_bm <- NULL
sdboot_b <- NULL

for (b in 1:B) {
    xb <- sample(x, size = n, replace = TRUE)
    Tboot_b[b] <- var(xb)
    for (m in 1:B) {
        xbm <- sample(xb, size = n, replace = TRUE)
        Tboot_bm[m] <- var(xbm)
    }
    sdboot_b[b] <- sd(Tboot_bm)
}

z_star <- (Tboot_b - Tn)/sdboot_b

hist(z_star)
```




```
c(Tn - quantile(z_star, 1 - 0.05/2) * sdboot, Tn -
    quantile(z_star, 0.05/2) * sdboot)
```

```
##     97.5%      2.5%
```

```
## 317.7259 707.0044
```

3.3.2. Resumiendo

Resumiendo todos lo métodos de cálculo de intervalos obtenemos

```
knitr::kable(data.frame(Metodo = c("Jackknife", "Bootstrap Normal",
    "Bootstrap Pivotal", "Bootstrap Pivotal Estudentizado"),
    Inferior = c(Tjack - z * sdjack/sqrt(n), Tn - z *
        sdboot, 2 * Tn - quantile(Tboot_b, 1 - 0.05/2),
        Tn - quantile(z_star, 1 - 0.05/2) * sdboot),
    Superior = c(Tjack + z * sdjack/sqrt(n), Tn + z *
        sdboot, 2 * Tn - quantile(Tboot_b, 0.05/2),
        Tn - quantile(z_star, 0.05/2) * sdboot)))
```

Metodo	Inferior	Superior
Jackknife	285.1679	573.3289
Bootstrap Normal	283.8315	574.6653
Bootstrap Pivotal	271.2827	551.4989
Bootstrap Pivotal Estudentizado	317.7259	707.0044

3.4. Ejercicios

1. Repita los ejercicios anteriores para calcular intervalos de confianza para la distancia promedio y la varianza del desplazamiento de las personas. Use los métodos de Jackknife y Bootstrap (con todos sus intervalos de confianza). Dada que la distancia es una medida que puede ser influenciada por distancias muy cortas o muy largas, se puede calcular el logaritmo de esta variable para eliminar la escala de la distancias.
2. Verifique que esta última variable se podría estimar paramétricamente con una distribución normal. Repita los cálculos anteriores tomando como cuantiles los de una normal con media 0 y varianza 1.
3. Compare los intervalos calculados y comente los resultados.
4. Del libro (Wasserman 2006) **Sección 3:** 2, 3, 7, 9, 11.

Capítulo 4

Métodos lineales de regresión

NOTA: Para los siguientes capítulos nos basaremos en los libros (Hastie, Tibshirani y Friedman 2009) y (James y col. 2013).

4.1. Introducción al Aprendizaje Estadístico.

Supongamos que tenemos p variables de entrada que provocan una respuesta Y (variable dependiente) a través de la siguiente relación:

$$Y = f(X_1, \ldots, X_p) + \varepsilon \tag{4.1}$$

donde f es desconocida, las variables X's son las variables de entrada (covariables o predictores) y ε representa un error aditivo a la relación definida por f.

Hay dos motivos por los que estimamos f:

1. **Predicción:** Si se estima f con $\hat{f}$ entonces

$$\hat{Y} = \hat{f}(X_1, \ldots, X_p).$$

asumiendo que el valor medio del error ϵ es cero. Si tuvieramos valores nuevos de los X's entonces podríamos estimar el valor que el corresponde a Y.

En este caso obtener una estructura óptima o precisa de la función $\hat{f}$ no es importante, siempre y cuando sea posible obtener buenas predicciones de Y. Para entender mejor esta idea se puede definir:

a. **Error reducible:** Error de $\hat{f}$ alrededor de f, el cual es propio de la escogencia del modelo.
b. **Error irreducible:** Error que escapa a una estimación perfecta de f. Puede venir de covariables no consideradas en el problema, fuentes de error que no se pueden cuantificar, etc.

$$\begin{aligned}\mathbb{E}\left[\left(\hat{Y}-Y\right)^2\right] &= \mathbb{E}\left[\left(f(X_1,\ldots,X_p)+\varepsilon-\hat{f}(X_1,\ldots,X_p)\right)^2\right] \\ &= \underbrace{\left(f(X_1,\ldots,X_p)-\hat{f}(X_1,\ldots,X_p)\right)^2}_{\text{Reducible}} + \underbrace{\text{Var}\left(\varepsilon\right)}_{\text{irreducible}}.\end{aligned}$$

asumiendo que f y X son conocidas y determinísticas.

2. **Inferencia:** Entender la relación entre X y Y, es decir entender cómo Y cambia como función de las covariables. En este caso sí nos interesa obtener un estimador preciso e interpretable de la función f. Las siguientes preguntas son de interés:

- ¿Cuáles covariables están asociadas con la variable respuesta o dependiente?
- ¿Cuál es la relación entre cada variable predictora y la respuesta?
- ¿La relación entre covariables y variable dependiente es lineal? o ¿la relación es más compleja?

4.1.1. Formas de estimar f

El proceso de estimación de f a través de $\hat{f}$ se realiza sobre un subconjunto de los datos disponibles. A este conjunto se le llama *datos de entrenamiento*. El resto de los datos se puede utilizar para probar la capacidad predictiva del modelo seleccionado.

Existen varias clasificaciones de modelos para estimar f:

- Modelos paramétricos vs modelos no parámetricos. Los modelos pueden tener parámetros que facilitan el proceso de estimación, pero el número

de parámetros debe ser conservador para evitar situaciones de *sobre-ajuste*. Los modelos no-paramétricos requieren de mucha información para dar un buen ajuste, sea a través de una muestra grande o a través de manipular parámetros generales de suavidad (ancho de banda).

- Modelos predictivos vs modelos interpretativos. Entre más flexible (complejo) sea un modelo, más dificil es su interpretación, por lo tanto más dificil es hacer inferencia. Hay modelos muy flexibles que permiten hacer muy buena predicción, pero fácilmente se puede caer en sobreajuste.
- Modelos supervisados vs no supervisados. ¿La variable Y está disponible en la muestra?
- Modelos de regresión vs modelos de clasificación. ¿La variable Y es continua o es una variable categórica?

4.1.2. Medidas de bondad de ajuste

En el caso de regresión, la medida más utilizada es el Error Cuadrático Medio (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

calculada sobre la base de entrenamiento del modelo para evaluar la capacidad de ajuste de $\hat{f}$. Para evaluar la capacidad predictiva del modelo se puede usar el mismo concepto sobre la *base de prueba*. La diferencia entre la magnitud del MSE en los dos conjuntos de datos, puede ser un indicador de sobreajuste.

Para el caso de un problema de aprendizaje estadístico hay interpretaciones de los componentes de sesgo y varianza:

- *Varianza*: variación de $\hat{f}$ ante cambios en los datos de entrenamiento. Modelos más flexibles tienen mayor varianza.
- *Sesgo*: error al aproximar la realidad complicada con un modelo más simple. Modelos más flexibles tienen menor sesgo.

Estrategia de búsqueda de modelos: conforme aumenta la flexibilidad de un modelo el sesgo disminuye, y la varianza no aumenta en el mismo ritmo. A partir de un cierto momento la disminución del sesgo no es lo suficientemente fuerte como para contrarrestar el crecimiento en varianza.

Conclusión: un modelo parsimonioso posiblemente garantizará un valor óptimo en MSE.

4.2. Regresión lineal

El caso más sencillo es cuando se asume que la relación es lineal y se describe de la siguiente forma:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon.$$

Aquí los valores β's son constantes a estimar, las variables X's son las variables de entrada y ε es el error irreducible cometido por hacer esta aproximación.

Las covariables en un modelo de regresión pueden ser:

1. Cuantitativas: variables continuas.
2. Categóricas: variables tipo factor que admiten un número de niveles. Estas variables pueden ser ordinales o nominales, dependiendo si hay un orden natural en la escala de los niveles. Para incorporarla en el modelo de regresión debemos *codificar* la variable:

Ejemplo 4.1. Se tiene la variable G codificada con Casado (1), Soltero (2), Divorciado (3) y Unión Libre (4). Si queremos incorporar esta variable en una regresión podríamos usar la siguiente codificación:

$$X_j = \mathbf{1}_{\{G=j+1\}}$$

que resulta en la matriz

X_1	X_2	X_3
0	0	0
1	0	0
0	1	0
0	0	1

Existen otras formas de codificar este tipo de variables, pero esta es una de las más usuales.

4.2.1. Forma matricial

Podemos escribir el modelo de regresión en forma matricial:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

donde

$$\boldsymbol{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}_{n\times 1} \qquad \boldsymbol{X} = \begin{pmatrix} 1 & X_{1,1} & \cdots & X_{p,1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & X_{1,n} & \cdots & X_{p,n} \end{pmatrix}_{n\times(p+1)}$$

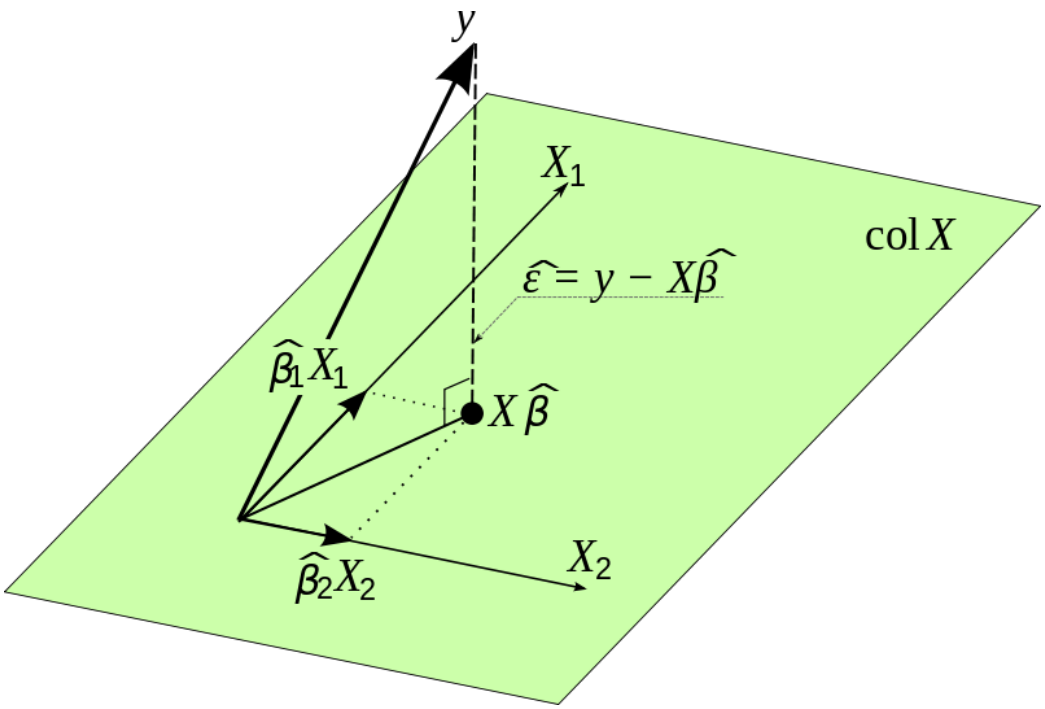
$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}_{n\times 1} \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}_{(p+1)\times 1}$$

Suponemos que $\mathbb{E}[\varepsilon_i] = 0$ y $\mathrm{Var}(\varepsilon_i) = \sigma^2$.

La forma de resolver este problema es por minimos cuadrados. Es decir, buscamos el $\hat{\beta}$ que cumpla lo siguiente:

$$\hat{\beta} = \mathrm{argmin}_\beta (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{4.2}$$

$$= \mathrm{argmin}_\beta \sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} X_{j,i}\beta_j \right)^2 \tag{4.3}$$



Por lo tanto buscaríamos minimizar la suma de *residuos* al cuadrado.

Suponga que γ es un vector cualquiera en $\mathbb{R}^{p+1}$ y defina $V := \{\boldsymbol{X}\boldsymbol{\gamma}, \gamma \in \mathbb{R}^{p+1}\}$, es decir el espacio lineal generado por las columnas (covariables) de $\boldsymbol{X}$. Buscamos entonces un vector β que cumpla:

$$\boldsymbol{X}\boldsymbol{\beta} = \text{Proy}_V \boldsymbol{Y}$$

Entonces dado que $\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \perp V$, es decir $\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \perp \boldsymbol{X}\boldsymbol{\gamma}, \forall \boldsymbol{\gamma} \in \mathbb{R}^{p+1}$ entonces:

$$\begin{aligned}
\boldsymbol{X}\boldsymbol{\gamma} \cdot (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) &= 0 \\
\boldsymbol{\gamma}^\top \boldsymbol{X}^\top (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) &= 0 \\
\boldsymbol{\gamma}^\top \boldsymbol{X}^\top \boldsymbol{Y} &= \boldsymbol{\gamma}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} \\
\boldsymbol{X}^\top \boldsymbol{Y} &= \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} \\
\boldsymbol{\beta} &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}
\end{aligned}$$

Donde se asume que $\boldsymbol{X}^\top \boldsymbol{X}$ debe ser invertible. Si no es así, se puede construir su inversa generalizada pero no garantiza la unicidad de los β's. Es decir, puede existir $\hat{\beta} \neq \tilde{\beta}$ tal que $\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}\tilde{\boldsymbol{\beta}}$. A $\hat{\beta}$ se le llama estimador por mínimos cuadrados de β.

En el caso de predicción tenemos que

$$\begin{aligned}\hat{Y} &= X\hat{\beta} \\ &= \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{Y} \\ &= H\boldsymbol{Y}\end{aligned}$$

Donde H es la matriz "techo" o "hat". La matriz H es la matriz de proyección de Y al espacio de las columnas de X.

Ejercicio 4.1. Suponga que tenemos la regresión simple

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon.$$

Verifique que los estimadores de mínimos cuadrados de β_0 y β_1 son:

$$\begin{aligned}\hat{\beta}_1 &= \frac{\sum_{i=1}^n \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sum_{i=1}^n \left(X_i - \overline{x}\right)^2} \\ \hat{\beta}_0 &= \bar{Y} - \widehat{\beta}_1 \bar{X}\end{aligned}$$

usando los siguiente métodos:

1. El método de proyecciones.
2. Aplicando el criterio de mínimos cuadrados. Ecuación (4.3).

4.2.2. Laboratorio

Usemos la base `mtcars` para los siguientes ejemplos. Toda la información de esta base se encuentra en `?mtcars`.

```
mtcars <- within(mtcars, {
    vs <- factor(vs, labels = c("V-Shape", "Straight-Line"))
    am <- factor(am, labels = c("automatic", "manual"))
    cyl <- factor(cyl)
    gear <- factor(gear)
    carb <- factor(carb)
})

head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec           vs        a
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46      V-Shape    manua
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02      V-Shape    manua
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61 Straight-Line    manua
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44 Straight-Line automati
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02      V-Shape automati
## Valiant           18.1   6  225 105 2.76 3.460 20.22 Straight-Line automati
##                   gear carb
## Mazda RX4            4    4
## Mazda RX4 Wag        4    4
## Datsun 710           4    1
## Hornet 4 Drive       3    1
## Hornet Sportabout    3    2
## Valiant              3    1
```

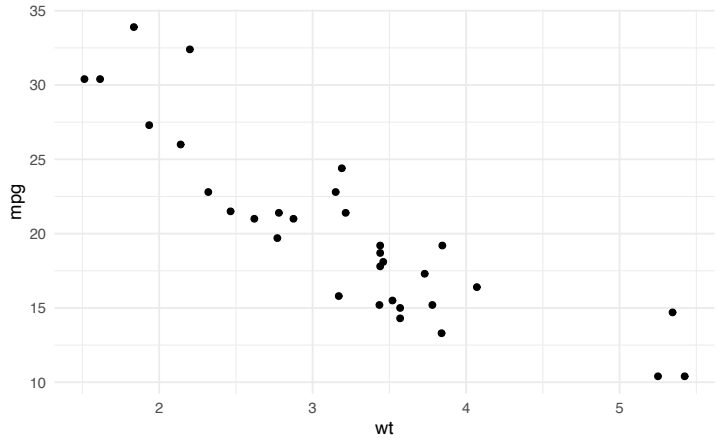
```
summary(mtcars)
```

```
##       mpg        cyl         disp             hp             drat
##  Min.   :10.40   4:11   Min.   : 71.1   Min.   : 52.0   Min.   :2.760
##  1st Qu.:15.43   6: 7   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080
##  Median :19.20   8:14   Median :196.3   Median :123.0   Median :3.695
##  Mean   :20.09          Mean   :230.7   Mean   :146.7   Mean   :3.597
##  3rd Qu.:22.80          3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920
##  Max.   :33.90          Max.   :472.0   Max.   :335.0   Max.   :4.930
##        wt             qsec                   vs             am     gear
##  Min.   :1.513   Min.   :14.50   V-Shape      :18   automatic:19   3:15
##  1st Qu.:2.581   1st Qu.:16.89   Straight-Line:14   manual   :13   4:12
##  Median :3.325   Median :17.71                                     5: 5
```

```
##  Mean   :3.217   Mean   :17.85
##  3rd Qu.:3.610   3rd Qu.:18.90
##  Max.   :5.424   Max.   :22.90
##  carb
##  1: 7
##  2:10
##  3: 3
##  4:10
##  6: 1
##  8: 1
```

Observemos las relaciones generales de las variables de esta base de datos

```
ggplot(mtcars) + geom_point(aes(wt, mpg)) + theme_minimal()
```



El objetivo es tratar la eficiencia del automóvil `mpg` con respecto a su peso `wt`.

Usaremos una regresión lineal para encontrar los coeficientes.

Primero hay que construir la matriz de diseño

```
X <- mtcars$wt
head(X)
```

```
## [1] 2.620 2.875 2.320 3.215 3.440 3.460
```

```
Y <- mtcars$mpg
head(Y)
```

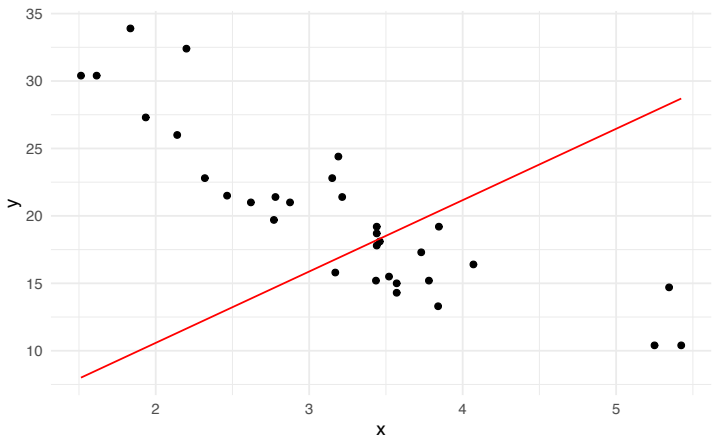
```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1
```

```
(beta1 <- solve(t(X) %*% X) %*% t(X) %*% Y)
```

```
##          [,1]
## [1,] 5.291624
```

```
dfreg <- data.frame(x = X, yreg = X %*% beta1) %>%
    arrange(x)
```

```
ggplot(data = data.frame(x = X, y = Y)) + geom_point(aes(x,
    y)) + geom_line(data = dfreg, aes(x, yreg), color = "red") +
    theme_minimal()
```



en donde podemos concluir que la relación lineal no modela de manera apropiada la relación observada en los datos. Por lo tanto es necesario incluir el intercepto β_0 al modelo lineal:

```
X <- cbind(1, mtcars$wt)
head(X)
```

```
##      [,1]  [,2]
## [1,]    1 2.620
## [2,]    1 2.875
## [3,]    1 2.320
## [4,]    1 3.215
## [5,]    1 3.440
## [6,]    1 3.460
```

```
Y <- mtcars$mpg
head(Y)
```

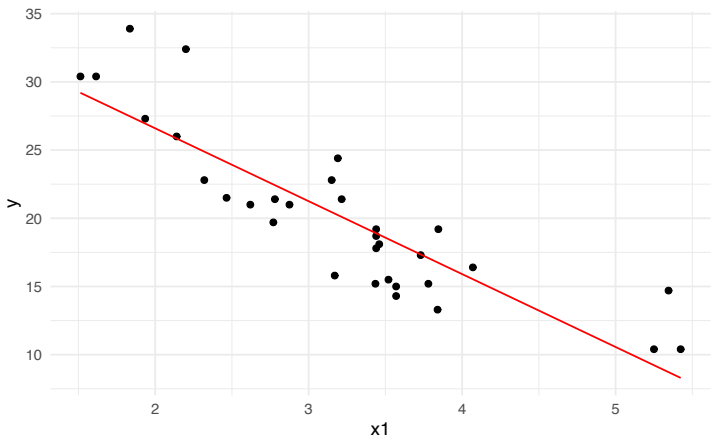
```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1
```

```
(beta01 <- solve(t(X) %*% X) %*% t(X) %*% Y)
```

```
##           [,1]
## [1,] 37.285126
## [2,] -5.344472
```

```
dfreg <- data.frame(x = X, yreg = X %*% beta01) %>%
    arrange(x.2)
```

```
ggplot(data = data.frame(x0 = X[, 1], x1 = X[, 2],
    y = Y)) + geom_point(aes(x1, y)) + geom_line(data = dfreg,
    aes(x.2, yreg), color = "red") + theme_minimal()
```



El mismo resultado se puede obtener a través del comando `lm`:

```
lm(mpg ~ -1 + wt, data = mtcars)
```

```
##
## Call:
## lm(formula = mpg ~ -1 + wt, data = mtcars)
##
## Coefficients:
##    wt
```

```
## 5.292
```

```
lm(mpg ~ wt, data = mtcars)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Coefficients:
## (Intercept)           wt
##      37.285       -5.344
```

Suponga que queremos trabajar con la variable categorica `cyl` (Número de cilindros) como única covariable. Lo que se debe hacer es codificar la variable categórica:

```
X <- model.matrix(mpg ~ cyl, data = mtcars)

head(X)
```

```
##                   (Intercept) cyl6 cyl8
## Mazda RX4                   1    1    0
## Mazda RX4 Wag               1    1    0
## Datsun 710                  1    0    0
## Hornet 4 Drive              1    1    0
## Hornet Sportabout           1    0    1
## Valiant                     1    1    0
```

```
(betas <- solve(t(X) %*% X) %*% t(X) %*% Y)
```

```
##                    [,1]
## (Intercept)  26.663636
## cyl6         -6.920779
## cyl8        -11.563636
```

```
(cylreg <- lm(mpg ~ cyl, data = mtcars))
```

```
##
## Call:
## lm(formula = mpg ~ cyl, data = mtcars)
##
```

```
## Coefficients:
## (Intercept)         cyl6         cyl8
##      26.664       -6.921      -11.564
```

```
(betaslm <- coefficients(cylreg))
```

```
## (Intercept)        cyl6        cyl8
##   26.663636   -6.920779  -11.563636
```

```
# Efecto cyl4: cyl4 = 1, cyl6 = 0, cyl8 = 0

betaslm[1]
```

```
## (Intercept)
##    26.66364
```

```
# Efecto cyl6: cyl4 = 1, cyl6 = 1, cyl8 = 0

betaslm[1] + betaslm[2]
```

```
## (Intercept)
##    19.74286
```

```
# Efecto cyl8: cyl4 = 1, cyl6 = 0, cyl8 = 1

betaslm[1] + betaslm[3]
```

```
## (Intercept)
##        15.1
```

4.3. Propiedades estadísticas

Hasta ahora se han hecho pocos supuestos acerca de la distribución de los datos. Si asumimos que las observaciones Y_i son no correlacionadas y que tienen varianza constante σ^2 y además las covariables son fijas (no aleatorias), entonces:

$$\begin{aligned} E[\hat{\beta}] &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top E[\boldsymbol{Y}] \\ &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{X} \beta \\ &= \beta \\ \mathrm{Var}[\hat{\beta}] &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \mathrm{Var}[\boldsymbol{Y}]((\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top)^\top \\ &= \sigma^2 (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \end{aligned}$$

Note que σ^2 puede ser estimado a través de:

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n-p-1} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i \right)^2 \\ &= \frac{1}{n-p-1} \left\| Y - X\hat{\beta} \right\|^2 \\ &= \frac{1}{n-p-1} \left\| Y - \mathrm{Proy}_V Y \right\|^2 \end{aligned}$$

Otra forma de verlo es

$$\begin{aligned} Y - \mathrm{Proy}_V Y &= X\beta + \varepsilon - \mathrm{Proy}_V (X\beta + \varepsilon) \\ &= X\beta - \mathrm{Proy}_V (\underbrace{X\beta}_{\in V}) + \varepsilon - \underbrace{\mathrm{Proy}_V (\varepsilon)}_{=0} \\ &= X\beta - X\beta + \varepsilon \\ &= \mathrm{Proy}_{V^\top} (\varepsilon) \end{aligned}$$

$$\hat{\sigma}^2 = \frac{1}{\dim(V^\top)} \left\| \mathrm{Proy}_{V^\top} \varepsilon \right\|$$

Cumple con la propiedad que $\mathbb{E}\left[\hat{\sigma}^2\right] = \sigma^2$ (estimador insesgado).

Para poder hacer inferencia sobre β y σ^2 se puede asumir además que los errores son gaussianos:

$$\varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right).$$

y de esta forma se obtiene:

$$Y = X\beta + \varepsilon \sim \mathcal{N}\left(X\beta, \sigma^2 I\right)$$

Y además:

$$\hat{\beta} \sim \mathcal{N}\left(\beta, \sigma^2 (X^\top X)^{-1}\right)$$

Por otro lado se puede comprobar que:

$$(n-p-1)\hat{\sigma}^2 \sim \sigma^2 \chi^2_{n-p-1}.$$

y además se puede comprobar que $\hat{\beta}$ y $\hat{\sigma}^2$ son independientes.

Ejercicio 4.2. Encuentre la varianza para $\hat{\beta}_0$ y $\hat{\beta}_1$ para el caso de la regresión simple.

4.3.1. Prueba t

La significancia de los parámetros β_j se puede verificar a través de la siguiente prueba de hipótesis:

$$H_0 : \beta_j = 0 \quad \text{vs} \quad H_1 : \beta_j \neq 0.$$

En donde el estadístico de prueba es:

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}}$$

donde v_j es el j-esimo elemento de la diagonal de $(X^\top X)^{-1}$.

Bajo H_0: $z_j \sim t_{n-p-1}$ y se rechaza H_0 al nivel α si:

$$|z_j| > t_{n-p-1,1-\frac{\alpha}{2}}$$

4.3.2. Prueba F

Si uno busca medir la significancia de todos los parámetros β_j de forma simultánea, excepto el intercepto. En este caso podemos definir la siguiente hipótesis nula:

$$H_0 : \beta_1 = \cdots = \beta_p = 0 \quad \text{vs} \quad H_1 : \text{ al menos un } \beta \text{ no es cero.}$$

Lo cual es equivalente a comparar el modelo nulo $Y = \beta_0 + \varepsilon$ contra el modelo completo $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$.

Defina la suma total de cuadrados (TSS) y la suma de residuos al cuadrado (RSS) como:

$$TSS = \sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2$$
$$RSS = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2$$

Entonces el estadístico de prueba es:

$$F = \frac{\frac{TSS - RSS}{p}}{\frac{RSS}{n-p-1}} \overset{H_0}{\sim} \frac{\chi^2_p}{\chi^2_{n-p-1}}.$$

y rechazaríamos H_0 al nivel α si:

$$F > F_{p,n-p-1,1-\alpha}.$$

Si por otro lado queremos probar que un conjunto de q covariables son no-significativas entonces probamos (sin pérdida de generalidad):

$$H_0 : \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0$$

a través de la comparación de un modelo completo y uno reducido:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon \qquad \text{Modelo completo}$$
$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-q} X_{p-q} + \varepsilon \qquad \text{Modelo reducido}$$

usando el estadístico de prueba:

$$F = \frac{\frac{RSS_0 - RSS}{q}}{\frac{RSS}{n-p-1}} \overset{H_0}{\approx} \frac{\chi^2_q}{\chi^2_{n-p-1}}.$$

donde RSS_0 es la suma de residuos al cuadrado del modelo reducido. En este caso se rechazaría H_0 al nivel α si $F > F_{q,n-p-1,1-\alpha}$.

4.3.3. Laboratorio

Siguiendo con nuestro ejemplo, vamos a explorar un poco más la función `lm`.

```
modelo_wt <- lm(mpg ~ wt, data = mtcars)
summary(modelo_wt)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

```
modelo_wt_cyl <- lm(mpg ~ wt + cyl, data = mtcars)
summary(modelo_wt_cyl)
```

```
##
## Call:
```

```
## lm(formula = mpg ~ wt + cyl, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5890 -1.2357 -0.5159  1.3845  5.7915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.9908     1.8878  18.006  < 2e-16 ***
## wt           -3.2056     0.7539  -4.252 0.000213 ***
## cyl6         -4.2556     1.3861  -3.070 0.004718 **
## cyl8         -6.0709     1.6523  -3.674 0.000999 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.557 on 28 degrees of freedom
## Multiple R-squared:  0.8374, Adjusted R-squared:   0.82
## F-statistic: 48.08 on 3 and 28 DF,  p-value: 3.594e-11
```

```
anova(modelo_wt, modelo_wt_cyl)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt
## Model 2: mpg ~ wt + cyl
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     30 278.32
## 2     28 183.06  2    95.263 7.2856 0.002835 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
modelo_nulo <- lm(mpg ~ 1, data = mtcars)
summary(modelo_nulo)
```

```
##
## Call:
## lm(formula = mpg ~ 1, data = mtcars)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -9.6906 -4.6656 -0.8906  2.7094 13.8094
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.091      1.065   18.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.027 on 31 degrees of freedom
```

```
anova(modelo_nulo, modelo_wt_cyl)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ 1
## Model 2: mpg ~ wt + cyl
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     31 1126.05
## 2     28  183.06  3    942.99 48.079 3.594e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit <- lm(mpg ~ ., data = mtcars)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5087 -1.3584 -0.0948  0.7745  4.6251
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    23.87913   20.06582   1.190   0.2525
## cyl6           -2.64870    3.04089  -0.871   0.3975
## cyl8           -0.33616    7.15954  -0.047   0.9632
```

```
## disp             0.03555    0.03190   1.114   0.2827
## hp              -0.07051    0.03943  -1.788   0.0939 .
## drat             1.18283    2.48348   0.476   0.6407
## wt              -4.52978    2.53875  -1.784   0.0946 .
## qsec             0.36784    0.93540   0.393   0.6997
## vsStraight-Line  1.93085    2.87126   0.672   0.5115
## ammanual         1.21212    3.21355   0.377   0.7113
## gear4            1.11435    3.79952   0.293   0.7733
## gear5            2.52840    3.73636   0.677   0.5089
## carb2           -0.97935    2.31797  -0.423   0.6787
## carb3            2.99964    4.29355   0.699   0.4955
## carb4            1.09142    4.44962   0.245   0.8096
## carb6            4.47757    6.38406   0.701   0.4938
## carb8            7.25041    8.36057   0.867   0.3995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.833 on 15 degrees of freedom
## Multiple R-squared:  0.8931, Adjusted R-squared:  0.779
## F-statistic:  7.83 on 16 and 15 DF,  p-value: 0.000124
```

4.4. Medida de bondad de ajuste

A través de la prueba F uno puede concluir si un modelo es significativo o no bajo un cierto nivel de confianza, o bien puede comparar si un modelo reducido es más significativo que uno completo, pero no nos da herramientas para decidir si un modelo es mejor que otro.

Hay varias medidas para comparar modelos (la veremos con más detalle en otro capítulo):

- Error estándar residual (σ)
- R^2 y R^2 ajustado
- C_p de Mallows
- Akaike Information Criterion (AIC)
- Bayesian Information Criterion (BIC)

Los índices C_p de Mallows, AIC y BIC los veremos después.

Error estándar residual Se define como

$$\begin{aligned}\text{RSE} &= \sqrt{\hat{\sigma^2}} \\ &= \sqrt{\frac{1}{n-p-1}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \\ &= \sqrt{\frac{\text{RSS}}{n-p-1}}\end{aligned}$$

Entre más pequeño mejor, pero **depende de las unidades de** Y.

Estadístico R^2

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

- **RSS:** Varianza sin explicar por el modelo **completo**.
- **TSS:** Varianza sin explicar por el modelo **nulo**.

Interpretación: proporción de variabilidad en Y que es explicada a través de las covariables en X. Ya que $TSS - RSS$ representa la variabilidad explicada a través del modelo de regresión.

Limitación: puede tener un valor alto bajo un número grande de covariables, ya que RSS tiende a ser bajo conforme aumenta la complejidad del modelo (sobreajuste).

Estadístico R^2 **ajustado**

$$R^2_{adj} = 1 - \frac{\frac{\text{RSS}}{n-p-1}}{\frac{\text{TSS}}{n-1}}$$

4.4.1. Laboratorio

```
# Número de datos
n <- 1000
# Número de variables
p <- 2

x1 <- rnorm(1000)
```

```
x2 <- runif(1000)
y <- 1 + x1 + x2 + rnorm(1000, sd = 0.5)

fit <- lm(y ~ x1 + x2)
```

4.4.1.1. R^2

```
(TSS <- sum((y - mean(y))^2))
```

```
## [1] 1404.421
```

```
(RSS <- sum((y - fitted(fit))^2))
```

```
## [1] 256.8679
```

```
1 - RSS/TSS
```

```
## [1] 0.8171005
```

Otra forma de entender el R^2 es notando que

```
cor(y, fitted(fit))^2
```

```
## [1] 0.8171005
```

4.4.1.2. R^2 ajustado

```
(TSS_adj <- TSS/(n - 1))
```

```
## [1] 1.405827
```

```
(RSS_adj <- RSS/(n - p - 1))
```

```
## [1] 0.2576408
```

```
1 - RSS_adj/TSS_adj
```

```
## [1] 0.8167336
```

4.4.1.3. summary

```
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73583 -0.35052  0.01175  0.33270  1.42618
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.05443    0.03252   32.42   <2e-16 ***
## x1           1.02131    0.01573   64.92   <2e-16 ***
## x2           0.91189    0.05655   16.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5076 on 997 degrees of freedom
## Multiple R-squared:  0.8171, Adjusted R-squared:  0.8167
## F-statistic:  2227 on 2 and 997 DF,  p-value: < 2.2e-16
```

4.5. Predicción

Hay dos tipos de errores que se deben considerar en regresiones lineales:

1. **Error Reducible:** Recuerde que $\hat{Y} = X\hat{\beta}$ es el estimador de la función $f(X) = X\beta = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$.

Por lo tanto su error (reducible) es:

$$\left(f(X) - \hat{Y}\right)^2.$$

Para un conjunto de datos X_0, tenemos que

$$\hat{\beta} \sim \mathcal{N}\left(\beta, \sigma^2\left((X_0^\top X_0)^{-1}\right)\right)$$
$$\implies \hat{Y} = X_0\hat{\beta} \sim \mathcal{N}\left((X_0\beta, \sigma^2 X_0^\top((X_0^\top X_0)^{-1}X_0))\right)$$

Por lo tanto un **intervalo de confianza** al $1-\alpha$ para $X_0\beta$ es

$$X_0\hat{\beta} \pm z_{1-\frac{\alpha}{2}}\hat{\sigma}\sqrt{X_0^\top(X_0^\top X_0)^{-1}X_0}.$$

2. **Error irreducible:** Aún conociendo perfectamente los β's, existe el error desconocido $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ del modelo

$$Y = X\beta + \varepsilon.$$

Entonces la varianza total de la predicción sería

$$\sigma^2 + \sigma^2 X_0^\top((X_0^\top X_0)^{-1}X_0)$$

Entonces un **intervalo de predicción** al $1-\alpha$ debe tomar en cuenta ese error y por lo tanto

$$X_0\beta \pm z_{1-\frac{\alpha}{2}}\hat{\sigma}\sqrt{1 + X_0^\top(X_0^\top X_0)^{-1}X_0}.$$

4.5.1. Laboratorio

```
lm.r <- lm(mpg ~ wt, data = mtcars)

range(mtcars$wt)
```

```
## [1] 1.513 5.424
```

```
(datos_nuevos <- data.frame(wt = c(2.5, 3, 3.5)))
```

```
##    wt
## 1 2.5
## 2 3.0
## 3 3.5
```

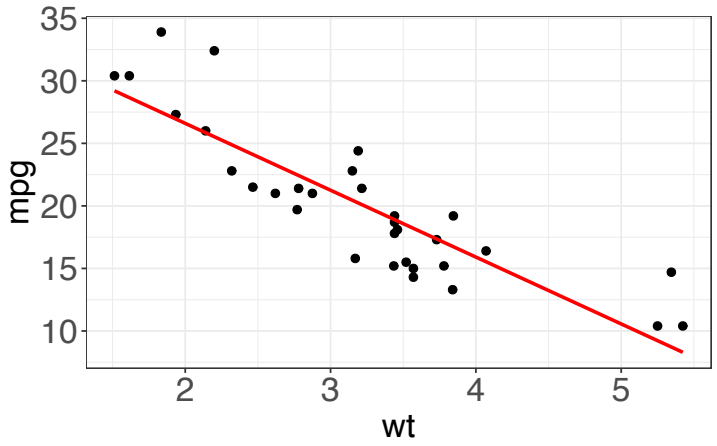
```
predict(object = lm.r, newdata = datos_nuevos, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 23.92395 22.55284 25.29506
## 2 21.25171 20.12444 22.37899
## 3 18.57948 17.43342 19.72553
```

```
predict(object = lm.r, newdata = datos_nuevos, interval = "prediction")
```

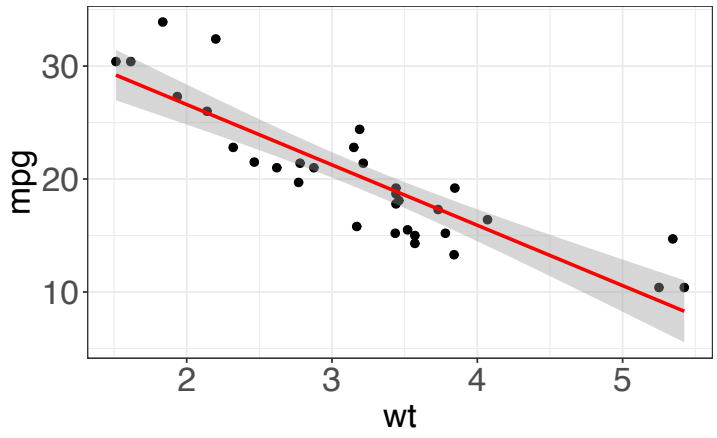
```
##        fit      lwr      upr
## 1 23.92395 17.55411 30.29378
## 2 21.25171 14.92987 27.57355
## 3 18.57948 12.25426 24.90469
```

```
p <- ggplot(mtcars, aes(x = wt, y = mpg))
p <- p + geom_point(size = 2)        # Use circulos de tamaño 2
p <- p + geom_smooth(method = lm,    # Agregar la línea de regresión
              se = FALSE,            # NO incluir el intervalo de confianza
              size = 1,
              col = "red")           # Línea de color rojo
p <- p + theme_bw()                  # Tema de fondo blanco
p <- p + theme(axis.text = element_text(size = 20),  # Aumentar el tamaño
               axis.title = element_text(size = 20)) # de letra en los ejes

# Dibujar el gráfico
p
```




```
# # Guardar el gráfico en un archivo pdf
# ggsave(filename = 'linear_reg_sin_IC.pdf') #
```

```
p <- ggplot(mtcars, aes(x = wt, y = mpg))
p <- p + geom_point(size = 2)        # Use circulos de tamaño 2
p <- p + geom_smooth(method = lm,    # Agregar la línea de regresión
              se = TRUE,             # Incluir el intervalo de confianza
              size = 1,
              col = "red")           # Línea de color rojo
p <- p + theme_bw()                  # Tema de fondo blanco
p <- p + theme(axis.text = element_text(size = 20),  # Aumentar el tamaño
               axis.title = element_text(size = 20)) # de letra en los ejes

# Dibujar el gráfico
p
```

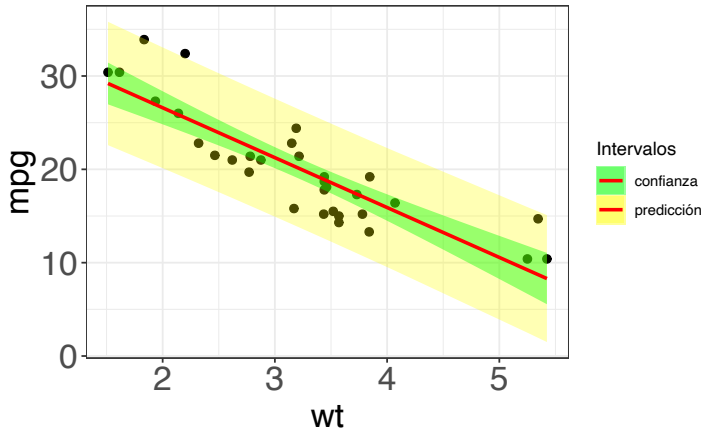



```
# # Guardar el gráfico en un archivo pdf
# ggsave(filename = 'linear_reg_con_IC.pdf') #
```

4.5.1.3. Ajuste de la regresión con intervalos de confianza y predicción

```
# Agregamos a mtcars el intervalo de predicción
# para cada dato
mtcars.pred <- data.frame(mtcars, predict(lm.r, interval = "prediction"))

p <- ggplot(mtcars.pred, aes(x = wt, y = mpg))
# Use circulos de tamaño 2
p <- p + geom_point(size = 2)
# Agregue una banda de tamaño [lwr, upr] para
# cada punto y llamela 'predicción'
p <- p + geom_ribbon(aes(ymin = lwr, ymax = upr, fill = "predicción"),
    alpha = 0.3)
# Agregue el intervalo de confianza usual y llame
# a ese intervalo 'confianza'
p <- p + geom_smooth(method = lm, aes(fill = "confianza"),
    size = 1, col = "red")
# Para agregar bien las leyendas
p <- p + scale_fill_manual("Intervalos", values = c("green",
    "yellow"))
p <- p + theme_bw()
```

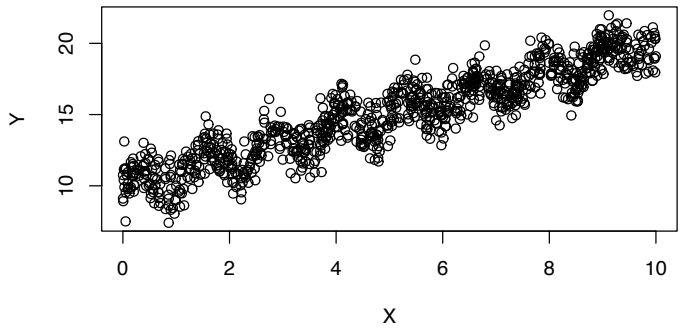
```
p <- p + theme(axis.text = element_text(size = 20),
    axis.title = element_text(size = 20))

# Dibujar el gráfico
p
```




```
# # Guardar el gráfico en un archivo pdf
# ggsave(filename = 'linear_reg_con_IC_IP.pdf') #
```

Repitamos el mismo ejercicio anterior pero con un caso más sencillo.

```
n <- 1000

X <- runif(n, 0, 10)
Y <- 10 + sin(5 * X) + X + rnorm(1000, 0, 1)
toyex.initial <- data.frame(X, Y) %>%
    arrange(X)

plot(toyex.initial)
```




```
lm.toyex.initial <- lm(Y ~ X, data = toyex.initial)

summary(lm.toyex.initial)
```

```
##
## Call:
## lm(formula = Y ~ X, data = toyex.initial)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4587 -0.8232  0.0468  0.8709  3.4115
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.01402    0.07847  127.61   <2e-16 ***
## X            0.98895    0.01340   73.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.208 on 998 degrees of freedom
## Multiple R-squared:  0.8452, Adjusted R-squared:  0.845
## F-statistic:  5448 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
toyex.pred.initial <- data.frame(toyex.initial, predict(lm.toyex.initial,
    interval = "prediction"))
```

Ahora, quisiera generar muchas muestras del mismo experimento

```
toyex.pred <- NULL

for (i in 1:10) {
    X <- runif(n, 0, 10)
    Y <- 10 + sin(5 * X) + X + rnorm(1000, 0, 1)
    toyexi <- data.frame(im = i, X, Y)
    toyexi <- toyexi %>%
        arrange(X)
    toyex.pred <- bind_rows(toyex.pred, data.frame(toyexi,
        predict(lm.toyex.initial, interval = "prediction")))
}


for (i in 1:10) {
    toyex.pred$fit <- fitted(lm(formula = Y ~ X, data = toyex.pred[toyex.pred$
        i, ]))
}

toyex.pred$im <- as.factor(toyex.pred$im)
```

```
library(gganimate)

ggplot(data = toyex.pred, aes(x = X, y = Y)) + geom_point(size = 1) +
    geom_smooth(data = toyex.initial, method = lm,
        mapping = aes(fill = "confianza"), size = 1,
        col = "red") + geom_ribbon(data = toyex.pred.initial,
    mapping = aes(x = X, ymin = lwr, ymax = upr, fill = "predicción",
        ), alpha = 0.3) + labs(title = paste0("Muestra #: {closest_state}")) +
    scale_fill_manual("Intervalos", values = c("green",
        "yellow")) + theme_bw() + theme(axis.text = element_text(size = 20),
    axis.title = element_text(size = 20)) + transition_states(im)
```

4.6. Interacciones

Suponga un modelo lineal con dos covariables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

Aumentemos en 1 unidad X_1 y rescribamos el modelo original

$$\begin{aligned} Y &= \beta_0 + \beta_1(X_1 + 1) + \beta_2 X_2 + \varepsilon \\ Y &= (\beta_0 + \beta_1) + \beta_1 X_1 + \beta_2 X_2 + \varepsilon \\ Y &= \tilde{\beta}_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon \end{aligned}$$

Es decir, el modelo original sigue siendo teniendo la misma estructura aunque hayamos cambiado el X_1. Este fenómeno ocurre siempre bajo transformaciones lineales de las variables.

Ahora suponga que tenemos el siguiente modelo:

$$Y = \beta_0 + \beta_1 X_1 X_2 + \varepsilon$$

y aumentamos en 1 el X_1:

$$\begin{aligned} Y &= \beta_0 + \beta_1(X_1 + 1)X_2 + \varepsilon \\ Y &= \beta_0 + \beta_1 X_2 + \beta_1 X_1 X_2 + \varepsilon \end{aligned}$$

Note que en este caso no se logra mantener el mismo tipo de estructura. Una forma de arreglar el problema es incluir las *interacciones* junto con todos sus *efectos principales*.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$$

Este modelo se le conoce como modelo lineal con interacciones (caso de 2 covariables). Este modelo considera la posible interacción entre las covariables

X_1 y X_2 que permiten cambiar tanto el intercepto como las pendientes de los efectos principales.

Principio de jerarquía. Con el fin de mantener la estructura del modelo lineal, siempre es necesario incluir los efectos principales cuando se determina que una interacción entre ellos es significativa.

Ejercicio 4.3. Compruebe que para el caso anterior, si aumenta en una unidad X_1, el modelo preserva su estructura.

4.6.1. Laboratorio

Generamos una base de datos nueva con solamente `wt` centrado

```
# La función across y where solo funciona solo
# para dplyr 1.0 Si tienen otra versión, pueden
# usar mutate_if

mtcars_centered <- mtcars %>%
    mutate(across("wt", scale, scale = FALSE, center = TRUE))

# Si no se tiene dplyr 1.0

mtcars_centered <- mtcars %>%
    mutate_at("wt", scale, scale = FALSE, center = TRUE)
```

Compare lo que ocurre con los coeficientes de la base original y la nueva base.

```
summary(lm(mpg ~ wt + disp, data = mtcars))
```

```
##
## Call:
## lm(formula = mpg ~ wt + disp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4087 -2.3243 -0.7683  1.7721  6.3484
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 34.96055    2.16454  16.151 4.91e-16 ***
## wt          -3.35082    1.16413  -2.878  0.00743 **
## disp        -0.01773    0.00919  -1.929  0.06362 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.917 on 29 degrees of freedom
## Multiple R-squared:  0.7809, Adjusted R-squared:  0.7658
## F-statistic: 51.69 on 2 and 29 DF,  p-value: 2.744e-10
```

```
summary(lm(mpg ~ wt + disp, data = mtcars_centered))
```

```
##
## Call:
## lm(formula = mpg ~ wt + disp, data = mtcars_centered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4087 -2.3243 -0.7683  1.7721  6.3484
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.18011    2.18221  11.081 6.12e-12 ***
## wt          -3.35082    1.16413  -2.878  0.00743 **
## disp        -0.01773    0.00919  -1.929  0.06362 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.917 on 29 degrees of freedom
## Multiple R-squared:  0.7809, Adjusted R-squared:  0.7658
## F-statistic: 51.69 on 2 and 29 DF,  p-value: 2.744e-10
```

Supongamos que formamos un modelo con solo la interacción y no incluimos los efectos directos.

```
summary(lm(mpg ~ wt * disp - wt - disp, data = mtcars))
```

```
##
## Call:
```

```
## lm(formula = mpg ~ wt * disp - wt - disp, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.259 -2.603 -1.657  2.165  8.589
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.2621926  1.0418029  25.208  < 2e-16 ***
## wt:disp     -0.0072897  0.0009721  -7.499 2.33e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.614 on 30 degrees of freedom
## Multiple R-squared:  0.6521, Adjusted R-squared:  0.6405
## F-statistic: 56.24 on 1 and 30 DF,  p-value: 2.329e-08
```

```
summary(lm(mpg ~ wt * disp - wt - disp, data = mtcars_centered))
```

```
##
## Call:
## lm(formula = mpg ~ wt * disp - wt - disp, data = mtcars_centered)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.878 -2.775 -1.162  2.409 11.150
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.460008   0.859706  24.962  < 2e-16 ***
## wt:disp     -0.013127   0.002714  -4.837 3.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.592 on 30 degrees of freedom
## Multiple R-squared:  0.4382, Adjusted R-squared:  0.4195
## F-statistic:  23.4 on 1 and 30 DF,  p-value: 3.686e-05
```

El modelo correcto sería el siguiente:

```
summary(lm(mpg ~ wt + disp + wt * disp, data = mtcars))
```

```
##
## Call:
## lm(formula = mpg ~ wt + disp + wt * disp, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.267 -1.677 -0.836  1.351  5.017
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 44.081998   3.123063  14.115 2.96e-14 ***
## wt          -6.495680   1.313383  -4.946 3.22e-05 ***
## disp        -0.056358   0.013239  -4.257  0.00021 ***
## wt:disp      0.011705   0.003255   3.596  0.00123 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.455 on 28 degrees of freedom
## Multiple R-squared:  0.8501, Adjusted R-squared:  0.8341
## F-statistic: 52.95 on 3 and 28 DF,  p-value: 1.158e-11
```

```
summary(lm(mpg ~ wt + disp + wt * disp, data = mtcars_centered))
```

```
##
## Call:
## lm(formula = mpg ~ wt + disp + wt * disp, data = mtcars_centered)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.267 -1.677 -0.836  1.351  5.017
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.183772   1.857605  12.480 5.87e-13 ***
## wt          -6.495680   1.313383  -4.946 3.22e-05 ***
## disp        -0.018699   0.007741  -2.416  0.02248 *
```

```
## wt:disp      0.011705   0.003255   3.596  0.00123 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.455 on 28 degrees of freedom
## Multiple R-squared:  0.8501, Adjusted R-squared:  0.8341
## F-statistic: 52.95 on 3 and 28 DF,  p-value: 1.158e-11
```

Ejercicio 4.4. Repita los comandos anteriores con la siguiente base de datos y explique los resultados.

```
mtcars_scaled <- mtcars %>%
    mutate(across(c("wt", "disp"), scale, scale = TRUE,
        center = TRUE))
```

4.7. Supuestos

El modelo lineal tiene los siguientes supuestos:

Linealidad En la forma lineal de la relación variable dependiente-covariables.

Errores centrados $\mathbb{E}(\varepsilon_i) = 0$.

Homocedasticidad $\text{Var}(\varepsilon_t) = \mathbb{E}(\varepsilon_t - \mathbb{E}\varepsilon_t)^2 = \mathbb{E}\varepsilon_t^2 = \sigma^2$ para todo t. Es decir, la varianza del modelo (**error irreducible**) no depende de las variables independientes u otro factor.

Normalidad de los residuos $\varepsilon \sim N(0, \sigma^2)$.

Independencia de los errores $\text{Cov}(\varepsilon_t, \varepsilon_s) = \mathbb{E}(\varepsilon_t - \mathbb{E}\varepsilon_t)(\varepsilon_s - \mathbb{E}\varepsilon_s) = \mathbb{E}\varepsilon_t\varepsilon_s = 0$ para todo t, s con $t \neq s$: si para una observación dada existe un error, este no debe depender del error de otra observación.

Si este supuesto no se cumple puede provocar que los errores estándar en intervalos de confianza y predicción sean subestimados. Es decir que un intervalo del 95 % tendrá un margen de error menor y se rechazaría más fácilmente la hipotesis nula de las pruebas t y F.

Multicolinealidad Se asume que la matriz X^TX es invertible, es decir X es una matriz de rango completo. Para esto cada una las covariables no debe ser linealmente dependientes, es decir X^TX de debe acercarse a ser a una matriz singular con determinante cercano a 0. Es decir que cada variable explica aproximadamente "un aspecto o característica"

del modelo. Sin embargo puede pasar que varias variables expliquen la misma característica y el modelo se vuelve **inestable** por decidir entre las dos variables. Por ejemplo: la temperatura en grados centigrados y farenheit.

Esto generaría que Var (β) sea alto ya que

$$\text{Var}(\beta) = \sigma^2(X^\top X)^{-1}$$

Más observaciones que predictores En caso contrario existen formas alternativas de definir el problema de regresión. (Volveremos a esto cuando veamos selección de modelos)

4.7.1. Chequeos básicos de las hipótesis de regresión lineal

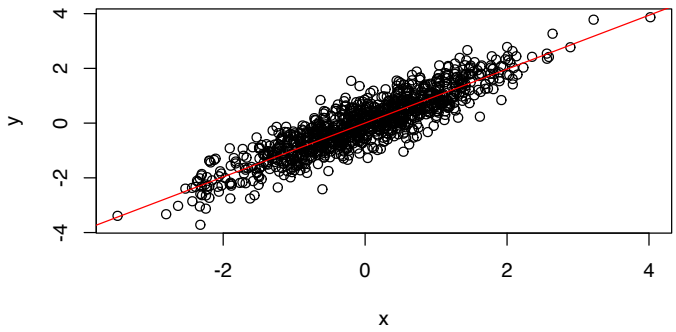
4.7.1.1. Linealidad, Errores con esperanza nula, Homocedasticidad

Estos supuestos se puede constantar a partir de un gráfico de residuos ya que en el caso ideal $e_i = \hat{Y}_i - Y_i \perp \hat{Y}_i$. Entonces si este gráfico presenta patrones, quiere indicar que la regresión, no es lineal, que los errores no tienen esperanza nula y que la varianza no es constante.

Se pueden aplicar transformaciones para resolver estos problemas. Normalmente se usan transformaciones como raiz cuadrada o logaritmos.

Ejemplo 4.2. Caso ideal

```
x <- rnorm(1000)
y <- x + rnorm(1000, sd = 0.5)

fit <- lm(y ~ x)
plot(x, y)
abline(a = coef(fit)[1], b = coef(fit)[2], col = "red")
```




```
plot(fitted(fit), residuals(fit))
abline(h = 0, col = "red")
```

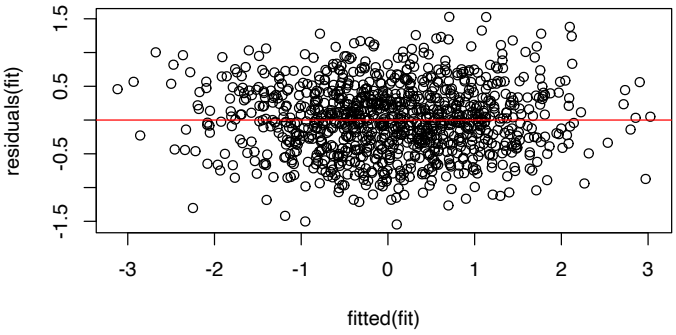
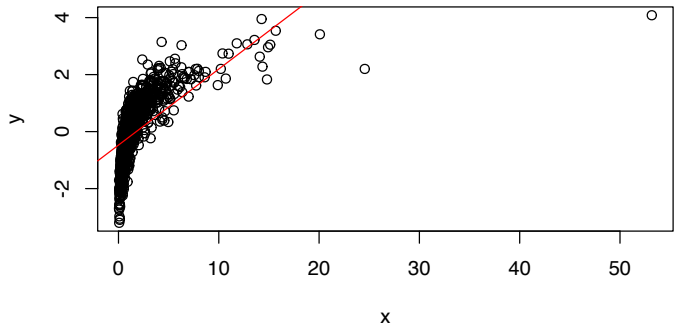


Figura 4.1: Gráfico de residuos caso lineal

Caso no-lineal

```
x <- exp(rnorm(1000))
y <- log(x) + rnorm(1000, sd = 0.5)

fit <- lm(y ~ x)
plot(x, y)
abline(a = coef(fit)[1], b = coef(fit)[2], col = "red")
```




```
plot(fitted(fit), residuals(fit))
abline(h = 0, col = "red")
```

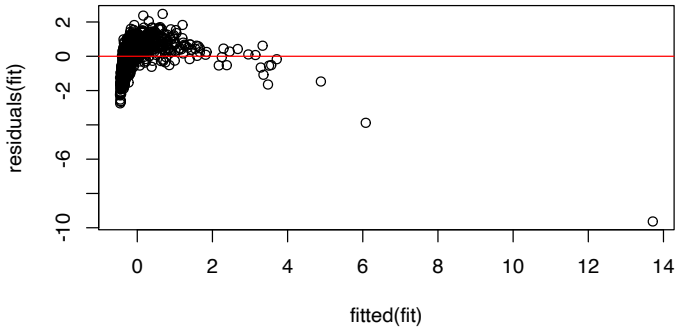
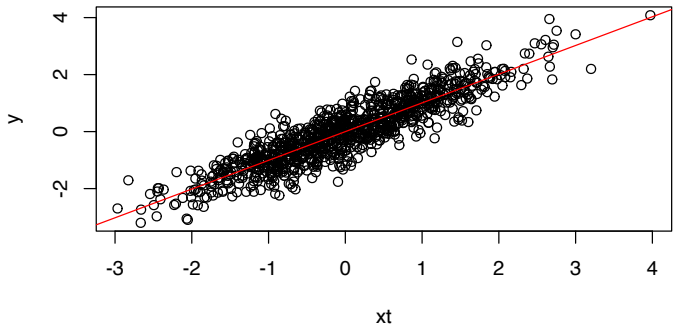


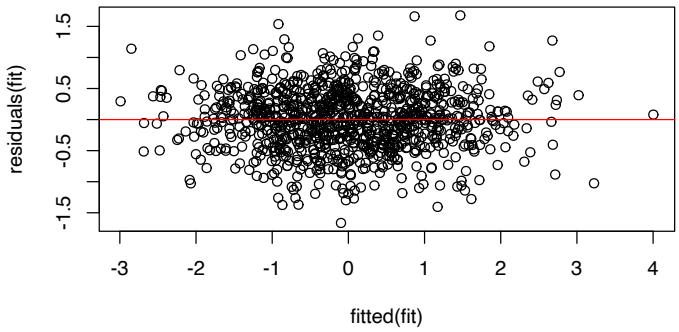
Figura 4.2: Gráfico de residuos caso no-lineal

Caso no-lineal transformado

```
xt <- log(x)


fit <- lm(y ~ xt)
plot(xt, y)
abline(a = coef(fit)[1], b = coef(fit)[2], col = "red")
```




```
plot(fitted(fit), residuals(fit))
abline(h = 0, col = "red")
```



4.7.1.2. Independencia de los errores

En este caso defina $\rho(k) = \text{Cov}(\varepsilon_i, \varepsilon_{i+k})$. Si los residuos son independientes, entonces debe ocurrir que

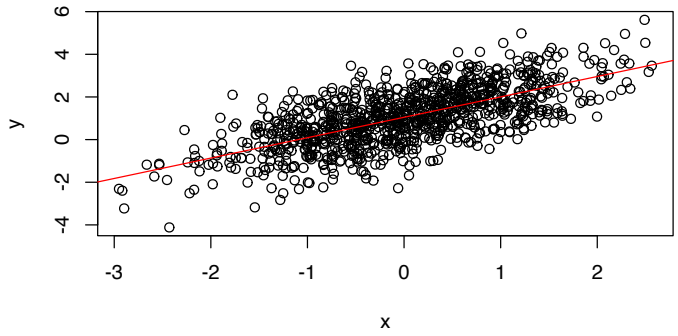
$$\rho(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0. \end{cases}$$

Se calcula la función de autocorrelación empírica y se grafica para analizar su comportamiento

Caso ideal

```
x <- rnorm(1000)
y <- 1 + x + rnorm(1000, sd = 1)
```

```
fit <- lm(y ~ x)
plot(x, y)
abline(a = coef(fit)[1], b = coef(fit)[2], col = "red")
```

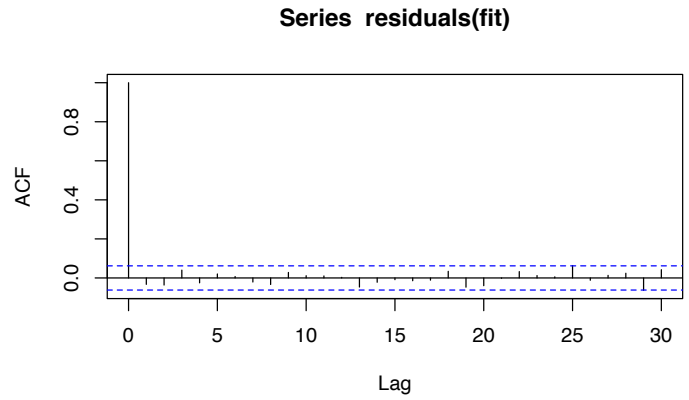



```
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2665 -0.6871  0.0002  0.6670  2.9410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.04643    0.03183   32.87   <2e-16 ***
## x            0.95650    0.03287   29.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.007 on 998 degrees of freedom
```

```
## Multiple R-squared:  0.4589, Adjusted R-squared:  0.4584
## F-statistic: 846.6 on 1 and 998 DF,  p-value: < 2.2e-16
```

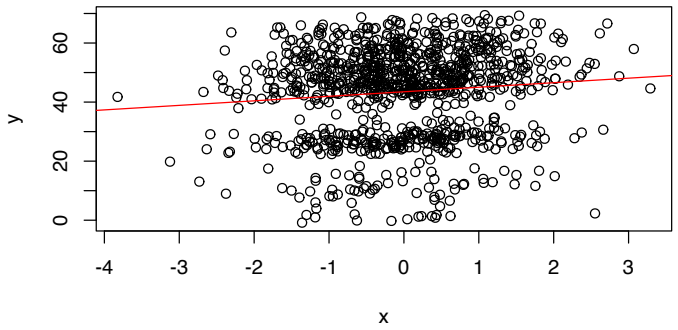
```
acf(residuals(fit))
```



Caso errores auto-correlacionados

```
x <- rnorm(1000)
y <- 1 + x + diffinv(rnorm(999, sd = 1), lag = 1)
```

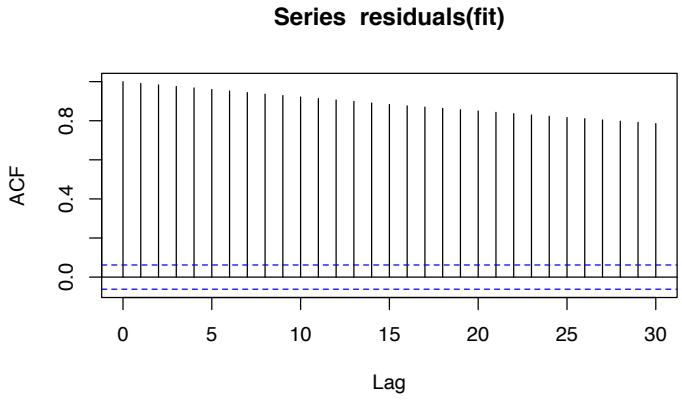
```
fit <- lm(y ~ x)
plot(x, y)
abline(a = coef(fit)[1], b = coef(fit)[2], col = "red")
```



```
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.109 -13.583   3.439  11.036  26.104
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  43.4918     0.4791  90.787  < 2e-16 ***
## x             1.5347     0.4771   3.217  0.00134 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.15 on 998 degrees of freedom
## Multiple R-squared:  0.01026,    Adjusted R-squared:  0.00927
## F-statistic: 10.35 on 1 and 998 DF,  p-value: 0.001339
```

```
acf(residuals(fit))
```



4.7.1.3. Normalidad de los errores

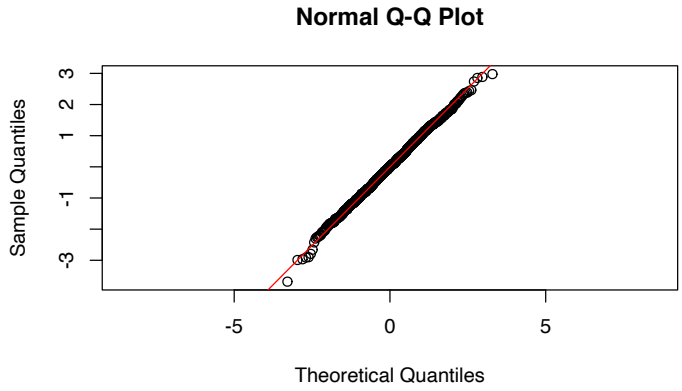
Este hipótesis es crucial para hacer las pruebas t y F que vimos anteriormente.

Para revisar si se cumple solo basta hacer una `qqplot` de los residuos.

Caso ideal

```
x <- rnorm(1000)
y <- 1 + x + rnorm(1000, sd = 1)
fit <- lm(y ~ x)
```

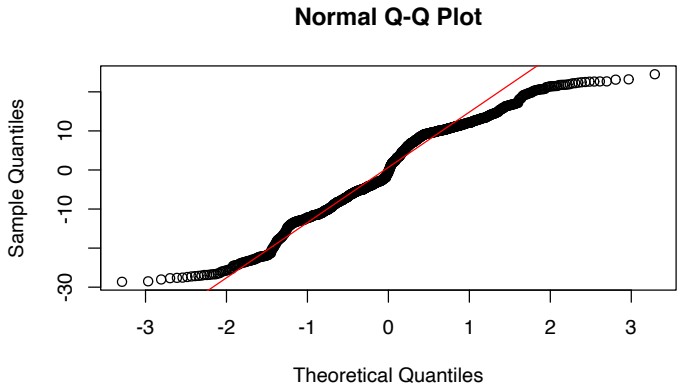
```
qqnorm(residuals(fit), asp = 1)
qqline(residuals(fit), col = "red")
```



Caso errores auto-correlacionados

```
x <- rnorm(1000)
y <- 1 + x + diffinv(rnorm(999, sd = 1), lag = 1)
fit <- lm(y ~ x)
```

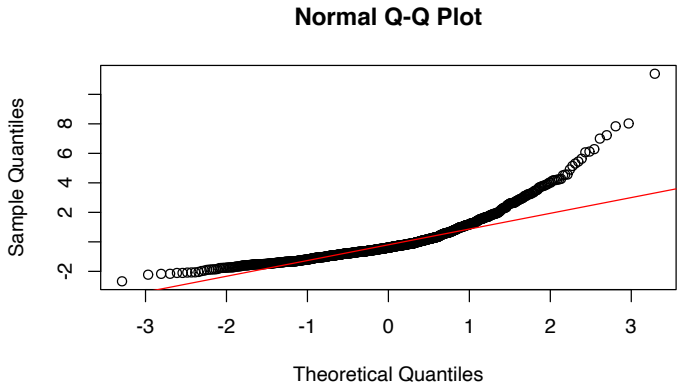
```
qqnorm(residuals(fit), asp = 0)
qqline(residuals(fit), col = "red")
```



Caso no-lineal

```
x <- rnorm(1000)
y <- x^2 + rnorm(1000, sd = 0.5)
fit <- lm(y ~ x)
```

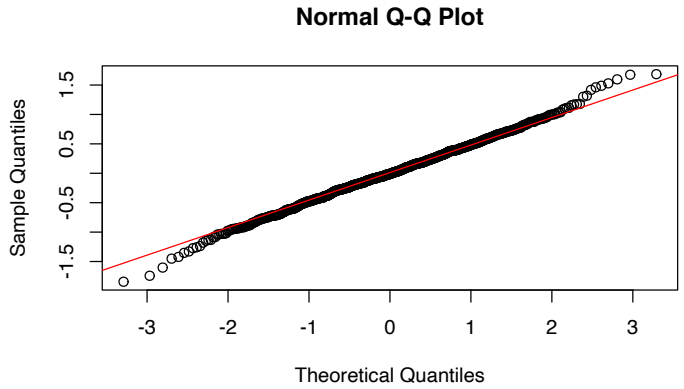
```
qqnorm(residuals(fit), asp = 0)
qqline(residuals(fit), col = "red")
```




```
x <- rnorm(1000)
y <- x^2 + rnorm(1000, sd = 0.5)
fit <- lm(y ~ x + I(x^2))
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.84371 -0.30372 -0.01256  0.32728  1.68466
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01010    0.01979   0.511   0.6098
## x           -0.03009    0.01587  -1.896   0.0582 .
## I(x^2)       0.99172    0.01196  82.906   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5001 on 997 degrees of freedom
## Multiple R-squared:  0.8734, Adjusted R-squared:  0.8731
## F-statistic:  3438 on 2 and 997 DF,  p-value: < 2.2e-16
```

```
qqnorm(residuals(fit), asp = 0)
qqline(residuals(fit), col = "red")
```



4.7.1.4. Multicolinealidad

Hay dos formas de detectar multicolinealidad

1. Analizar la matriz de correlaciones de las variables (solamente detecta colinealidad entre pares).
2. Analizar la correlación multiple entre un predictor y el resto.

Defina $R^2_{X_j|X_{-j}}$ como el R^2 de la regresión multiple entre X_j vs el resto de covariables.

Si $R^2_{X_j|X_{-j}}$ es cercano a 1 entonces hay alta correlación entre X_j y el resto.

Defina el factor de inflación de la varianza como:

$$\text{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

Si VIF es alto

- Quitar las variables
- Combinar variables

Hay muchos paquetes que tienen implementado la función `vif` (car, rms, entre otros).

Caso variables colineales

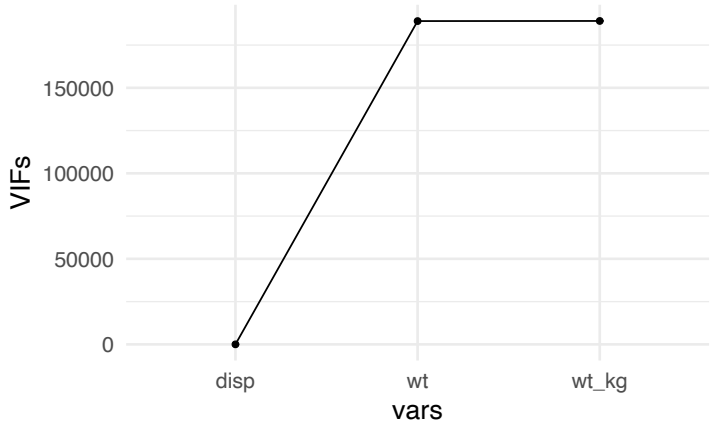
La variable `wt` está en unidades de 1000lb. La convertimos a Kilogramos.

```
mtcars_kg <- mtcars %>%
    mutate(wt_kg = wt * 1000 * 0.4535 + rnorm(32))


fit_kg <- lm(mpg ~ disp + wt + wt_kg, data = mtcars_kg)
summary(fit_kg)
```

```
##
## Call:
## lm(formula = mpg ~ disp + wt + wt_kg, data = mtcars_kg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0609 -1.8566 -0.6442  1.1658  6.1471
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.263681   2.143546  16.918 3.11e-16 ***
## disp         -0.016980   0.008712  -1.949   0.0614 .
## wt          455.378192 220.448899   2.066   0.0482 *
## wt_kg        -1.012338   0.486488  -2.081   0.0467 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.762 on 28 degrees of freedom
## Multiple R-squared:  0.8103, Adjusted R-squared:  0.7899
## F-statistic: 39.86 on 3 and 28 DF,  p-value: 3.079e-10
```

```
library(car)
options(scipen = 1000)

VIFs <- vif(fit_kg)

VIFs <- as.data.frame(VIFs) %>%
    rownames_to_column(var = "vars")

ggplot(VIFs, aes(x = vars, y = VIFs, group = 1)) +
    geom_point() + geom_line() + theme_minimal(base_size = 16)
```



4.7.2. Otros chequeos importantes

4.7.2.1. Puntos extremos

Estos puntos son aquellos que Y_i esta lejos de $\hat{Y}_i$, es decir son puntos en donde los residuos son particularmente muy altos.

Se puede hacer un gráfico de los residuos vs los valores ajustados como en 4.1 y 4.2.

¿Qué tan grande deben ser los residuos?

Solución: Se debe escalar los residuos adecuadamente.

Se construyen los residuos semi-studendizados

$$r_i^s = \frac{e_i}{\sqrt{\text{Var}\,(e_i)}}$$

donde $e_i = Y_i - \hat{Y}_i$. Como $H = X(X^\top X)^{-1}X^\top$ es la matriz de proyección entonces sabemos que

$$\begin{aligned} \hat{Y} &= HY \\ e &= Y - \hat{Y} \end{aligned}$$

Entonces tenemos que

$$\begin{aligned} \text{Var}\,(e) &= \text{Var}\,((I-H)Y) \\ &= (I-H)^2\text{Var}\,(Y) \\ &= (I-H)\sigma^2 \end{aligned}$$

ya que $I - H$ es idempotente. Por lo tanto

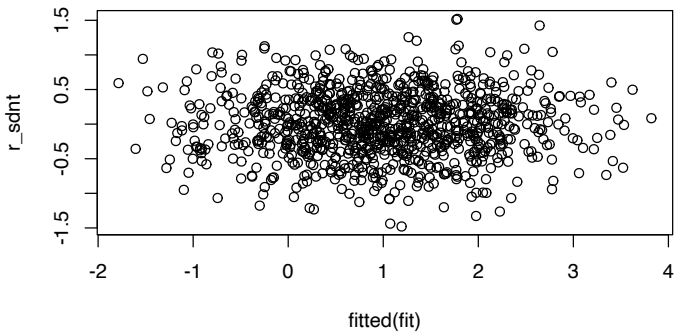
$$\text{Var}\,(e_i) = (1 - h_{ii})\sigma^2$$

Para cada observación se estandarizan los residuos de siguiente forma

$$r_i^s = \frac{e_i}{\sqrt{(1-h_{ii})\sigma^2}}$$

Caso sin valores extremos

```
x <- rnorm(1000)
y <- 1 + x + rnorm(1000, sd = 0.5)
fit <- lm(y ~ x)

X <- model.matrix(y ~ x)
H <- X %*% solve(t(X) %*% X) %*% t(X)
I <- diag(1, nrow = 1000)
I_H <- I - H
r_sdnt <- residuals(fit)/sqrt(diag(I_H) * var(y))
plot(fitted(fit), r_sdnt)
```

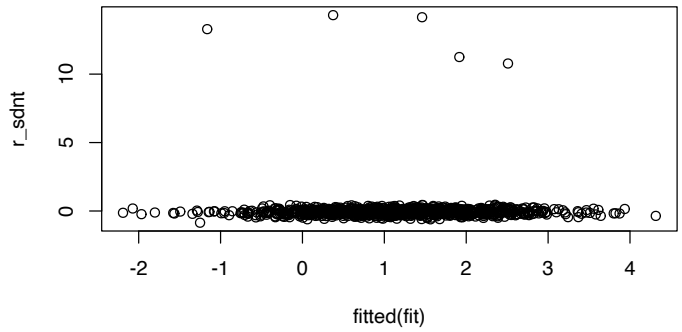



```
fit
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##      0.9893       0.9701
```

Caso con valores extremos

```
x <- rnorm(1000)
y <- 1 + x + rnorm(1000, sd = 0.5)
y[1:5] <- runif(5, 30, 40)
fit <- lm(y ~ x)

X <- model.matrix(y ~ x)
H <- X %*% solve(t(X) %*% X) %*% t(X)
I <- diag(1, nrow = 1000)
I_H <- I - H
r_sdnt <- residuals(fit)/sqrt(diag(I_H) * var(y))
plot(fitted(fit), r_sdnt)
```




```
fit
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##      1.1505       0.9397
```

4.7.2.2. Puntos de apalancamiento (leverage)

Un outlier puede ser detectado pero aún así este puede no afectar el modelo como un todo.

El r_i^s puede ser alto por 2 razones:

1. los residuos e_i son altos (un outlier)
2. el valor h_{ii} es cercano a 1. (Se tiene que $0 \leq h_{ii} \leq 1$).

Los valores donde $h_{ii} \approx 1$ se les denomina de **gran apalancamiento**.

Como la matriz H es de idempotente y de rango completo:

$$\sum_{i=1}^{n} h_{ii} = p + 1 \text{ (Los predictores más el intercepto)}$$

Regla empírica: Si $h_{ii} > \frac{p+1}{n}$ entonces decimos que el punto de **gran apalancamiento**.

Distancia de Cook. La distancia de Cook mide la influencia de las observaciones con respecto al ajuste del modelo lineal con p variables. Esta se define como:

$$D_i = \frac{\sum_{j=1}^{n} (\hat{Y}_j - \hat{Y}_{j(-i)})^2}{(p+1)\sigma^2}$$

donde $\hat{Y}_{j(-i)}$ significa el ajuste del modelo lineal, removiendo la observación i-ésima.

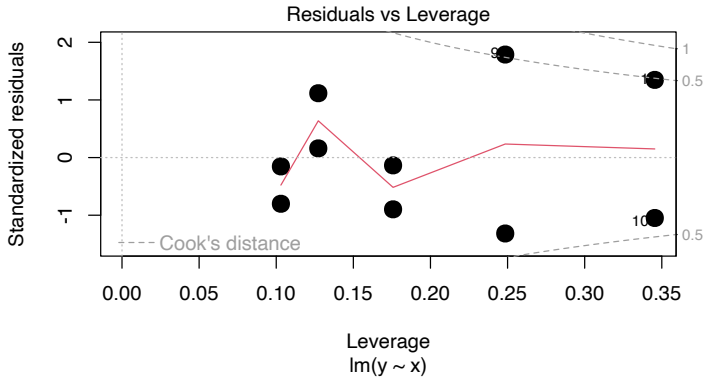
Caso base

```
set.seed(42)
apa_df = data.frame(x = 1:10, y = 10:1 + rnorm(n = 10))
```

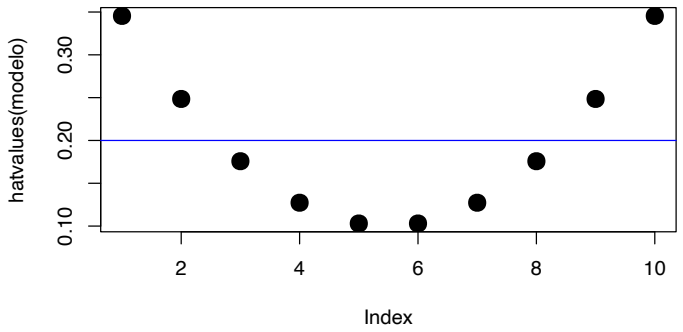
```
modelo <- lm(y ~ x, data = apa_df)
coef(modelo)
```

```
## (Intercept)           x
##  11.3801152  -0.9696033
```

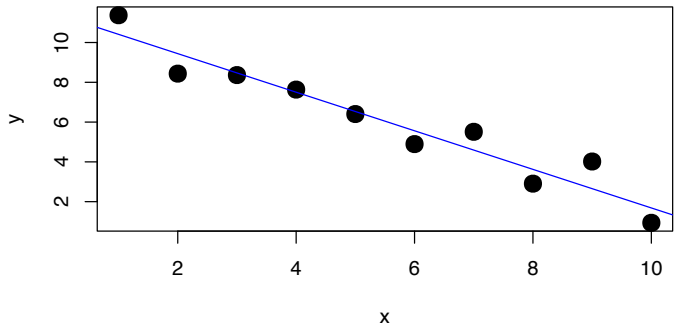
```
plot(modelo, 5, col = c(rep("black", 10), "red"), cex = 2,
    pch = 16)
```




```
plot(hatvalues(modelo), col = c(rep("black", 10), "red"),
    cex = 2, pch = 16)
abline(h = 2/10, col = "blue")
```




```
plot(apa_df, col = c(rep("black", 10), "red"), cex = 2,
    pch = 16)
abline(a = coef(modelo)[1], b = coef(modelo)[2], col = "blue")
```

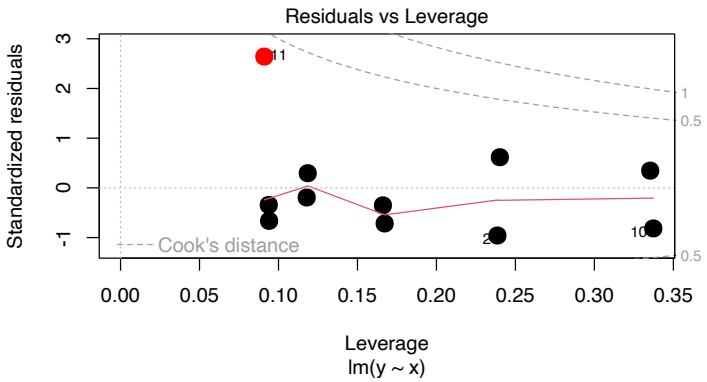


Bajo apalancamiento, residuos grandes, influencia pequeña

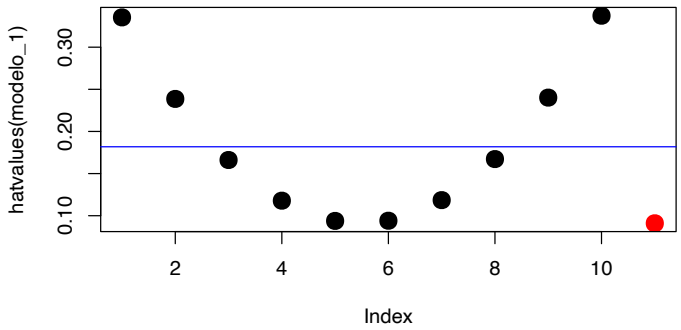
```
p_1 <- c(5.4, 11)
apa_df_1 <- rbind(apa_df, p_1)
modelo_1 <- lm(y ~ x, data = apa_df_1)
coef(modelo_1)
```

```
## (Intercept)           x
##  11.8509232  -0.9749534
```

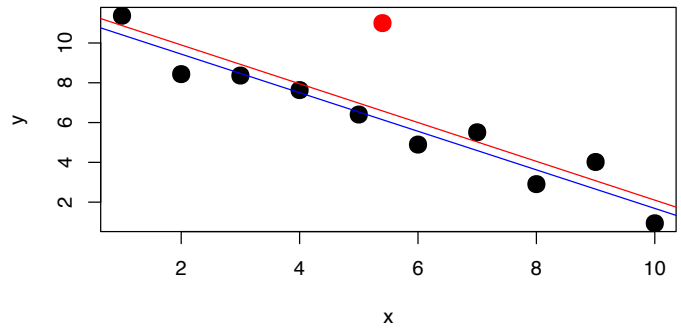
```
plot(modelo_1, 5, col = c(rep("black", 10), "red"),
    cex = 2, pch = 16)
```



```
plot(hatvalues(modelo_1), col = c(rep("black", 10),
    "red"), cex = 2, pch = 16)
abline(h = 2/11, col = "blue")
```




```
plot(apa_df_1, col = c(rep("black", 10), "red"), cex = 2,
    pch = 16)
abline(a = coef(modelo)[1], b = coef(modelo)[2], col = "blue")
abline(a = coef(modelo_1)[1], b = coef(modelo_1)[2],
    col = "red")
```



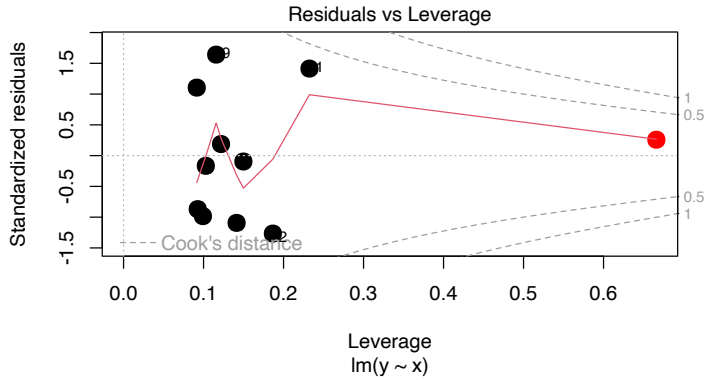
Alto apalancamiento, residuo pequeño, influencia pequeña

```
p_2 <- c(18, -5.7)
apa_df_2 <- rbind(apa_df, p_2)
```

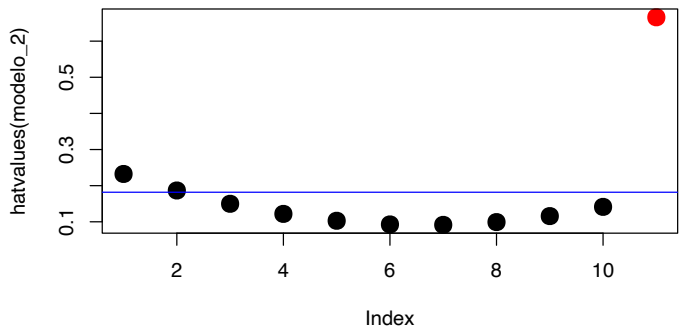
```
modelo_2 <- lm(y ~ x, data = apa_df_2)
coef(modelo_2)
```

```
## (Intercept)           x
##  11.2888153  -0.9507397
```

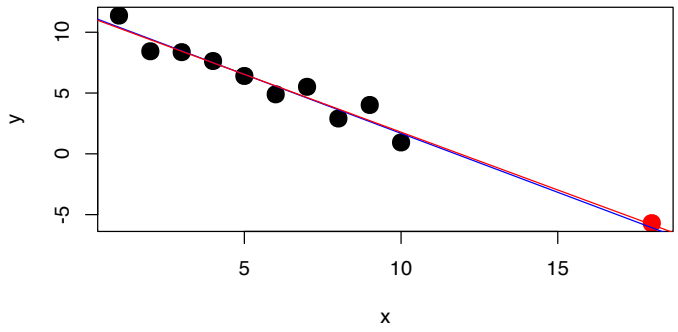
```
plot(modelo_2, 5, col = c(rep("black", 10), "red"),
    cex = 2, pch = 16)
```




```
plot(hatvalues(modelo_2), col = c(rep("black", 10),
    "red"), cex = 2, pch = 16)
abline(h = 2/11, col = "blue")
```



```
plot(apa_df_2, col = c(rep("black", 10), "red"), cex = 2,
    pch = 16)
abline(a = coef(modelo)[1], b = coef(modelo)[2], col = "blue")
abline(a = coef(modelo_2)[1], b = coef(modelo_2)[2],
    col = "red")
```

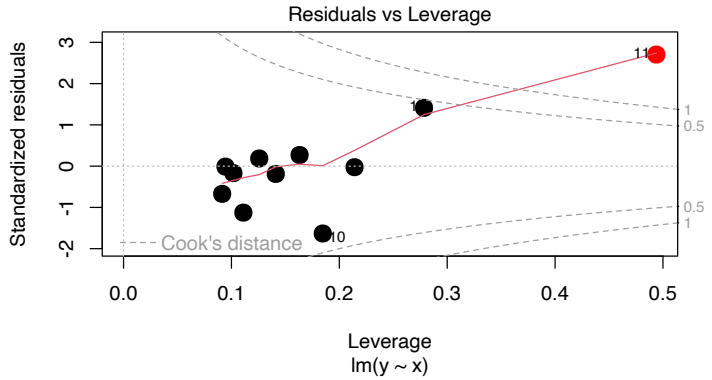


Alto apalancamiento, residuo altos, influencia grande

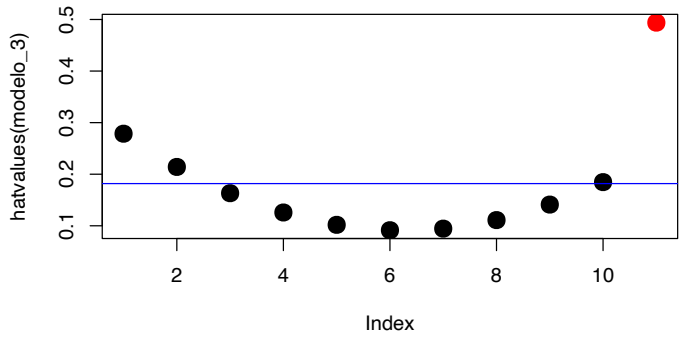
```
p_3 <- c(14, 5.1)
apa_df_3 <- rbind(apa_df, p_3)
modelo_3 <- lm(y ~ x, data = apa_df_3)
coef(modelo_3)
```

```
## (Intercept)           x
##   9.6572209  -0.5892241
```

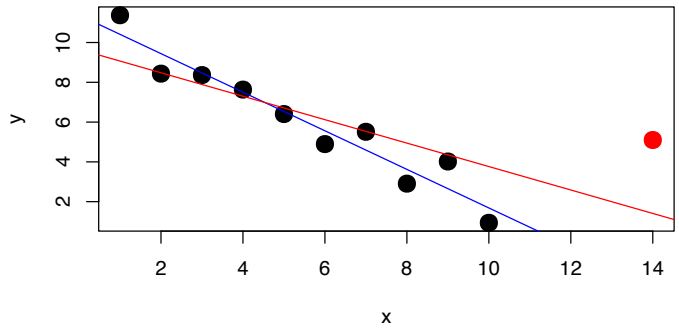
```
plot(modelo_3, 5, col = c(rep("black", 10), "red"),
    cex = 2, pch = 16)
```




```
plot(hatvalues(modelo_3), col = c(rep("black", 10),
    "red"), cex = 2, pch = 16)
abline(h = 2/11, col = "blue")
```

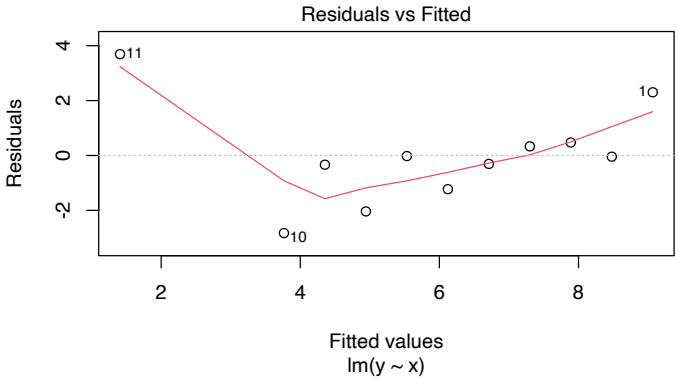


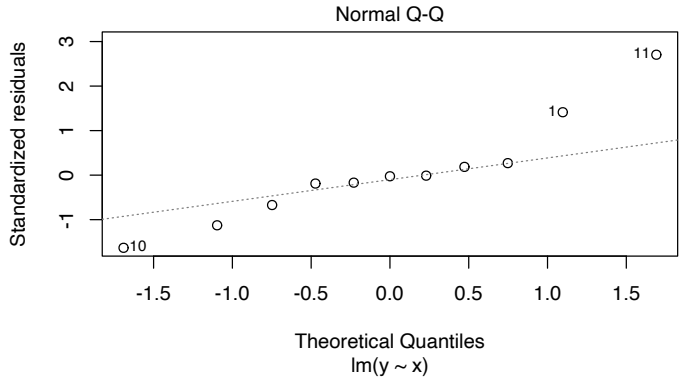

```
plot(apa_df_3, col = c(rep("black", 10), "red"), cex = 2,
    pch = 16)
abline(a = coef(modelo)[1], b = coef(modelo)[2], col = "blue")
abline(a = coef(modelo_3)[1], b = coef(modelo_3)[2],
    col = "red")
```

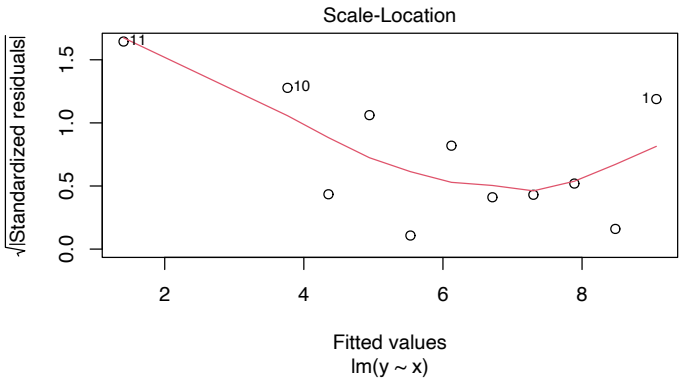


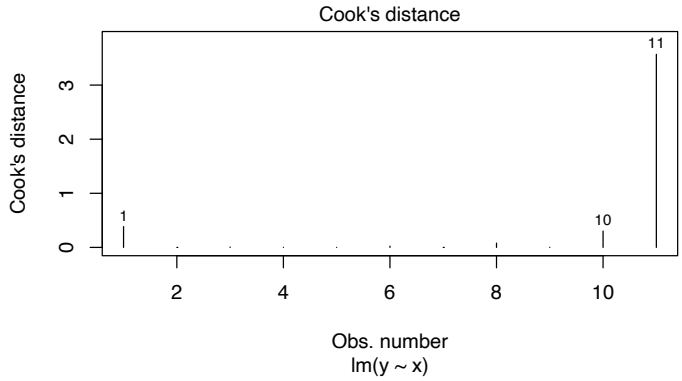

```
`?`(stats:::plot.lm)
```

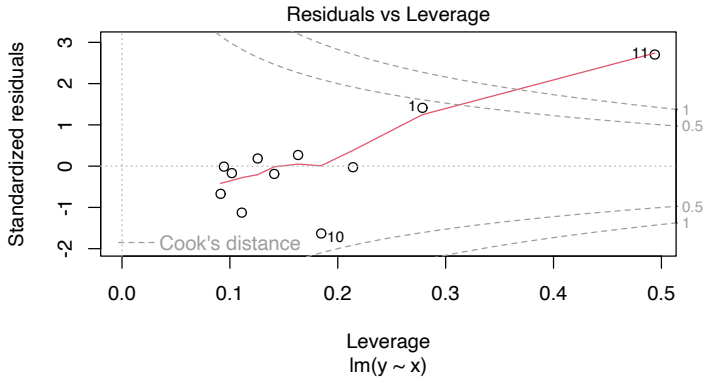
```
plot(modelo_3, which = 1:6)
```

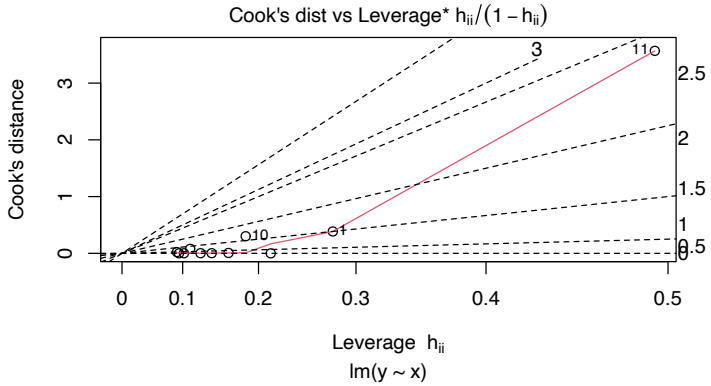



Normal Q-Q
Standardized residuals
-1
0
1
2
3
11
1
10
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(y ~ x)

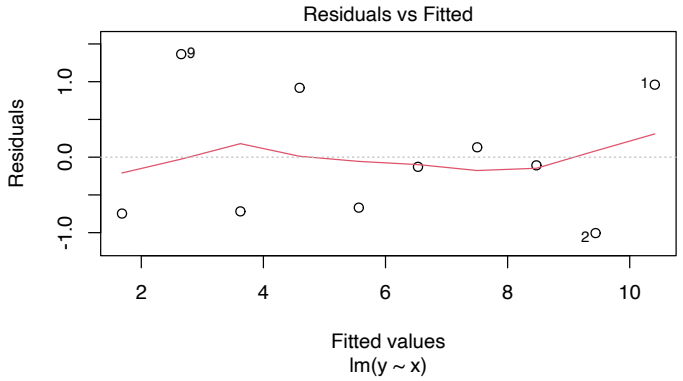

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
11
10
1
2
4
6
8
Fitted values
lm(y ~ x)

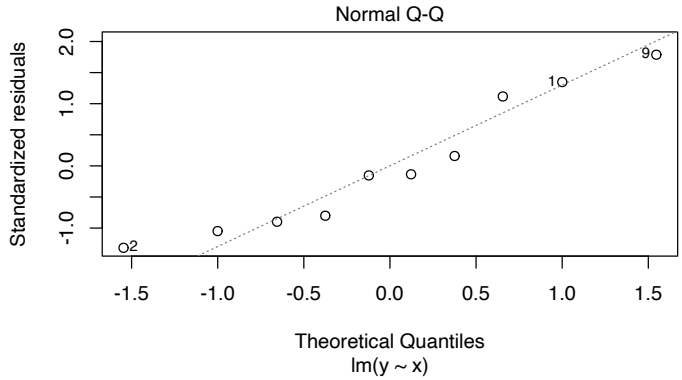

Cook's distance
Cook's distance
0
1
2
3
1
10
11
2
4
6
8
10
Obs. number
lm(y ~ x)

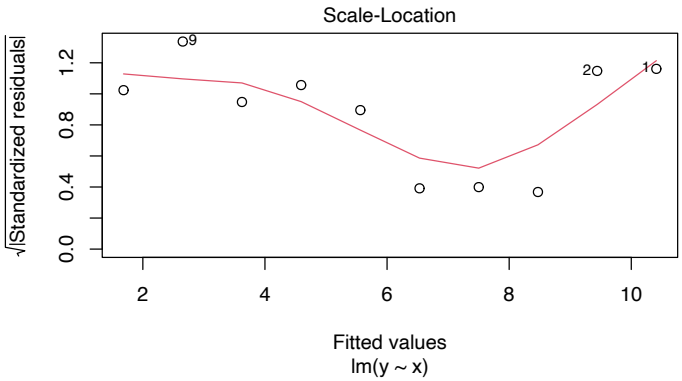


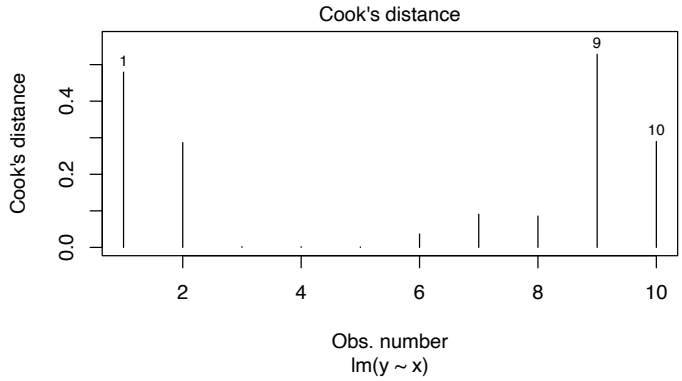


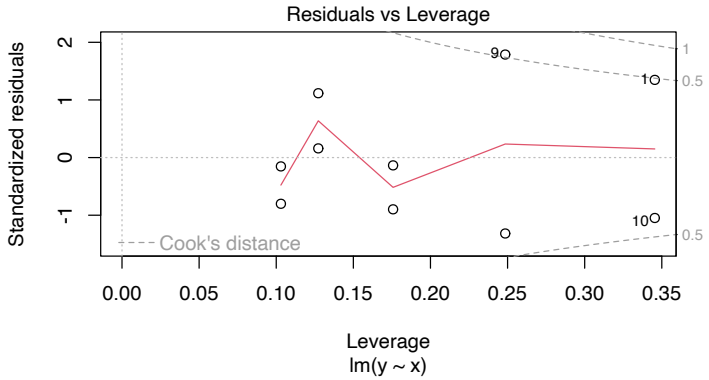

```
plot(modelo, which = 1:6)
```

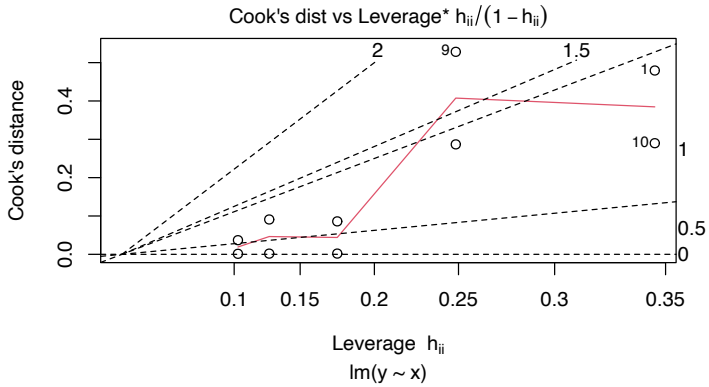



Normal Q-Q
Standardized residuals
-1.0
0.0
1.0
2.0
2
9
10
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(y ~ x)


Scale-Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
9
2
1
2
4
6
8
10
Fitted values
lm(y ~ x)


Cook's distance
Cook's distance
0.0
0.2
0.4
1
9
10
2
4
6
8
10
Obs. number
lm(y ~ x)





4.8. Ejercicios

Del libro (James y col. 2013)

- Capítulo 3: 1, 3, 4, 5, 8, 9

Capítulo 5

Regresión Logística

5.1. Preliminares

Asuma que la variable dependiente Y solo contiene valores 0 o 1 y queremos hacer la regresión:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon.$$

El problema es que $\mathbb{E}[Y|\boldsymbol{X}] = \mathbb{P}(Y = 1|\boldsymbol{X})$ y se debe cumplir que

$$0 \leq \mathbb{E}[Y|\boldsymbol{X}] \leq 1.$$

pero el rango de $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ es todo $\mathbb{R}$.

Solución: Cambiar Y por $g(Y) \in [0, 1]$, donde:

$$g(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)}}$$

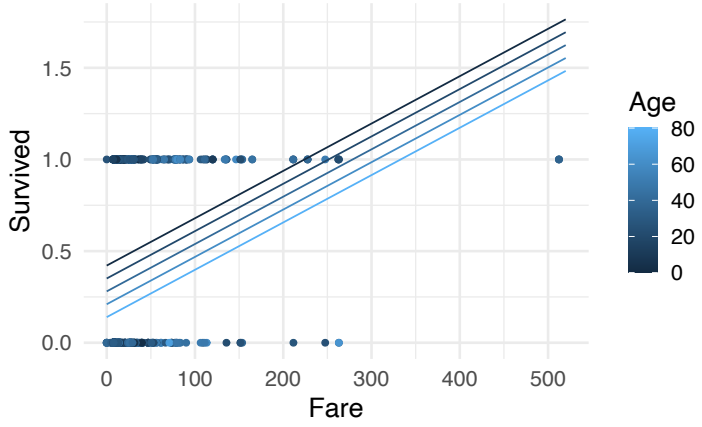
```
titanic <- read.csv("data/titanic.csv")
summary(titanic)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
```

```
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin              Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

```
titanic <- titanic %>%
    select(Survived, Fare, Age) %>%
    drop_na()

fit_lm <- lm(Survived ~ Fare + Age, data = titanic)
```

```
library(ggiraphExtra)
ggPredict(fit_lm) + theme_minimal(base_size = 16)
```



En lugar de esto, definamos el siguiente modelo

$$Y \sim Bernoulli(g_\beta(\boldsymbol{X}))$$

con $g_\beta(\boldsymbol{X}) = \mathbb{P}(Y = 1|\boldsymbol{X})$.

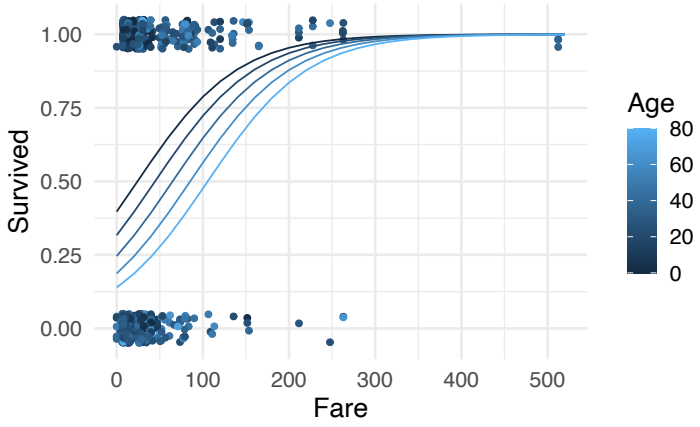
En `R` usaremos la función `glm`

```
fit_glm <- glm(Survived ~ Fare + Age, data = titanic,
    family = "binomial")
summary(fit_glm)
```

```
##
## Call:
## glm(formula = Survived ~ Fare + Age, family = "binomial", data = titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7605  -0.9232  -0.8214   1.2362   1.7820
##
## Coefficients:
##              Estimate Std. Error z value        Pr(>|z|)
## (Intercept) -0.417055   0.185976  -2.243         0.02493 *
## Fare         0.017258   0.002617   6.596 0.0000000000423 ***
## Age         -0.017578   0.005666  -3.103         0.00192 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 964.52  on 713  degrees of freedom
## Residual deviance: 891.34  on 711  degrees of freedom
## AIC: 897.34
##
## Number of Fisher Scoring iterations: 5
```

```
ggPredict(fit_glm) + theme_minimal(base_size = 16)
```



Nota: Existen otros tipos de regresión y estas se definen a través del parámetro `family`. En este curso solo nos enfocaremos en el parámetro `family="binomial"`.

5.1.1. Oportunidad relativa (Odds Ratio)

Defina la oportunidad relativa:

$$O(X) = \frac{g(X)}{1 - g(X)} = e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p} \in [0, 1].$$

como la razón de la probabilidad de obtener un 1 con respecto a la de obtener un 0.

Por ejemplo, suponga que $\mathbb{P}(Y=1|\boldsymbol{X}) = g(\boldsymbol{X}) = 0{,}8$ es la probabilidad de pagar la tarjeta de crédito y $1 - g(\boldsymbol{X}) = 0{,}2$ es la probabilidad de no pagar.

Entonces la oportunidad relativa de pagar la tarjeta es $O(X) = \frac{0{,}8}{0{,}2} = \frac{4}{1}$, lo que se interpreta como que es 4 veces más probable de pagar que no pagar.

5.2. Máxima verosimilitud

Los valores de β se pueden encontrar por máxima verosimilitud.

Defina $p_\beta(\boldsymbol{X}) = \mathbb{P}(Y=1|\boldsymbol{X})$.

La verosimilitud es (donde asumimos sin pérdida de generalidad que $p(\boldsymbol{X}) := p_\beta(\boldsymbol{X})$):

$$L(\beta) = \prod_{i=1}^{n} p(\boldsymbol{X}_i)^{Y_i} (1 - p(\boldsymbol{X}_i))^{1-Y_i}$$

$$\begin{aligned}
\ell(\beta) &= \sum_{i=1}^{n} Y_i \log p(\boldsymbol{X}_i) + (1 - Y_i) \log(1 - p(\boldsymbol{X}_i)) \\
&= \sum_{i=1}^{n} \log(1 - p(\boldsymbol{X}_i)) + \sum_{i=1}^{n} Y_i \log \frac{p(\boldsymbol{X}_i)}{1 - p(\boldsymbol{X}_i)} \\
&= \sum_{i=1}^{n} \log(1 - p(\boldsymbol{X}_i)) + \sum_{i=1}^{n} Y_i (\boldsymbol{X}_i \cdot \beta) \\
&= \sum_{i=1}^{n} -\log\left(1 + e^{\boldsymbol{X}_i \cdot \beta}\right) + \sum_{i=1}^{n} Y_i (\boldsymbol{X}_i \cdot \beta)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ell}{\partial \beta} &= -\sum_{i=1}^{n} \frac{1}{1 + e^{\boldsymbol{X}_i \cdot \beta}} e^{\boldsymbol{X}_i \cdot \beta} \boldsymbol{X}_i + \sum_{i=1}^{n} Y_i \boldsymbol{X}_i \\
&= \sum_{i=1}^{n} (Y_i - p(\boldsymbol{X}_i)) \boldsymbol{X}_i \\
&= X^\top (Y - p(\boldsymbol{X}))
\end{aligned}$$

Solución: Algoritmo de Netwon-Raphson.

Ejercicio 5.1. Muestre que

$$\frac{\partial^2 \ell}{\partial \beta^2} = -\boldsymbol{X} W \boldsymbol{X}$$

donde $W_\beta = \text{diag}\{p(\boldsymbol{X}_i)(1 - p(X_i))\}$.

El algoritmo de Netwon-Raphson usa el hecho que

$$\beta^{(t)} = \beta^{(t-1)} - \left(\frac{\partial^2 \ell}{\partial \beta^2}\right)^{-1} \frac{\partial \ell}{\partial \beta}\bigg|_{\beta^{(t-1)}}$$

Ejercicio 5.2. Muestre que

$$\beta^{(t)} = \left(X^\top W_\beta X\right)^{-1} X^\top Z_\beta,$$

donde $Z_\beta = Z\beta + W_\beta^{-1}(Y - p(X))$ y $\beta = \beta^{(t-1)}$.

A esta técnica se le conoce como **mínimos cuadrados ponderados e iterados** o en inglés **Iteratively Re-Weighted Least Squares** (IRLS).

5.2.1. Resultados adicionales

La suma al cuadrado de los residuos estandarizados se convierte en el estadístico de pearson:

$$\chi^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{p}(X_i))^2}{\hat{p}(X_i)}$$

la cual es una aproximación cuadrática de la devianza (Curso pasado).

$$D = -2\ell(\hat{\beta})$$

Además tenemos los resultados que

- $\hat{\beta} \xrightarrow{\mathbb{P}} \beta$

- $\hat{\beta} \xrightarrow{\mathcal{D}} \mathcal{N}\left(\beta, (X^\top W X)^{-1}\right)$ (Prueba de Wald)
- Se pueden comparar un modelo completo con un reducido a través de pruebas asintóticas LRT:

$$D_c - D_r \overset{H_0}{\sim} = \chi^2_{df_c - df_r}.$$

5.3. Diágnosticos del modelo

Advertencia: La función `glm` no tiene un equivalente de `plot` como en los modelos lineales. De esta forma, si se aplica `plot` a un objeto `glm` solo generará los mismos chequeos que el capítulo anterior. Sin embargo estos podrían estar equivocados si no se leen con cuidado.

5.3.1. Supuesto de linealidad

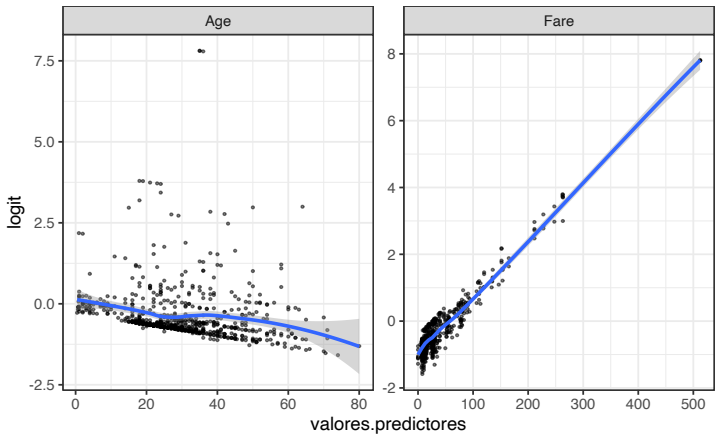
Este supuesto debe ser chequeado con la función logit de las respuestas.

```
fit_glm <- glm(Survived ~ Fare + Age, data = titanic,
    family = "binomial")
summary(fit_glm)
```

```
##
## Call:
## glm(formula = Survived ~ Fare + Age, family = "binomial", data = titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7605  -0.9232  -0.8214   1.2362   1.7820
##
## Coefficients:
##              Estimate Std. Error z value        Pr(>|z|)
## (Intercept) -0.417055   0.185976  -2.243         0.02493 *
## Fare         0.017258   0.002617   6.596 0.0000000000423 ***
## Age         -0.017578   0.005666  -3.103         0.00192 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 964.52  on 713  degrees of freedom
## Residual deviance: 891.34  on 711  degrees of freedom
## AIC: 897.34
##
## Number of Fisher Scoring iterations: 5
```

```
probs <- predict(fit_glm, type = "response")

df <- titanic %>%
    select(Fare, Age) %>%
    mutate(logit = qlogis(probs)) %>%
    pivot_longer(names_to = "predictores", values_to = "valores.predictores",
        -logit)


ggplot(df, aes(valores.predictores, logit)) + geom_point(size = 0.5,
    alpha = 0.5) + geom_smooth(method = "loess") +
    theme_bw() + facet_wrap(~predictores, scales = "free")
```



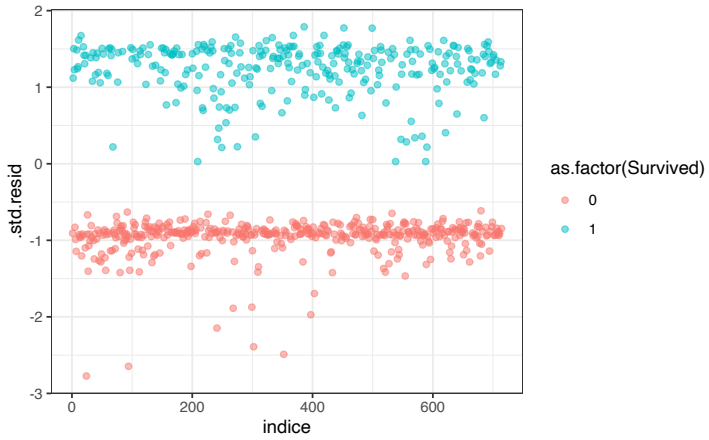
5.3.2. Valores de gran influencia

```
library(broom)
fit_data <- broom::augment(fit_glm) %>%
    mutate(indice = 1:n())
```

```
fit_data %>%
    top_n(3, .cooksd)
```

```
## # A tibble: 3 x 10
##   Survived  Fare   Age .fitted .resid .std.resid    .hat .sigma .cooksd indice
##      <int> <dbl> <dbl>   <dbl>  <dbl>      <dbl>   <dbl>  <dbl>   <dbl>  <int>
## 1        0  263     19    3.79  -2.76      -2.77 0.00862   1.12   0.129     24
## 2        0  248.    24    3.43  -2.63      -2.65 0.0103    1.12   0.109     94
## 3        0  263     64    3.00  -2.47      -2.49 0.0171    1.12   0.118    352
```

```
ggplot(fit_data, aes(indice, .std.resid)) + geom_point(aes(color = as.factor(Survived
    alpha = 0.5) + theme_bw()
```




```
fit_data %>%
    filter(abs(.std.resid) > 3)
```

```
## # A tibble: 0 x 10
## # ... with 10 variables: Survived <int>, Fare <dbl>, Age <dbl>, .fitted <dbl>,
## #   .resid <dbl>, .std.resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,
## #   indice <int>
```

5.3.3. Multicolinealidad

```
car::vif(fit_glm)
```

```
##     Fare      Age
```

```
## 1.033878 1.033878
```

5.4. Predicción y poder de clasificación

La capacidad predictiva de un modelo de clasificación como el de regresión logística se debe medir conforme a la naturaleza de la variable dependiente. Primero recordemos que el modelo predictivo en este caso estaría definido por:

$$\hat{p}(X) = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p)}}$$

donde los β's son estimados usando IRLS.

Ahora imaginemos que tenemos un conjunto de datos nuevo $(X_1^*, \ldots, X_p^*)$ y queremos ver que tipo de respuesta Y^* obtenemos (0 o 1) para este conjunto de datos.

Obviamente nuestro modelo puede equivocarse y darnos una respuesta errónea. Por ejemplo digamos que en el caso del `titanic` uno esperaría que personas más jóvenes y que hayan pagado más por su tiquete tengan mayor probabilidad de sobrevivencia.

Entonces tenemos realmente 4 opciones

		Clase	**Predicción**	
		0	1	
Clase	0	Verdaderos Negativos. (TN)	Falsos Positivos (FP)	N
Real	1	Falsos Negativos (FN)	Verdaderos Positivos (TP)	P
	Total	N^*	P^*	

```
predict_numeric <- predict(fit_glm, type = "response")
predict_01 <- as.numeric(predict_numeric >= 0.5)

matriz_confusion <- table(titanic$Survived, predict_01)
```

```
colnames(matriz_confusion) <- c("N", "P")
rownames(matriz_confusion) <- c("N", "P")

matriz_confusion
```

```
##    predict_01
##       N   P
##   N 380  44
##   P 201  89
```

Noten que se utilizó un umbral de .5 como separador de que un evento genera un 1 o un 0 a nivel de predicción. Para entender la siguiente tabla vamos a definir los siguientes términos:

Exactitud (Accuracy) Es la tasa de que un individuo esté bien identificado por el modelo de clasificación $(TP + TN)/(TP + TN + FN + FP)$.

Precisión Es la tasa de elementos identificados como 1 de forma correcta con respecto a los que fueron identificados con un valor de 1 $Precisión = TP/P^*$

Sensibilidad (Exhaustividad) Es la tasa de elementos identificados como 1 de forma correcta con respecto a los que realmente son 1. $Sensibilidad = TP/P$

F-Score Es la media armónica entre la precisión y la sensibilidad. $F-Score = 2 * (Sensibilidad * Precisión)/(Sensibilidad + Precisión)$

Especificidad Es la tasa de elementos identificados correctamente como 0 que realmente estaban etiquetados como 0.

Entonces esto nos da las siguientes posibilidades.

Tipo	Cálculo	Sinónimos
Tasa Falsos Positivos	FP/N	Error Tipo I, 1-Especificidad
Tasa Verdaderos Positivos	TP/P	1-Error Tipo II, Poder, Sensibilidad, Exhaustividad (Recall)
Valor de Predicción Positivos	TP/P^*	Precisión, 1 - Proporción de Falsos Descubrimientos

Tipo	Cálculo	Sinónimos
Valor de Predicción Negativos	TN/N^*	
F-Score	$\frac{2(TP/P^* \times TP/P)}{(TP/P^* + TP/P)}$	

Nota:

- Exactitud es un buen indicador cuando los datos son simétricos (igual número de FP y FN).
- F-Scores es un mejor indicador cuando los datos son asimétricos
- La precisión nos permite describir la capacidad del modelo de predecir verdaderos positivos.
- La sensibilidad nos permite describir la capacidad de categorizar los verdaderos positivos de forma correcta.

En un modelo se debe escoger entre sensibilidad y precisión de acuerdo a ciertas ideas:

- **Sensibilidad** es importante si la ocurrencia de **falsos negativos** es inaceptable. Por ejemplo en el caso de pruebas clínicas. Posiblemente obtener falsos positivos en este caso es aceptable.
- **Precisión** es importante si se quiere estar más seguro de los **verdaderos positivos**. Por ejemplo detectar **spam** en correos electrónicos.
- **Especificidad** es importante si lo que se quiere es cubrir todos los **verdaderos negativos**, es decir, que no se quieren falsas alarmas. Por ejemplo se hacen pruebas de detección de drogas y si es positivo va a la cárcel. Los **falsos positivos** son intolerables.

```
(TN <- matriz_confusion["N", "N"])
```

```
## [1] 380
```

```
(TP <- matriz_confusion["P", "P"])
```

```
## [1] 89
```

```
(FP <- matriz_confusion["N", "P"])
```

```
## [1] 44
```

```
(FN <- matriz_confusion["P", "N"])
```

```
## [1] 201
```

```
(exactitud <- (TP + TN)/(TP + TN + FP + FN))
```

```
## [1] 0.6568627
```

```
(precision <- TP/(TP + FP))
```

```
## [1] 0.6691729
```

```
(sensibilidad <- TP/(TP + FN))
```

```
## [1] 0.3068966
```

```
(F_score <- 2 * (precision * sensibilidad)/(precision +
    sensibilidad))
```

```
## [1] 0.4208038
```

```
(especificidad <- TN/(TN + FP))
```

```
## [1] 0.8962264
```

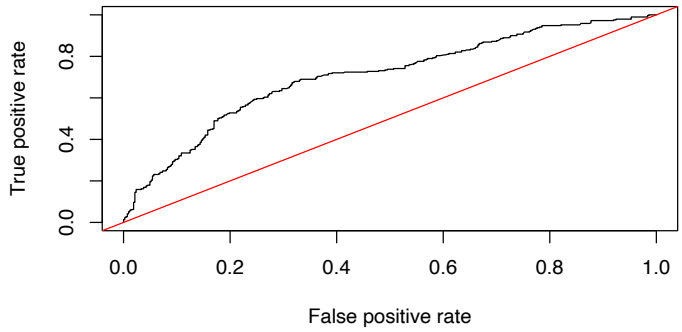
5.4.1. Curva ROC

Un excelente clasificador debería detectar correctamente los **verdaderos positivos (TP)** e ignorar los **falsos positivos (FP)**. Puesto de otra forma, si el clasificador es malo, los **verdaderos positivos** serían indistingibles de los **falsos positivos**.

La curva ROC (Receiver Operation Curve) grafica la Tasa Falsos Positivos vs Sensibilidad del modelo. Y el estadístico AUC mide el área bajo la curva ROC.

```
library(ROCR)

logist.pred.ROCR <- prediction(predict_numeric, titanic$Survived)

logist.perf <- performance(logist.pred.ROCR, "tpr",
    "fpr")
```

```
plot(logist.perf)
abline(0, 1, col = "red")
```




```
auc <- performance(logist.pred.ROCR, measure = "auc")

auc@y.values
```

```
## [[1]]
## [1] 0.7063313
```

Nota:

Hasta este momento estamos verificando el poder de clasificación del modelo con los mismos datos que usamos para ajustarlo. Es decir, le estamos diciendo al modelo que compruebe la veracidad de la clasificación que ya se hizo previamente.

Esto es incorrecto, ya que el modelo ya sabe "las respuestas" y no estamos midiendo su poder de clasificación sino más bien su capacidad predictiva dentro del conjunto de entrenamiento.

Para resolver esto, debemos tomar otra muestra de prueba (**training**) que nos diga si el ajuste que hicimos es correcto.

```
titanic$id <- 1:nrow(titanic)

train <- titanic %>%
```

```
    sample_frac(0.75)

test <- titanic %>%
    anti_join(train, by = "id")
```

```
fit_glm <- glm(Survived ~ Fare + Age, data = train,
    family = "binomial")
```

```
predict_numeric <- predict(fit_glm, newdata = test,
    type = "response")
predict_01 <- as.numeric(predict_numeric >= 0.5)

matriz_confusion <- table(test$Survived, predict_01)

colnames(matriz_confusion) <- c("N", "P")
rownames(matriz_confusion) <- c("N", "P")

matriz_confusion
```

```
##    predict_01
##      N  P
##   N 93  9
##   P 57 19
```

```
(TN <- matriz_confusion["N", "N"])
```

```
## [1] 93
```

```
(TP <- matriz_confusion["P", "P"])
```

```
## [1] 19
```

```
(FP <- matriz_confusion["N", "P"])
```

```
## [1] 9
```

```
(FN <- matriz_confusion["P", "N"])
```

```
## [1] 57
```

```
(exactitud <- (TP + TN)/(TP + TN + FP + FN))
```

```
## [1] 0.6292135
```

```
(precision <- TP/(TP + FP))
```

```
## [1] 0.6785714
```

```
(sensibilidad <- TP/(TP + FN))
```

```
## [1] 0.25
```

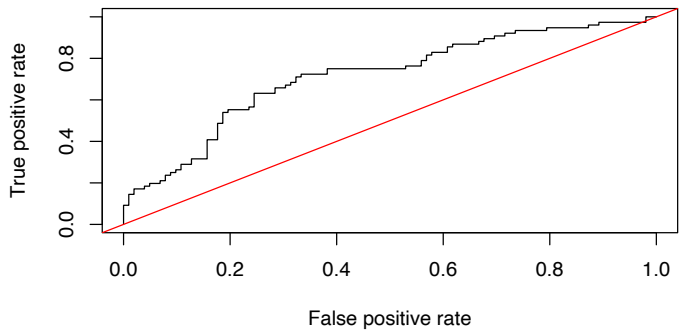
```
(F_score <- 2 * (precision * sensibilidad)/(precision +
    sensibilidad))
```

```
## [1] 0.3653846
```

```
(especificidad <- TN/(TN + FP))
```

```
## [1] 0.9117647
```

```
logist.pred.ROCR <- prediction(predict_numeric, test$Survived)

logist.perf <- performance(logist.pred.ROCR, "tpr",
    "fpr")

plot(logist.perf)
abline(0, 1, col = "red")
```



```
auc <- performance(logist.pred.ROCR, measure = "auc")

auc@y.values
```

```
## [[1]]
## [1] 0.7138803
```

5.5. Ejercicios

- Del libro (James y col. 2013):
  - Capítulo 4: 1, 6, 10, 11. (En esta sección no vimos LDA, QDA ni k-vecinos más cercanos, así que ignoren esas partes y concentrense en hacer los análisis correctos para regresión logística).

Bibliografía

Efron, B. (ene. de 1979). «Bootstrap Methods: Another Look at the Jackknife». En: *The Annals of Statistics* 7.1, págs. 1-26.

Hall, Peter (dic. de 1987). «On Kullback-Leibler Loss and Density Estimation». En: *The Annals of Statistics* 15.4, págs. 1491-1519.

Härdle, Wolfgang y col. (2004). *Nonparametric and Semiparametric Models*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hastie, Trevor, Robert Tibshirani y Jerome Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

James, Gareth y col. (2013). *An Introduction to Statistical Learning*. Vol. 103. New York, NY: Springer New York.

Quenouille, M. H. (ene. de 1949). «Approximate Tests of Correlation in Time-Series». En: *Journal of the Royal Statistical Society: Series B (Methodological)* 11.1, págs. 68-84.

Wasserman, Larry (2006). *All of Nonparametric Statistics*. New York, NY: Springer New York.