

TRABAJO FINAL DE MÁSTER

Título: *Interpolación de superficies de volatilidad mediante splines cúbicos*

Autoría: Minsu-Luis Cho Mun

Tutoría: Luis Ortiz Gracia

Curso académico: 2020-2021

Facultad de Economía y Empresa

Universidad de Barcelona

Trabajo Final de Máster

Máster en Ciencias Actuariales y Financieras

Interpolación de superficies de volatilidad mediante splines cúbicos

Autoría: Minsu-Luis Cho Mun

Tutoría: Luis Ortiz Gracia

Resumen

El presente trabajo tiene como objetivo describir, implementar y validar los métodos de interpolación de splines cúbicos univariado y bivariado en superficies de volatilidad. La superficie de volatilidad es la herramienta esencial para que los participantes del mercado puedan valorar opciones que no cotizan en los mercados organizados. El trabajo se fundamenta en el modelo generalizado de Black-Scholes-Merton, el modelo más estandarizado para la valoración de opciones europeas. Mediante métodos de búsqueda de raíces utilizaremos la ecuación de Black-Scholes para obtener las volatilidades implícitas del índice de opciones europeas de renta variable SPX, cuyo subyacente es el índice S&P 500. Se muestra como se construye una superficie de volatilidad paso a paso y posteriormente usaremos los métodos de interpolación para extraer volatilidades implícitas no cotizadas. Finalmente se validarán ambos métodos de interpolación con técnicas de Machine Learning.

Abstract

The objective of this paper is to describe, implement and validate both the univariate and bivariate cubic spline interpolation methods on volatility surfaces. The volatility surface is the essential tool for market participants to value options that are not traded on organized markets. The work is based on the generalized Black-Scholes-Merton model, the most standardized model for valuation of european options. Using numerical methods for solving root-finding problems we will use the Black-Scholes equation to obtain the implied volatilities of the SPX european options index, whose underlying is the S&P 500 index. We will show how to construct a volatility surface step by step and later we will use the interpolation methods to extract unquoted implied volatilities. Finally, both interpolation methods will be validated with Machine Learning techniques.

El contenido de este documento es de exclusiva responsabilidad del autor, quien declara que no ha incurrido en plagio y que la totalidad de referencias a otros autores han sido expresadas en el texto.

Índice

1. Introducción	7
1.1. Base de datos	7
2. Sonrisa de volatilidad	8
2.1. Moneyness	8
2.2. Características principales de la sonrisa de volatilidad	9
2.3. Construcción de la sonrisa de volatilidad	11
3. Modelo de Black-Scholes	14
3.1. Delta Hedging en la práctica	16
3.2. Paridad Put-Call	17
4. Métodos numéricos de búsqueda de raíces	18
4.1. Bisección	19
4.2. Newton-Raphson	21
5. Métodos de interpolación	23
5.1. Interpolación univariada de splines cúbicos	23
5.2. Interpolación bivariada de splines cúbicos	26
6. Validación de los métodos de búsqueda de raíces	28
7. Validación de la interpolación de splines cúbicos univariada	29
8. Validación de la interpolación de splines cúbicos bivariada	31
9. Conclusión	33
10. Anexos	35
10.1. Interpolación univariada de splines cúbicos	35
10.2. Interpolación bivariada de splines cúbicos	36
10.3. Error cuadrático medio	38
10.4. Coeficiente de determinación (R cuadrado)	39
10.5. Validación cruzada dejando uno fuera	40
10.6. Código R	41
11. Bibliografía	57

“La volatilidad implícita es el valor incorrecto que insertado en la fórmula incorrecta nos da un precio de opción correcto”

—Riccardo Rebonato

1. Introducción

El modelo de Black Scholes está condicionado bajo unas hipótesis económicas que no son muy representativas del entorno económico real, sin embargo éste es el modelo utilizado habitualmente en los mercados financieros. El principal motivo de confusión, es que el modelo se basa en el supuesto de que la volatilidad es constante hasta el vencimiento de la opción. Cómo se aplica la fórmula de Black Scholes sabiendo que en realidad la volatilidad varía significativamente durante la vida del derivado? Si este supuesto no se cumple, la valoración mediante la fórmula de Black Scholes no debería ser errónea?

Tengamos en cuenta el principio de la parsimonia, cuantos mas parámetros haya que calcular en un modelo, más probabilidad de cometer errores habrá. Es conveniente encontrar un equilibrio entre la sofisticación y el valor añadido que ello conlleva.

Toda modelización será siempre una simplificación de la realidad. La fórmula de Black-Scholes es simple y depende de un único parámetro, la volatilidad σ . Una fórmula simple tiene algunas ventajas, es fácil de entender y aplicar. Pero lo más importante es que también es fácil darse cuenta de sus limitaciones. Se pueden introducir ajustes en el modelo para complementar los riesgos y costes no considerados en él y obtener una valoración más precisa del derivado.

Un aspecto importante de la fórmula de Black-Scholes es que se ha convertido en una convención [Neftci(2008)]. El modelo vincula una cotización de la volatilidad a un valor en dólares del derivado. Esto ayuda a desarrollar plataformas comunes para estrategias de coberturas, gestión de riesgos y el trading de volatilidad. Por lo tanto, la cotización de referencia no es el precio del derivado si no el valor que le asigne al parámetro σ el mercado. Es habitual incorporar los riesgos y costes no considerados en el modelo de Black-Scholes reajustando el parámetro de volatilidad, en forma de prima de riesgo. Por tanto, la sonrisa y la superficie de volatilidad, contienen información muy valiosa sobre las volatilidades implícitas y son herramientas esenciales para la negociación y valoración de derivados, y para acometer estrategias de cobertura y arbitraje.

1.1. Base de datos

Basaremos este trabajo en los datos obtenidos a través de Yahoo Finance del índice de opciones SPX que se negocia en el Chicago Board Options Exchange (CBOE), el mayor mercado organizado de opciones de los Estados Unidos. El subyacente de este derivado es el índice Standard & Poor's 500 (S&P 500), que se constituye de las 500 empresas con más capitalización bursátil de Estados Unidos, cotiza en las bolsas New York Stock Exchange (NYSE) y NASDAQ y captura aproximadamente el 80 % de toda la capitalización de mercado en Estados Unidos. Al S&P 500 se le considera el índice que mejor representa la situación real del mercado. El SPX opera con opciones de tipo europeo que sólo pueden ser ejercidas en fecha de vencimiento y además no reparte dividendos y la liquidación es en efectivo.

Como en la mayoría de mercados, en el de opciones se designan a los market-makers (bancos de inversión, brokers institucionales u otros miembros de bolsas organizadas). Su presencia en el mercado es facilitar el trading y proporcionar liquidez. Ellos proveen al mercado de cotizaciones de oferta (Bid) y demanda (Ask). Ganan dinero comprando a precios Bid y vendiendo a precios

Ask. El margen entre los precios de venta y los precios de compra (Bid-Ask spread) es la fuente de su beneficio.

En el mercado de opciones, el Bid-Ask spread es mucho mayor que en mercados de renta variable, lo cual refleja el riesgo de apalancamiento, inherente en productos derivados. Esto también es debido a la relativa iliquidez de las opciones respecto a los valores de renta variable [Pan(2019)].

Los market-makers están en continua exposición al riesgo de mercado al no poder tener una cobertura perfecta sobre sus posiciones abiertas de compra y venta de opciones en sus carteras. En el mercado de opciones es típico que los market-makers minimicen su exposición sobre el subyacente mediante la denominada cobertura delta (delta-hedging), que explicaremos en el capítulo 3.

Al contrario que las cotizaciones de precios de valores de renta variable, tener acceso a cotizaciones actuales de opciones con sus respectivas volatilidades implícitas suele tener un coste. El código creado en R nos permite automatizar la descarga de datos sin ningún coste, para ello se han usado la librerías `quantmod` y `rvest` de R para extraer los datos del índice de opciones SPX de Yahoo Finance. Para sincronizar en la medida de lo posible el precio del subyacente con las volatilidades implícitas mostradas en Yahoo Finance y con las que calcularemos nosotros con los métodos de búsqueda de raíces, bisección y Newton-Raphson hemos utilizado los datos de las opciones cotizadas durante el día 08 de Enero de 2021. Para acotar nuestra base de datos hemos limitado el ratio moneyness (strike/spot) a $0,75 - 1,25$ y los días de vencimiento a $7 - 180$ días.

En la figura 1 se muestra un extracto de los datos obtenidos mediante web scraping de Yahoo Finance.

```
> head(calls)
  contract name      last trade date strike last price  bid  ask change % change volume open interest implied volatility ttm moneyness
208 SPXW210115C03450000 2021-01-08 9:41AM EST 3450 374.32 366.70 370.90 19.48 +5.49% 10 238 43.87% 7 0.9043582
215 SPXW210115C03485000 2021-01-08 10:29AM EST 3485 319.50 332.10 336.20 0.00 - 5 65 40.78% 7 0.9135329
220 SPXW210115C03510000 2021-01-08 9:39AM EST 3510 312.64 308.80 312.70 22.70 +7.83% 81 1,165 40.19% 7 0.9200862
233 SPXW210115C03575000 2021-01-08 10:43AM EST 3575 244.01 244.70 248.60 56.74 +30.30% 3 415 34.25% 7 0.9371248
238 SPX210115C03600000 2021-01-08 9:36AM EST 3600 222.36 217.40 221.70 16.16 +7.84% 300 19,651 29.53% 7 0.9436781
240 SPX210115C03610000 2021-01-08 10:48AM EST 3610 209.76 208.60 213.40 8.03 +3.98% 2 267 30.21% 7 0.9462995
> tail(puts)
  contract name      last trade date strike last price  bid  ask change % change volume open interest implied volatility ttm moneyness
3684 SPX210618P03500000 2021-01-08 10:49AM EST 3500 117.18 117.80 118.50 -2.82 -2.35% 1,112 11,873 25.22% 161 0.9174649
3688 SPX210618P03550000 2021-01-08 10:57AM EST 3550 128.04 128.90 129.70 -4.25 -3.21% 3 8,096 24.52% 161 0.9305715
3692 SPX210618P03600000 2021-01-08 9:31AM EST 3600 139.80 140.90 141.60 -5.00 -3.45% 12 6,869 23.77% 161 0.9436781
3713 SPX210618P03850000 2021-01-08 10:48AM EST 3850 220.64 220.10 220.90 -34.29 -13.45% 1 173 19.99% 161 1.0092113
3821 SPXW210630P03800000 2021-01-08 10:07AM EST 3800 207.32 209.50 210.70 -31.12 -13.05% 500 1,223 20.86% 173 0.9961047
3822 SPXW210630P03825000 2021-01-08 10:52AM EST 3825 218.00 219.40 220.40 -8.20 -3.63% 500 1,502 20.53% 173 1.0026580
```

Figura 1: Data Frame de Calls y Puts del índice de opciones SPX

2. Sonrisa de volatilidad

2.1. Moneyness

En los mercados de opciones, con el fin de estandarizar las cotizaciones de diferentes volatilidades implícitas para diferentes niveles de strikes, se emplean medidas de moneyness para relacionar el strike K y el precio del activo subyacente S_t . Cuanto más alejado estén los strikes K del precio spot S_t del subyacente, menos volumen de contratos negociados suele haber. El ratio Moneyness, nos muestra cuan alejado se encuentra el strike del valor At-The Money.

El ratio moneyness siempre se interpreta desde la perspectiva del comprador de la opción al ser la parte contratante que tiene el derecho a ejercerla.

$$\text{Moneyness Call} = \frac{S_t}{K} \quad \text{Moneyness Put} = \frac{K}{S_t}$$

ATM = At-The-Money (Moneyness = 1)

NTM = Near-The-Money (Moneyness \approx 1)

OTM = Out-of-The-Money (Moneyness $<$ 1)

Deep-OTM = Deep Out-of-The-Money (Moneyness \ll 1)

ITM = In-The-Money (Moneyness $>$ 1)

Deep-ITM = Deep-In-The-Money (Moneyness \gg 1)

2.2. Características principales de la sonrisa de volatilidad

Antes del crash bursátil de 1987, el modelo de valoración de opciones Black-Scholes describía razonablemente bien los mercados de opciones. La sonrisa de volatilidad en mercados de índices de acciones era prácticamente plana en relación al strike, tal y como se supone en el modelo. Desde 1987, las volatilidades implícitas en mercados de índices de opciones muestran una forma sesgada violando la hipótesis del modelo de Black-Scholes. A continuación se resumen las características más relevantes de la sonrisa de volatilidad según [Derman and Miller(2016)].

- La característica más notable en la sonrisa de volatilidad en índices de equities es su pendiente negativa en función del strike como se ilustra en la figura 2. Generalmente tiende a ser más pronunciada para vencimientos cortos y a aplanarse a medida que aumentan los precios de ejercicio.

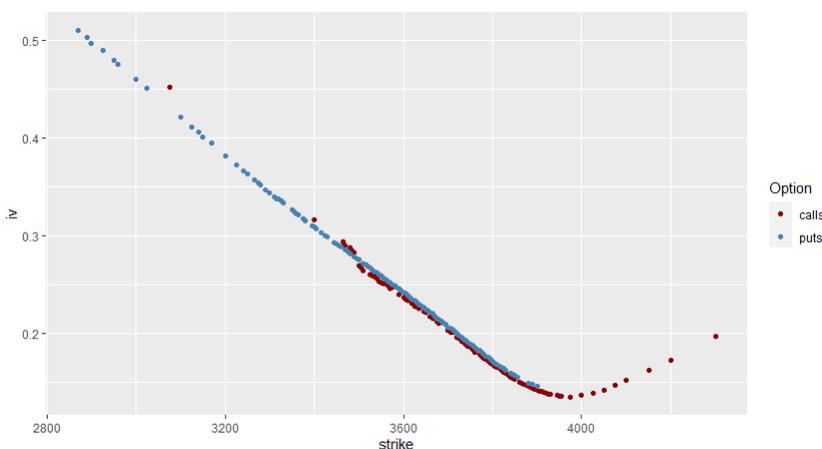


Figura 2: Smiles de un call del SPX para el vencimiento 25 de Enero 2021

- Las volatilidades implícitas suelen alcanzar un mínimo alrededor de los strikes ATM, y luego aumentan para strikes más elevados.

- El sesgo negativo se debe en parte a la asimetría de los movimientos del índice. Grandes movimientos negativos son mucho más frecuentes que grandes movimientos positivos. Dada la dificultad de cubrirse ante desplomes bursátiles, los market-makers tienden a elevar el coste relativo en opciones OTM. Los inversores que poseen acciones pueden querer protegerse ante grandes pérdidas aumentando la demanda de opciones put OTM, ante lo cual están dispuestos a pagar una prima en forma de seguro.
- La volatilidad implícita y los rendimientos del índice están correlacionados negativamente (ver figura 3). Los mercados de índices de acciones tienden a subir gradualmente en periodos de volatilidad baja, en cambio suelen desplomarse bruscamente cuando la volatilidad es alta. Un crash provoca que las desviaciones típicas (volatilidad histórica) en los rendimientos de las acciones aumenten, y en consecuencia se dispare la volatilidad implícita.

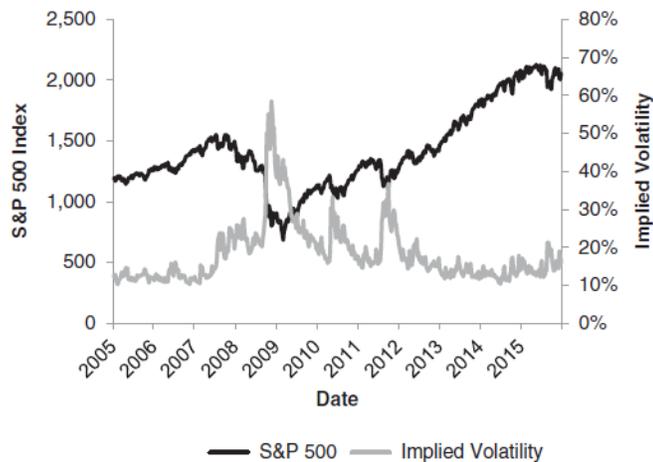


Figura 3: Correlación negativa entre S&P 500 y la volatilidad implícita (*Fuente:* [Derman and Miller(2016)])

- La volatilidad de la volatilidad implícita es mayor para los vencimientos cortos.
- La volatilidad implícita tiene un comportamiento de reversión a la media, que tiende a disminuir cuando es muy alta y a aumentar cuando es muy baja.
- La volatilidad implícita aumenta rápidamente a raíz de malas noticias, y luego declina lentamente.
- La volatilidad implícita tiende a ser mayor que la volatilidad realizada (histórica). Esto es debido a las fricciones del mercado y otros factores, incluidos los costes de cobertura, la imposibilidad de hacer coberturas perfectas, y la incertidumbre con respecto a la volatilidad futura.

Podemos pensar en la volatilidad implícita como la expectativa conjunta del mercado sobre la volatilidad futura más alguna prima asociada con el costo de estos factores.

2.3. Construcción de la sonrisa de volatilidad

Consideremos la siguiente serie de opciones de venta con fecha de vencimiento T , strikes K_i OTM, en un mercado altamente líquido y libre de arbitraje (ver [Neftci(2008)]).

$$P_{K_1}, \dots, P_{K_n}, \text{ para } K_n < \dots < K_1 < K_0 = S_t$$

De acuerdo con esto, la opción de venta en K_0 está ATM y a medida que K_i disminuye, las opciones de venta estarán más Deep-OTM (ver figura 4). Teniendo los precios de las opciones, podemos usar la inversa de la fórmula de Black-Scholes y extraer el σ_i de la opción de venta P_{K_i} . Si las suposiciones del modelo Black-Scholes se cumplen, todas las volatilidades implícitas resultarían ser las mismas,

$$\sigma_{K_0} = \sigma_{K_1} = \dots = \sigma_{K_n} = \sigma$$

ya que las opciones de venta tendrían idénticos parámetros exceptuando el strike. Así, en un entorno que cumple con los supuestos del modelo de Black-Scholes, todos los traders usarían el mismo σ_i en la fórmula de Black-Scholes para valorar cada opción de venta $P_{K_i}, i = 0, \dots, n$. Sin embargo, en la realidad si hacemos este cálculo con las cotizaciones de los precios de opciones, observaríamos que las volatilidades implícitas no son constantes, y se muestran como

$$\sigma_{K_0} < \sigma_{K_1} < \dots < \sigma_{K_n}$$

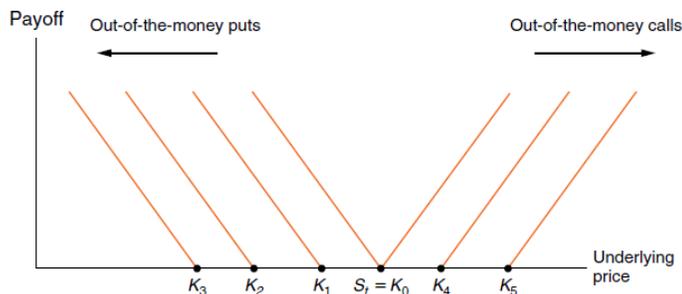


Figura 4: Payoff de Out-of-the-Money Puts y Calls (*Fuente:* [Neftci(2008)])

Cuanto más OTM esté la opción de venta, mayor será su correspondiente volatilidad implícita. Como resultado se obtiene la llamada sonrisa de volatilidad. Aplicando la misma lógica a las opciones de venta, se podrá acabar de formar el smile tal y como aparece en la figura 5. Esta sonrisa de volatilidad de forma simétrica de tipo cuadrático, es típica de opciones sobre divisas.

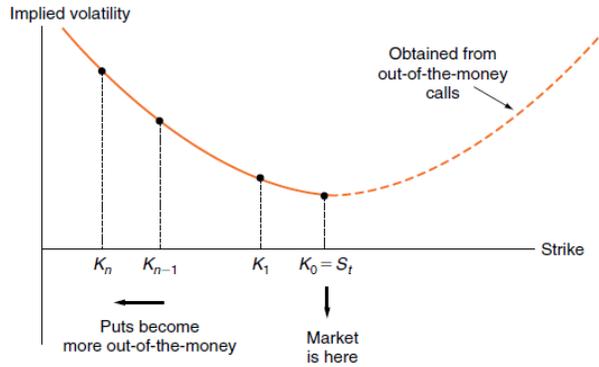


Figura 5: Smile en mercado de divisas (*Fuente:* [Neftci(2008)])

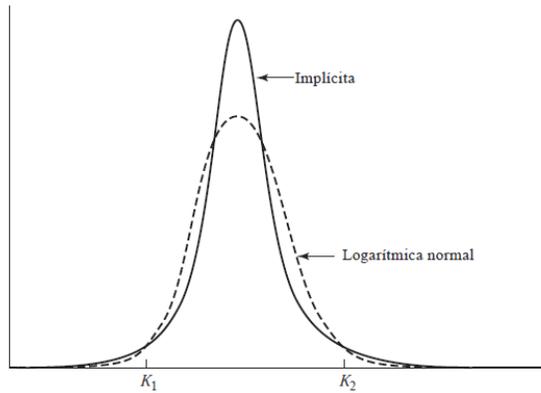


Figura 6: Distribución implícita y distribución logarítmica normal para opciones sobre divisas (*Fuente:* [Hull et al.(2009)])

La sonrisa de volatilidad de la figura 5 corresponde a la distribución implícita de probabilidades, representada con una línea continua en la figura 6. Se contrasta en la misma grafica con una distribución log-normal. Ambas comparten misma media y desviación estándar.

Observamos como la distribución implícita es leptocúrtica y presenta colas más pesadas que la distribución log-normal del modelo de Black Scholes. Se aprecian muchas más observaciones con movimientos pequeños alrededor de la media, más con movimientos grandes por presentar colas pesadas y menos con movimientos intermedios que los que supone una distribución lognormal. Esta clase de distribución es típica en el mercado de divisas.

Para explicar su relación, consideremos primero una opción call Deep-OTM con un strike elevado de K_2 ([Hull et al.(2009)]). Esta opción tendrá un beneficio si a fecha de vencimiento se ha producido un gran aumento en el precio del subyacente y supera el nivel K_2 .

La figura 6 muestra que la probabilidad de que esto ocurra es mayor para la distribución implícita que para la distribución log-normal. Esto implicará un precio de opción más elevado que el calculado con la fórmula de Black-Scholes y por lo tanto una volatilidad implícita más elevada.

En el caso de las opciones put Deep-OTM con un strike K_1 , sucederá a la inversa que en el caso de la opción call. Obtendremos un beneficio si se produce un gran descenso en el precio del subyacente inferior a K_1 . La probabilidad de que esto ocurra también será mayor en la distribución implícita que en la log-normal. Con lo cual tendremos también un precio y una volatilidad implícita más altos.

En otras palabras el modelo Black Scholes sustentado en rendimientos del subyacente log-normales infravalora la probabilidad de que opciones Deep-OTM tanto call como put sean ejecutadas con beneficio.

Según [Hull et al.(2009)] las sonrisas de volatilidad de tipo cuadrático serían consecuencia de variaciones súbitas en los tipos de cambio ante las acciones de los bancos centrales provocando las colas pesadas mostradas en la distribución implícita.

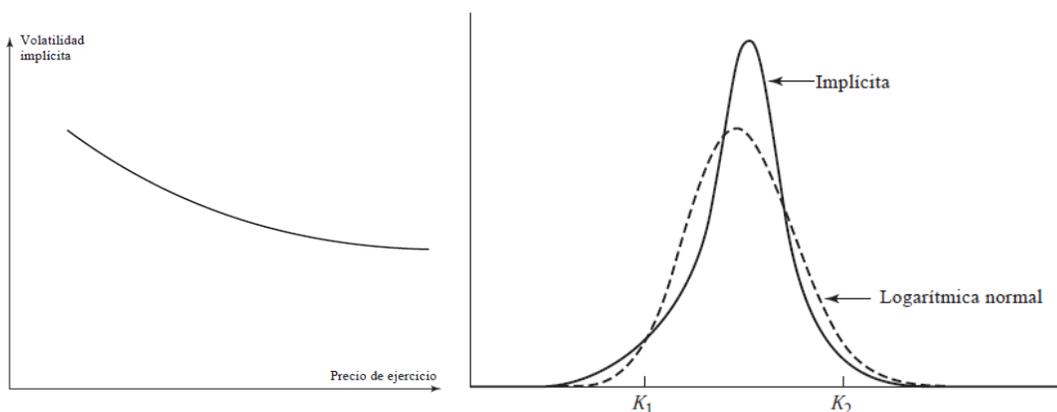


Figura 7: Sonrisa de patrón descendente y distribución implícita en el mercado de renta variable (*Fuente:* [Hull et al.(2009)])

Contrastándola con una distribución log-normal, la distribución implícita en el mercado de renta variable refleja las siguientes características. Como podemos observar en la figura 7 se aprecian muchas más observaciones con grandes bajadas así como subidas intermedias del activo subyacente en cambio existen menos probabilidades de grandes subidas así como de bajadas intermedias. Al mostrar una cola izquierda más pesada, existe una probabilidad más elevada de que el subyacente se desplome, resultando en una mayor demanda para opciones con strikes bajos.

Procederemos de la misma forma que en el caso de la sonrisa para divisas para explicar la relación entre la sonrisa de patrón descendente y su distribución implícita y llegaremos a la conclusión de que el modelo Black Scholes infravalora las opciones de compra Deep-OTM y subestima las opciones de venta Deep-OTM.

Una posible explicación a las sonrisas de patrón descendente es la ya mencionada correlación negativa entre el precio del subyacente y la volatilidad realizada (ver figura 3). Según [Rubinstein(1994)], el crash de 1987 fue un punto de inflexión en la valoración de las opciones,

la aversión al riesgo a partir de entonces se refleja en el encarecimiento de las opciones de venta en forma de prima de riesgo adicional.

3. Modelo de Black-Scholes

En 1973, Fischer Black y Myron Scholes publicaron el modelo que todavía hoy se usa de forma dogmática en la valoración de opciones financieras. Myron Scholes fue galardonado con el premio Nobel de economía en 1997 por este trabajo. El modelo está basado en las siguientes hipótesis económicas:

- El precio del subyacente se comporta según una distribución log-normal, con μ y σ constantes
- No existen costes de transacción ni impuestos.
- Todos los títulos son perfectamente divisibles
- No existen dividendos sobre el subyacente durante la vida de la opción
- No existen oportunidades de arbitraje libres de riesgo
- La negociación de valores es continua
- Los inversionistas pueden adquirir u otorgar préstamos a la misma tasa de interés libre de riesgo r que es constante

A continuación se deriva formalmente la ecuación de Black-Scholes basándonos en [Roch(2020)].

Sea $S(t)$ el precio de una acción que sigue un movimiento geométrico browniano,

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) \quad (1)$$

y sea $V(S, t)$ el valor de una opción europea sobre el subyacente $S(t)$. Puesto que el valor de la opción depende del paso del tiempo t , y del valor del subyacente $S(t)$, según el lema de Itô obtendremos

$$dV(S, t) = \left(\mu S \frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt + \sigma S \frac{\partial V}{\partial S} dW \quad (2)$$

Tanto el subyacente como la opción siguen el mismo movimiento browniano $W(t)$, por ello se puede crear una estrategia Delta-Hedging que nos permita eliminar esa aleatoriedad. Crearemos una cartera replicante $X(t)$ con una posición larga de Δ unidades del subyacente y otra posición corta de 1 unidad de la opción

$$dX(t) = \Delta dS - dV \quad (3)$$

En cada incremento infinitesimal de tiempo el valor de la cartera aumenta en Δ veces la variación de precio del activo subyacente menos una vez la variación de valor de la opción. Si sustituimos en esta última expresión dS y dV , obtenemos

$$\begin{aligned}
dX &= \Delta dS - dV \\
&= \Delta(\mu S dt + \sigma S dW) - \left(\frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt - \sigma S \frac{\partial V}{\partial S} dW \\
&= \left(\Delta \mu S - \frac{\partial V}{\partial t} - \mu S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt + \left(\Delta \sigma S - \sigma S \frac{\partial V}{\partial S} \right) dW
\end{aligned} \tag{4}$$

Eliminaremos el proceso aleatorio de esta ecuación si sustituimos Δ por $\Delta = \frac{\partial V}{\partial S}$

$$\Delta \sigma S - \sigma S \frac{\partial V}{\partial S} = 0 \Rightarrow \Delta = \frac{\partial V}{\partial S} \tag{5}$$

Una estrategia Delta-Hedging (cobertura Delta) dinámica consiste en cubrir la posición corta de la opción, con $\partial V / \partial S$ unidades del subyacente. La cartera replicante tendría la siguiente expresión:

$$dX = \left(\frac{\partial V}{\partial S} \mu S - \frac{\partial V}{\partial t} - \mu S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt \tag{6}$$

Suponiendo que no existen oportunidades de arbitraje en el mercado, el rendimiento de la cartera replicante sería igual a la tasa libre de riesgo r , por lo tanto $dX(t) = rX(t)dt$. Igualando ambas expresiones obtenemos

$$\left(\frac{\partial V}{\partial S} \mu S - \frac{\partial V}{\partial t} - \mu S \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt = rX dt \tag{7}$$

Simplificando obtenemos la ecuación de Black-Scholes.

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + r \frac{\partial V}{\partial S} = rV \tag{8}$$

La solución a esta ecuación es una función $V(t, S(t))$ que da el valor de la opción en cada momento del tiempo para cada precio del subyacente.

Una forma de interpretar esta ecuación es despejar $\frac{\partial V}{\partial t}$

$$\frac{\partial V}{\partial t} = r \left(V - \frac{\partial V}{\partial S} S \right) - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \tag{9}$$

La variación del valor de la opción en un intervalo de tiempo es igual al interés generado por la posición libre de riesgo menos el coste de la cobertura.

Para resolver la ecuación de Black-Scholes se deben también de tener en cuenta las condiciones de contorno. En el caso de una opción de compra, son:

$$C(S, T) = \max(S - K, 0), C(0, t) = 0 \text{ para todos los } t \text{ y } C(S, t) \rightarrow S \text{ si } S \rightarrow \infty.$$

En el caso de una opción de venta, son:

$P(S, T) = \max(K - S, 0)$, $P(0, t) = Ke^{-r(T-t)}$ para todos los t y $P(S, t) \rightarrow 0$ si $S \rightarrow \infty$.

De la ecuación 8 se derivan las siguientes fórmulas para calcular las opciones europeas de compra y venta:

$$C(t, S) = S_0 N(d_1) - Ke^{-r(T-t)} N(d_2) \quad (10)$$

$$P(t, S) = Ke^{-rT} N(-d_2) - S_0 N(-d_1) \quad (11)$$

$$d_1 = \frac{\ln(S/K) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} \quad (12)$$

$$d_2 = \frac{\ln(S/K) + \left(r - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t} \quad (13)$$

3.1. Delta Hedging en la práctica

Hemos explicado en el capítulo anterior la derivación formal de la cobertura Delta, en la cual por cada posición corta en una opción, se cubrirá con una cantidad $\Delta = \partial V / \partial S$ del subyacente. Esta cobertura se debería hacer de forma continua, pero en la práctica esto no es factible, debido a los altos costes de transacción [Roch(2020)].

El hecho de que no sea posible una cobertura de forma continua, implica la imposibilidad de una cobertura perfecta y que regularmente existan periodos donde la cartera tenga posiciones descubiertas y esté expuesta a un riesgo de pérdida. Cuanto mayor sea el número de periodos en los que rebalanceamos la cartera, menor será la exposición al riesgo.

En la realidad se discretiza el plazo de tiempo hasta vencimiento en varios periodos, y se reajusta el ratio de cobertura Delta en función de los cambios en el subyacente.

Esto se consigue por medio de Gamma, que es la segunda derivada de la opción respecto al subyacente, o la tasa de cambio en el Delta en relación al spot del subyacente. La cobertura Gamma se emplea para reducir la cantidad que tiene que ser rebalanceada en la posición para mantenerse Delta-neutral o para aumentar el tiempo entre cada rebalanceo, y así poder reducir los costes de transacción. Una cartera con cobertura Delta es insensible a los movimientos del subyacente siempre que esos movimientos sean bastante pequeños. La cobertura nunca será perfecta debido a la convexidad de la cartera con respecto al subyacente. La cobertura Gamma es una forma más precisa de cobertura que teóricamente elimina estos efectos de segundo orden.

Tanto la cobertura Delta como la Gamma se basan en la hipótesis de una volatilidad del subyacente constante. Como ya se ha comentado la volatilidad fluctúa en el transcurso del tiempo. Para asegurarnos que el valor de una cartera sea insensible a la volatilidad implícita podemos hacer una cobertura Vega. Vega es la sensibilidad del precio de una opción con respecto a un cambio en la volatilidad implícita. Para acometer coberturas Gamma y Vega se comprarán opciones OTM (por lo general mucho más asequibles) con vencimientos similares a la posición

por cubrir. La cobertura final de la posición por tanto está compuesta del subyacente (cobertura Delta) y dos opciones (cobertura Gamma y Vega).

Nombre	Expresión	Definición
Delta	$\Delta = \frac{\partial V}{\partial S}$	Exposición al Subyacente Cambio en precio de la opción debido al Spot
Gamma	$\Gamma = \frac{\partial^2 V}{\partial S^2}$	Convexidad del Payout Cambio en Delta debido al Spot
Theta	$\theta = \frac{\partial V}{\partial \tau}$	Time decay Cambio en precio Opcion debido al paso del tiempo
Vega	$v = \frac{\partial V}{\partial \sigma}$	Exposición Volatilidad Cambio en precio Opcion debido a la volatilidad

Cuadro 1: Letras Griegas

3.2. Paridad Put-Call

Para opciones europeas existe una relación entre los precios de una opción de compra y una opción de venta, denominada paridad put-call.

$$p + S_0 = c + Ke^{-rT}$$

Ambas opciones comparten fecha de vencimiento T y strike K , S_0 es el precio del activo subyacente a día de hoy y r es la tasa de interés libre de riesgo para el vencimiento T . En [Hull et al.(2009)] se demuestra como en base a esta expresión, aplicándose a opciones europeas la volatilidad implícita tiene que ser idéntica para contratos put y call.

Tal y como vemos en la figura 8, esta paridad no siempre se cumple y se puede obtener un beneficio por oportunidad de arbitraje. En los mercados de opciones es común explotar estas oportunidades para obtener un beneficio sin riesgo alguno [Neftci(2008)].

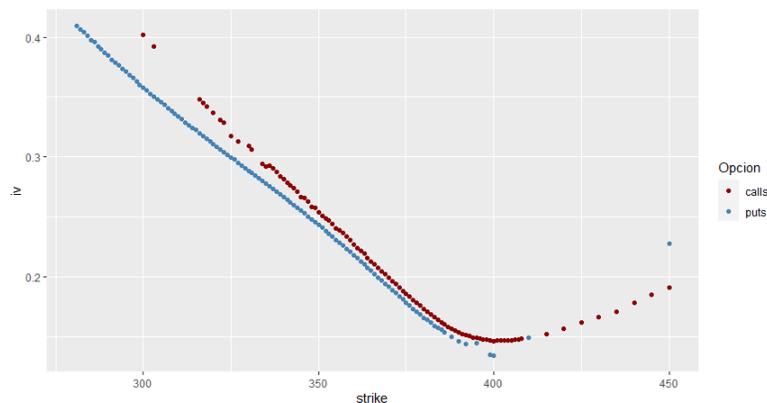


Figura 8: Sonrisas de volatilidad del SPY para vencimiento del 19 de Febrero 2021

A continuación se explica un ejemplo real aplicando esta estrategia utilizando un contrato del SPY¹ con fecha de vencimiento de 19 de febrero de 2021. A las 4:00pm del 30 de Diciembre de 2020, el activo subyacente tiene un precio de $S_t = 371,99\$$, el vencimiento de ambas opciones es en 51 días. El tipo de interés es del 0,30%. Los costes de transacción son de 0,60\$ por opción y de 0,20\$ por acción. Adicionalmente el market-maker necesita para poder seguir operando un margen de 0,75\$ por posición negociada.

La diferencia entre la prima del call $C(t)$ y del put $P(t)$, para un strike de 365\$ es de 10,24\$.

$$P(t) - C(t) = 16,44 - 6,20 = 10,24$$

El activo subyacente se compra pidiendo un préstamo a 51 días, mantendremos el contrato put ATM hasta su vencimiento, mientras que el call ATM será vendido. Esto implica un coste de financiación de:

$$371,99 \times (,003) \left(\frac{51}{360} \right) = 0,16$$

Con esta estrategia obtenemos el siguiente beneficio

coste por acción	\$
Coste Financiación	0,16
Compra Acción	0,20
Venta Call	0,60
Compra Put	0,60
Costes operacionales	0,75
Coste Total	2,31

El market-maker tendrá un coste total de 2,31\$ para acometer esta estrategia. Y la posición neta dará un beneficio libre de riesgo de 7,93\$.

$$\text{Beneficio neto} = 10,24 - 2,31 = 7,93\$$$

Se tiene que tener en cuenta que este tipo de estrategias es solamente accesible para market-makers o inversores institucionales que pagan unos costes de transacción mucho más bajos que el inversor retail.

4. Métodos numéricos de búsqueda de raíces

Para calcular el precio de una opción de compra mediante la fórmula de Black-Scholes necesitaremos los siguiente parámetros: el precio de ejercicio K , el vencimiento T , el precio S_0 del activo subyacente lo obtendremos del mercado de valores, la tasa libre de riesgo r del mercado del tesoro. El único parámetro que nos falta es σ .

¹El SPY es el Exchange-Traded-Fund más líquido que existe y replica al S&P 500. Las opciones sobre el SPY son de tipo americano

Podríamos derivar la volatilidad realizada o histórica σ directamente del activo subyacente analizando su comportamiento en el mercado de valores durante un rango de tiempo estipulado, es decir calcular la volatilidad como una desviación estándar de los retornos del subyacente para un periodo definido. La desviación estándar es una medida estadística que mira al pasado, y debido a que la volatilidad fluctúa con el tiempo, queremos saber que valor tomará en el futuro.

Según [Natenberg and Cohen(1994)], la mayoría de los participantes del mercado creen que los precios de mercado reflejan toda la información disponible que afecta al valor de un contrato, por ello el mejor predictor de la volatilidad futura es la volatilidad implícita. Lo que haremos por tanto será calcular la volatilidad según las expectativas que tiene el mercado respecto a la variabilidad futura del subyacente, es decir que expectativas tiene el mercado respecto a la distribución implícita de probabilidad futura del subyacente.

Es decir calcularemos el valor de la volatilidad implícita igualando el precio de cotización de mercado de la opción con la fórmula de Black Scholes y resolviendo según σ .

Debido a la no linealidad de la fórmula de Black Scholes, no es posible encontrar una solución analítica para la volatilidad implícita. Para ello debemos emplear métodos numéricos de búsqueda de raíces para resolver la ecuación $f(\sigma) = 0$.

$$f(\sigma) = BS(S, T, r, K, \sigma(K, T)) - V_{\text{market}} = 0 \quad (14)$$

V_{market} denota el precio de mercado actual de una opción con fecha de vencimiento T y strike K , y $BS()$ es la fórmula de Black-Scholes para valorar el precio de una opción.

Con esta derivación de la volatilidad implícita, no estamos asumiendo que el modelo Black-Scholes sea correcto. Solamente estamos usando el modelo como una herramienta para transformar la medida de valoración de una opción, es decir en vez de unidades monetarias la mediremos por su volatilidad implícita. ¿Por qué es esto útil? Porque las opciones con diferentes strikes y fechas de vencimiento presentan precios de mercado muy dispares. Una opción Deep-ITM puede valer cientos de euros, mientras que una opción Deep-OTM sobre el mismo subyacente puede valer unos pocos euros. Comparar el valor de las opciones a partir de su precio de mercado no es intuitivo [Pan(2019)]. Es por ello que habitualmente coticen las opciones en términos de volatilidades implícitas en vez de unidades monetarias.

Los dos métodos de búsqueda de raíces expuestos en el siguiente apartado están basados en el libro Análisis Numérico de [Burden and Faires(2011)].

4.1. Bisección

El primer método de búsqueda de raíces que emplearemos es el de bisección o búsqueda binaria, que consiste en dividir a la mitad un conjunto de manera continua para buscar la solución de un problema. Este proceso implica encontrar una raíz, o cero de una función f , para resolver una ecuación $f(x) = 0$

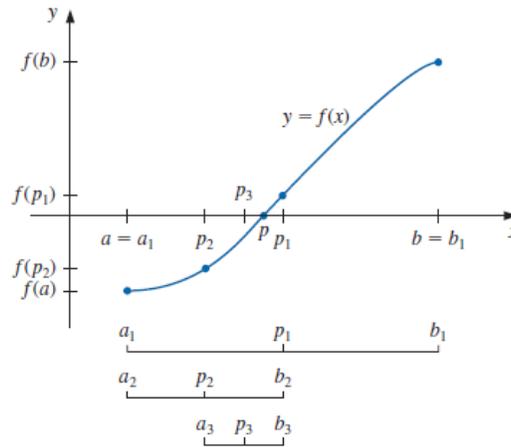


Figura 9: Método de Bisección (*Fuente:* [Burden and Faires(2011)])

Supongamos que f es una función continua definida dentro del intervalo $[a, b]$ teniendo $f(a)$ y $f(b)$ signos opuestos, tal que $f(a) \cdot f(b) < 0$. El método de bisección implica que existe un p en el intervalo (a, b) con $f(p) = 0$. El método divide reiteradamente a la mitad los subintervalos $[a, b]$ y en cada paso, localiza la mitad que contiene p .

Iniciaremos el método, asignando $a_1 = a$ y $b_1 = b$ y p_1 será el punto medio de $[a, b]$, es decir,

$$p_1 = \frac{a_1 + b_1}{2}$$

- Si $f(p_1) = 0$, entonces $p = p_1$ y habremos encontrado la solución
- Si $f(p_1) \neq 0$, entonces $f(p_1)$ tiene el mismo signo que $f(a_1)$ o bien $f(b_1)$
 - Si $f(p_1)$ y $f(a_1)$ tienen el mismo signo, $p \in (p_1, b_1)$. Asignaremos $a_2 = p_1$ y $b_2 = b_1$
 - Si $f(p_1)$ y $f(a_1)$ tienen signos opuestos, $p \in (a_1, p_1)$. Asignaremos $a_2 = a_1$ y $b_2 = p_1$

En cada subintervalo repetiremos el proceso las iteraciones que sean necesarias hasta encontrar la solución con la precisión que hayamos predeterminado. Para establecer una regla de parada, podemos seleccionar una tolerancia $\epsilon > 0$ y generar p_1, \dots, p_N hasta que se cumpla una de las siguientes condiciones:

- $|p_N - p_{N-1}| < \epsilon$
- $\frac{|p_N - p_{N-1}|}{|p_N|} < \epsilon, \quad p_N \neq 0$
- $|f(p_N)| < \epsilon$

El método de bisección es relativamente lento para converger (es decir, N puede volverse bastante grande antes de que $|p - p_N|$ sea lo suficientemente pequeño), y una buena aproximación intermedia podría descartarse inadvertidamente. Sin embargo, una búsqueda binaria siempre convergerá y siempre obtendremos una solución.

4.2. Newton-Raphson

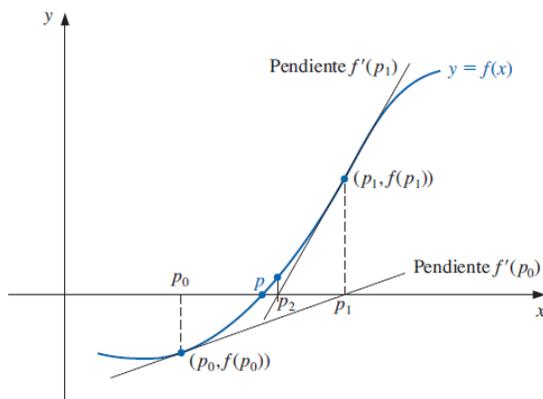


Figura 10: Método Newton-Raphson (*Fuente:* [Burden and Faires(2011)])

El segundo método de búsqueda de raíces que implementaremos es el de Newton-Raphson. Existen diferentes formas de explicar este método, el expuesto a continuación está basado en los polinomios de Taylor.

Sea $f(x)$ una función continua y diferenciable dos veces en el intervalo $[a, b]$, o lo que es lo mismo, $f \in C^2[a, b]$. Si $p_0 \in [a, b]$ es una aproximación para p , tal que $f'(p_0) \neq 0$ y $|p - p_0|$ es lo suficientemente pequeño. El primer polinomio de Taylor para $f(x)$ está expandido en torno a p_0 y evaluado en $x = p$ será:

$$f(p) = f(p_0) + (p - p_0) f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p))$$

donde $\xi(p)$ se encuentra entre p y p_0 . Dado que $f(p) = 0$, obtenemos:

$$0 = f(p_0) + (p - p_0) f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p))$$

El método de Newton-Raphson se deriva al suponer que como $|p - p_0|$ es suficientemente pequeño, el término $(p - p_0)^2$ tiende a cero, por lo tanto

$$0 \approx f(p_0) + (p - p_0) f'(p_0)$$

Despejando p resulta

$$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} \equiv p_1$$

El método de Newton consiste en tomar una aproximación inicial p_0 y generar recursivamente la sucesión $\{p_n\}_{n=0}^{\infty}$, hasta aproximarnos lo suficiente a la raíz p mediante

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{para } n \geq 1 \quad (15)$$

Estableceremos un criterio de parada análogo al método de bisección, es decir seleccionaremos una tolerancia $\epsilon > 0$ y generaremos p_1, \dots, p_N hasta que se cumpla una de las siguientes condiciones:

- $|p_N - p_{N-1}| < \epsilon$
- $\frac{|p_N - p_{N-1}|}{|p_N|} < \epsilon, \quad p_N \neq 0$
- $|f(p_N)| < \epsilon$

La figura 10 ilustra cómo se obtienen las aproximaciones usando tangentes sucesivas. Al empezar con la aproximación inicial p_0 , la aproximación p_1 es la intersección con el eje x de la recta tangente a la función de f en $(p_0, f(p_0))$. La aproximación p_2 es la intersección con el eje X de la recta tangente a la función f en $(p_1, f(p_1))$ y así sucesivamente.

Para calcular la volatilidad implícita del modelo de Black-Scholes, utilizaremos la ecuación 15 análogamente con los siguientes parámetros [Merk(2011)],

$$\sigma_{i+1} = \sigma_i - \frac{V(\sigma_i) - V_{market}}{\partial V / \partial \sigma_i}$$

siendo

- $\sigma_i :=$ para $i = 0$, valor inicial para la volatilidad implícita
- $V_{market} :=$ precio de mercado de la opción
- $V(\sigma_i) :=$ precio de la opción en σ_i
- $\partial V / \partial \sigma_i = S \cdot \phi(d_1) \sqrt{T} :=$ Vega de la opción en σ_i
- $d_1 = \frac{\log(\frac{S}{K}) + (r + \frac{\sigma^2}{2})T}{\sigma \sqrt{T}}$

Se puede dar el caso que el algoritmo dé error para Vegas cercanos a cero, como es el caso en opciones Deep-ITM o Deep-OTM (con vencimientos cortos). [Manaster and Koehler(1982)] sugieren como valor inicial para σ_0 :

$$\sigma_0 = \sqrt{\frac{2}{T} |\log(S/K) + rT|}$$

5. Métodos de interpolación

Como hemos visto en la web de Yahoo Finance, se negocian diariamente contratos de derivados con diferentes strikes y vencimientos con sus respectivas volatilidades implícitas. Estas cotizaciones son un conjunto de datos discretos. Cómo calculamos un derivado que no figura en la base de datos? Pongamos como ejemplo un hedge fund que tiene una cartera de opciones negociadas en mercados extrabursátiles (Over-The-Counter) con multitud de strikes y vencimientos. Muchos de estos contratos no cotizarán en mercados organizados con lo cual no podremos saber como fluctúan sus precios. Cómo calculamos una opción para una pareja de strike y vencimiento que no figura en el CBOE?

La principal utilidad de la superficie de volatilidad es extraer de ella volatilidades implícitas mediante interpolación para parejas de strikes y vencimientos no cotizados. En la siguiente figura 11 se muestra un extracto de los datos de la superficie de volatilidad que hemos creado. En el eje horizontal x se muestran los ratios moneyness K/S_0 y en el eje vertical y los días hasta vencimiento. Para cada pareja (x, y) encontramos su respectiva volatilidad implícita $z = f(x, y)$ en el cuerpo de la matriz. Como observamos existen muchos huecos vacíos debido a strikes que no están cotizados.

	1.00403755206133	1.0066797561457	1.00932196023007	1.01064306227226	1.01196416431444	1.01460636839882	1.01724857248319	
25	0.1987839	0.1952040	0.1920908	0.1901904		0.1889154	0.1843407	0.1817486
36	0.1965683	NA		NA			NA	0.1811553
39	0.1999771	0.1962207	0.1931318	NA		0.1900888	0.1870783	0.1849243
67	0.2035328	0.2011975	0.1988257	NA		0.1968772	0.1947272	0.1926818
79	0.2001642	0.1980205	0.1962833	NA		0.1940930	0.1914421	0.1891837
95	0.2021616	0.1995615	0.1981267	NA		0.1957749	NA	0.1927310
130	0.2051480	0.2036974	0.2022633	NA		0.2007332	0.1988826	0.1967082
158	0.2050805	0.2034544	0.2013486	NA		NA	NA	0.1961634
170	0.2023122	0.2013250	0.1998770	NA		NA	NA	0.1955724

Figura 11: Matriz de volatilidades implícitas para Calls del SPX

El scatter plot de esta matriz de observaciones se visualiza gráficamente en la figura 13. Cada fila de la matriz representa una sonrisa de volatilidad para su respectivo vencimiento.

Cuando se tiene que valorar una nueva opción de la cual no existen cotizaciones, se extrae la correspondiente volatilidad implícita de la matriz mediante interpolación. En este trabajo se han implementado dos metodologías.

5.1. Interpolación univariada de splines cúbicos

Cuando queremos evaluar volatilidades implícitas de strikes no cotizados de vencimientos negociados en el CBOE utilizaremos una interpolación univariada.

Utilizaremos la función *splinefun* de la librería *stats* de *R* para implementar una interpolación de splines cúbicos univariada para interpolar las volatilidades implícitas calculadas mediante la fórmula de Black-Scholes y los métodos de búsqueda de raíces. En el anexo se explica al detalle el procedimiento de esta interpolación.

En la figura 12 se muestra la matriz de la figura 11 interpolada.

	1.00403755206133	1.0066797561457	1.00932196023007	1.01064306227226	1.01196416431444	1.01460636839882	1.01724857248319
25	0.2058623	0.2031641	0.2004831	0.1991504	0.1978239	0.1951914	0.1925903
36	0.1983029	0.1956790	0.1931168	0.1918594	0.1906182	0.1881849	0.1858187
39	0.1968644	0.1942614	0.1917282	0.1904881	0.1892659	0.1868759	0.1845592
67	0.1989357	0.1965552	0.1942164	0.1930635	0.1919220	0.1896744	0.1874762
79	0.2028850	0.2006159	0.1983598	0.1972378	0.1961206	0.1939025	0.1917096
95	0.2055249	0.2034174	0.2013076	0.2002532	0.1991999	0.1970989	0.1950088
130	0.2044314	0.2026480	0.2008793	0.2000010	0.1991271	0.1973934	0.1956802
158	0.2034765	0.2018461	0.2002289	0.1994256	0.1986262	0.1970397	0.1954710
170	0.2033231	0.2017408	0.2001663	0.1993825	0.1986013	0.1970477	0.1955073

Figura 12: Matriz de volatilidades implícitas para calls del SPX interpolada

En la figura 14 se muestra el scatterplot de la matriz 12 interpolada. Las sonrisas de volatilidad de cada vencimiento se muestran en este caso interpoladas por splines cúbicos.

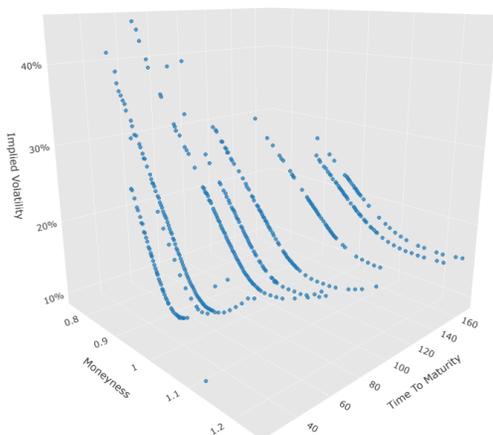


Figura 13: Scatter plot sin interpolar

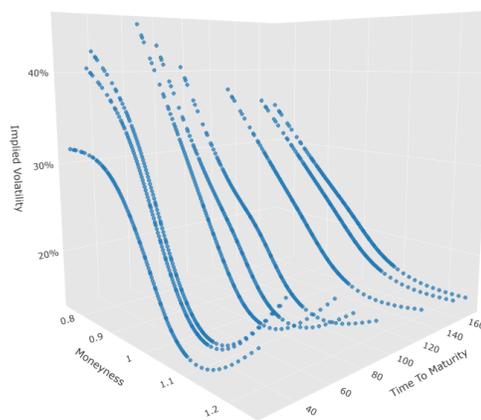


Figura 14: Scatter plot interpolado

Como se ilustra en la figura 15, desde el scatter plot interpolado por splines cúbicos univariado podemos automáticamente visualizar las volatilidades implícitas interpoladas para cada vencimiento. Sencillamente fijamos el cursor del mouse en la coordenada buscada en el plot creado y obtenemos la volatilidad implícita por ejemplo para $f(1,0013; 170) = 20\%$, es decir, el call con fecha de vencimiento de 7 de Septiembre de 2021 y strike 3790 obtiene una volatilidad implícita de 20%.

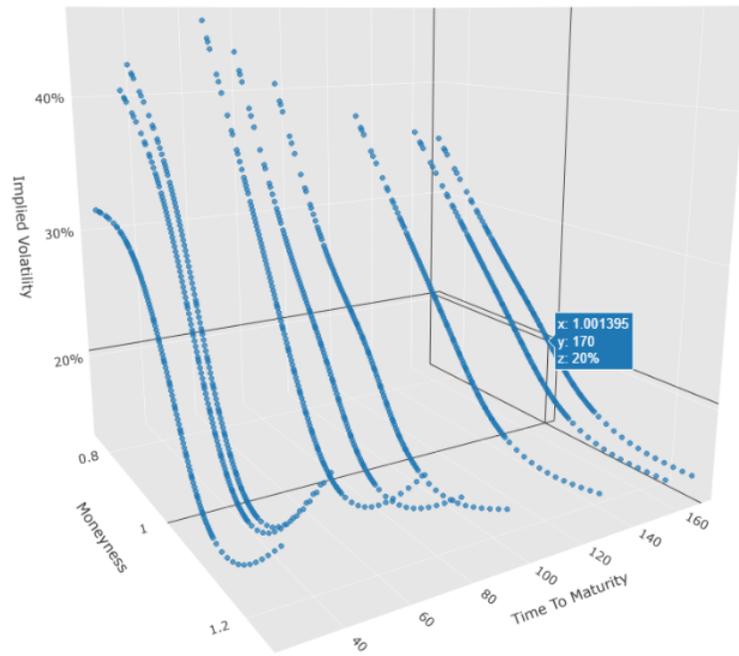


Figura 15: Buscador de volatilidad implícita automático

📷 🔄 🏠 📏 📐 📊 📋

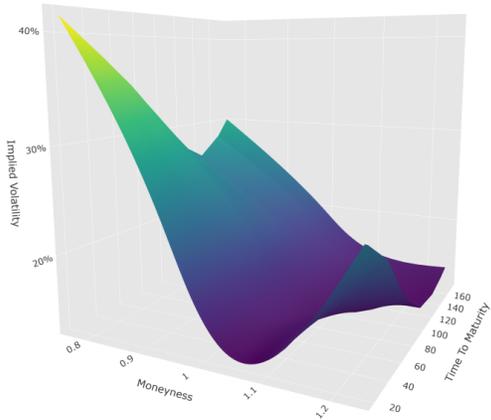


Figura 16: Superficie de volatilidad para vencimientos 7-180 días

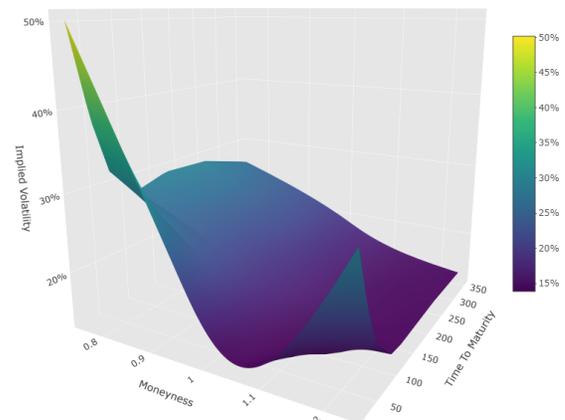


Figura 17: Superficie de volatilidad para vencimientos 7-360 días

Mediante la librería *plotly* de *R* se grafica la superficie de volatilidad (ver figura 16) a partir de la matriz interpolada de la figura 14. Se muestra para contrastar una segunda superficie de volatilidad del índice SPX mostrando vencimientos hasta 360 días (ver figura 17), y vemos claramente como la superficie reduce su pendiente y se va aplanando a medida que se alargan los vencimientos.

5.2. Interpolación bivariada de splines cúbicos

Si queremos interpolar volatilidades implícitas de parejas de vencimientos y strikes no cotizadas en los mercados organizados como por ejemplo $f(0,93; 100) = z\%$ en la tabla 2. Es decir un vencimiento y un strike para los cuales no tenemos datos de referencia ni en la dimensión x ni en la y , sobre los cuales implementar una interpolación univariada, necesitaremos usar una interpolación en dos dimensiones para poder estimar dicho punto [Hull et al.(2009)], para el cual tomaremos una región representativa alrededor del punto que queremos estimar. En este trabajo hemos utilizado una interpolación de splines cúbicos bivariada usando la función *interp* de la librería *Akima* de *R*. En el anexo se explica al detalle el procedimiento de esta interpolación.

	K/S_0				
	0,90	0,95	1,00	1,05	1,10
70 días	14,2	13,0	12,0	13,1	14,5
82 días	14,0	13,0	12,0	13,1	14,2
98 días	14,1	13,3	12,5	13,4	14,3
133 días	14,7	14,0	13,5	14,0	14,8
161 días	15,0	14,4	14,0	14,5	15,1
173 días	14,8	14,6	14,4	14,7	15,0

Cuadro 2: Muestra de superficie de volatilidad de mercado de divisas

La metodología de interpolación que proponemos es la siguiente.

- **Primero:** Necesitamos encontrar la posición aproximada del punto a estimar; para el ejemplo del valor buscado para $f(0,93; 100) = z\%$, esta coordinada se localizaría entre la primera y la segunda columna en el eje K/S_0 que corresponden a los valores 0,90 y 0,95 respectivamente, y en el eje de los días de vencimiento estaría entre la tercera y la cuarta fila correspondientes a los valores 98 y 133. La localización de esta coordinada será realizada mediante una adaptación del algoritmo de búsqueda binaria. Esta adaptación consiste en recorrer el vector mediante una búsqueda binaria normal hasta llegar al par de puntos más próximo al punto faltante, una vez allí el algoritmo escoge cual de los dos candidatos es el punto más próximo, y retorna la posición de éste.

En la figura 18 se ilustra gráficamente este primer paso y el siguiente.

- **Segundo:** Construimos un grid de radio 10 alrededor de la posición del punto más cercano obtenido en el paso anterior. De esta forma tenemos una nube de puntos alrededor del punto a estimar que nos servirá de base para usar la interpolación.
- **Tercero:** Usando el grid obtenido en el paso anterior se aplica el método de splines cúbicos bivariado para estimar el valor del punto buscado.

Es necesario observar que esta estrategia se base en encontrar la subsuperficie mas representativa en relación al dato faltante para que de este modo la estimación sea la más exacta posible.

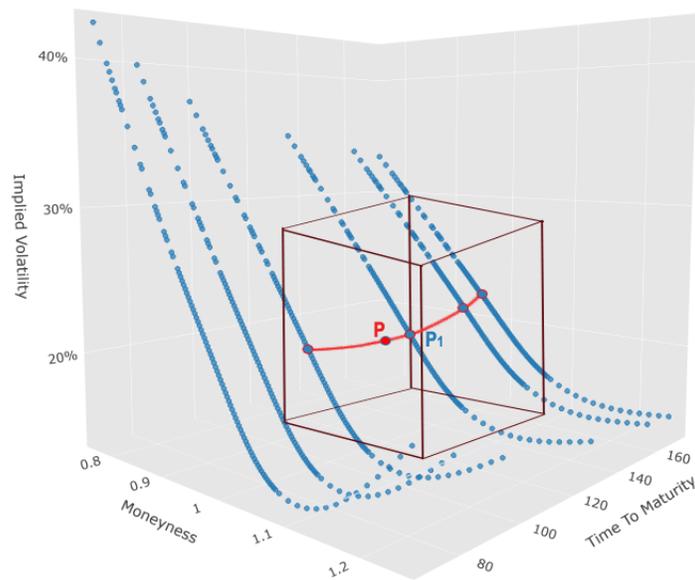


Figura 18: Ilustración del paso Primero y Segundo de la interpolación bivariada

6. Validación de los métodos de búsqueda de raíces

Con el fin de validar los métodos de bisección y Newton-Raphson, evaluaremos su eficacia en comparación con las volatilidades implícitas proporcionadas por Yahoo Finance. Una vez estimados los valores de las volatilidades por ambos métodos de búsqueda de raíces comparamos los resultados con los datos originales mediante el error cuadrático medio (ECM) y mediante el coeficiente de determinación R^2 . Como se puede observar en la tabla 3, el error de estimación en términos de ECM de ambos métodos es despreciable, apareciendo apenas en el sexto decimal para ambos casos, del mismo modo en términos de R^2 ambos métodos presentan un ajuste excelente alcanzando un R^2 muy próximo al ideal ($R^2 = 1$). Ambos estimadores se explican detalladamente en el anexo de este trabajo. De estos resultados podemos concluir que ambos métodos son altamente efectivos y que Yahoo Finance emplea también el modelo de Black-Scholes generalizado ya que nuestros resultados son prácticamente idénticos.

Método	ECM	R^2
Bisección	1.048324e-06	0.9996493
Newton	1.048398e-06	0.9996493
Diferencia	7.466638e-11	2.498023e-08

Cuadro 3: Error cuadrático medio (ECM) y coeficiente de determinación (R^2) para el método de bisección y el método de Newton-Raphson

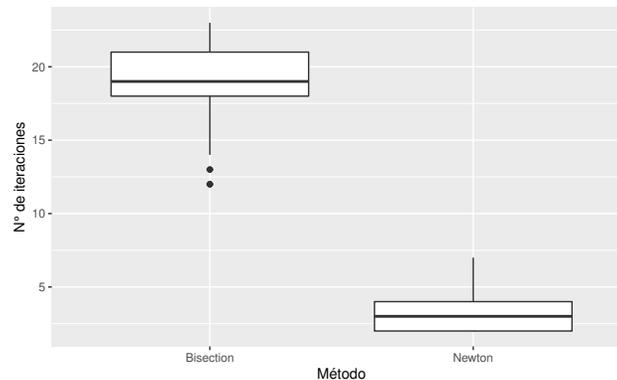


Figura 19: Eficiencia de los métodos de búsqueda de raíces

Para valorar la eficiencia de ambos métodos de búsqueda de raíces, estimaremos su tiempo de ejecución comparando el número de iteraciones que precisa cada método para converger. La figura 19 muestra el boxplot que compara ambos métodos, podemos observar claramente que el método de Newton-Raphson tiene un coste computacional mucho menor, necesitando muchas menos iteraciones que el método de bisección para calcular todos los puntos. Los valores del método de Newton varían entre $3,14 \pm 1,18$ iteraciones, mientras los valores del método de bisección varían entre $19,14 \pm 2,40$ iteraciones.

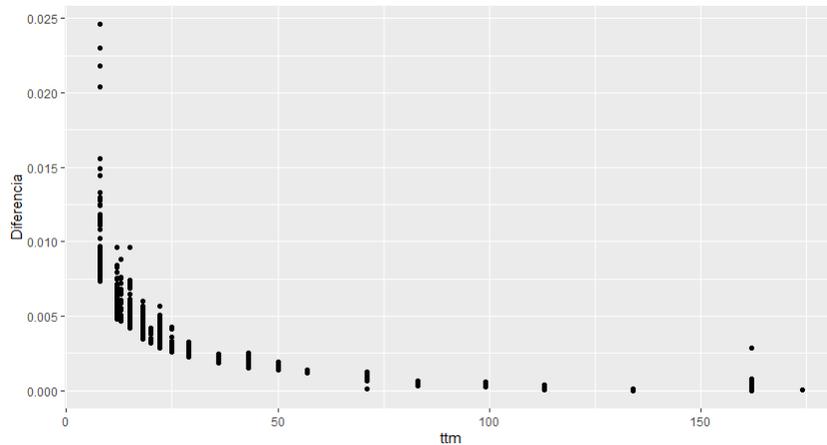


Figura 20: Comparativa entre volatilidades implícitas calculadas con método de bisección y datos originales de Yahoo Finance

En la figura 20 se muestra como la diferencia en valores absolutos entre las volatilidades implícitas obtenidas de Yahoo Finance y las calculadas con el modelo de Black Scholes y el método de Bisección y Newton-Raphson son mínimas y disminuyen a medida que aumenta el plazo de vencimiento. Esto es debido al hecho de calcular las volatilidades implícitas en base al precio de cierre, mientras que los precios de las opciones empleadas se basan en diferentes fluctuaciones del spot a lo largo de todo el día de cotización. Este efecto es más notorio en plazos de vencimiento cortos.

Las ventajas del método de bisección son que convergerá siempre, se puede implementar sin requerimiento de la derivada *Vega* y se puede utilizar también para valorar opciones americanas. Por otro lado, el método de bisección es menos eficiente que el de Newton-Raphson, y tiene una exigencia computacional elevada. Por tanto si el número de opciones que tenemos en cartera es muy grande esto puede suponer un problema. De ahí que el método de Newton-Raphson sea el que suelen utilizar los traders profesionales por su alta eficiencia.

7. Validación de la interpolación de splines cúbicos univariada

Con el objetivo de analizar el rendimiento del método de interpolación con splines cúbicos univariado se realiza una separación de los datos obtenidos por Yahoo Finance, en datos de entrenamiento (datos que serán usados para realizar la interpolación), y datos de prueba (datos que servirán para evaluar el rendimiento del método) [James and Tibshirani(2013)]. El procedimiento es el siguiente [Analytics(2021)]:

- **Primero:** Se calculan las volatilidades implícitas utilizando el modelo de Black Scholes y el método de bisección con los datos obtenidos de Yahoo Finance
- **Segundo:** Se realiza una separación de los datos originales en datos de entrenamiento y

datos de prueba de la siguiente forma: (90 % de los datos para entrenamiento y 10 % de los datos para prueba.)

- **Tercero:** Se realiza una estimación de los puntos faltantes (datos de prueba) usando la interpolación de splines cúbicos. Se utilizan solamente los datos de entrenamiento para generar la interpolación.
- **Cuarto:** Se realiza un análisis en base al ECM y al R^2 en los datos de prueba, comparando los valores obtenidos por el método de interpolación en comparación a los datos obtenidos directamente de Yahoo Finance (datos originales). Los estimadores ECM y R^2 también se calculan usando directamente los métodos de búsqueda de raíces, de esta forma podremos comparar estas métricas y ver si los valores interpolados se acercan lo suficiente a los valores obtenidos directamente por los métodos de búsqueda de raíces.

El cuadro 4 muestra una comparativa entre el método de interpolación por splines cúbicos y el cálculo directo con los métodos de búsqueda de raíces, en comparación de los datos reales obtenidos por Yahoo Finance. Como es de esperar el método de interpolación por splines cúbicos tiene una tasa de error mayor en términos de ECM, sin embargo la diferencia es muy pequeña, apareciendo en el quinto decimal, lo cual nos indica que el método de interpolación es muy preciso en términos de ECM. En términos de R^2 la diferencia es más considerable presentando una diferencia de 0,0201 es decir aproximadamente un 2 %, sin embargo un poder de estimación de $R^2 = 0,98$ sigue siendo un ajuste excelente.

Metodo	ECM	R^2
Bisección/Newton	1.554653e-06	0.9996211
Spline cúbico	8.408762e-05	0.9795051
Diferencia	8.253296e-05	0.02011594

Cuadro 4: Valores para el error cuadrático medio (ECM), y el coeficiente de determinación (R^2) para la interpolación de splines cúbicos univariada

En términos generales en base a los resultados obtenidos podemos concluir que el método de interpolación de splines cúbicos univariada estima muy bien los valores de las volatilidades implícitas, con resultados muy próximos a los calculados directamente por los métodos de búsqueda de raíces.

8. Validación de la interpolación de splines cúbicos bivariada

Para validar la estimación de las volatilidades implícitas mediante interpolación de splines cúbicos bivariada realizaremos una adaptación del método de validación cruzada dejando uno fuera (ver anexo) [James and Tibshirani(2013)]. A continuación se explica el procedimiento:

- **Primero:** Construimos una matriz $(m \times n)$ con las volatilidades implícitas por validar en base a las observaciones de Yahoo Finance (ver figura 21 (a)). El conjunto de puntos es reducido a aquellas combinaciones (*vencimiento, strike*) que estén dentro de la superficie de interpolación, es decir obiamos de la matriz la primera y última fila de vencimientos, y la primera y última columna de strikes (ver figura 21 (b)).
- **Segundo:** Si queremos validar un punto específico (*vencimiento, strike*) de la matriz obtenida en el primer paso, seleccionaremos primero todos los puntos de la fila y columna respectiva que interseccionan en el punto seleccionado (ver punto naranja en la figura 21 (c)). A continuación construimos un grid de interpolación, excluyendo toda la fila y toda la columna que interseccionan en el punto específico a validar. (ver figura 21 (d)). La dimensión del grid se habrá reducido en una columna y una fila $(m - 1) \times (n - 1)$.
- **Tercero:** Interpolamos los valores faltantes del grid obtenido en el paso anterior mediante splines cúbicos univariados (ver puntos azules en la figura 21 (e)).
- **Cuarto:** En base al grid interpolado obtenido en el paso anterior estimamos el valor de la volatilidad implícita correspondiente al punto buscado mediante la interpolación de splines cúbicos bivariada.
- **Quinto:** Repetimos el segundo, tercer y cuarto paso para todas las observaciones resultantes del primer paso (flechas rojas en la figura 21), y obtenemos todo el conjunto de puntos estimados por validación cruzada usando la interpolación de splines cúbicos bivariada (ver figura 21 (f)).
- **Sexto:** Con el conjunto de puntos estimados en el paso anterior calculamos los valores del error cuadrático medio (ECM) y coeficiente de determinación R^2 , comparando los valores estimados con las volatilidades implícitas obtenidas directamente de Yahoo Finance.

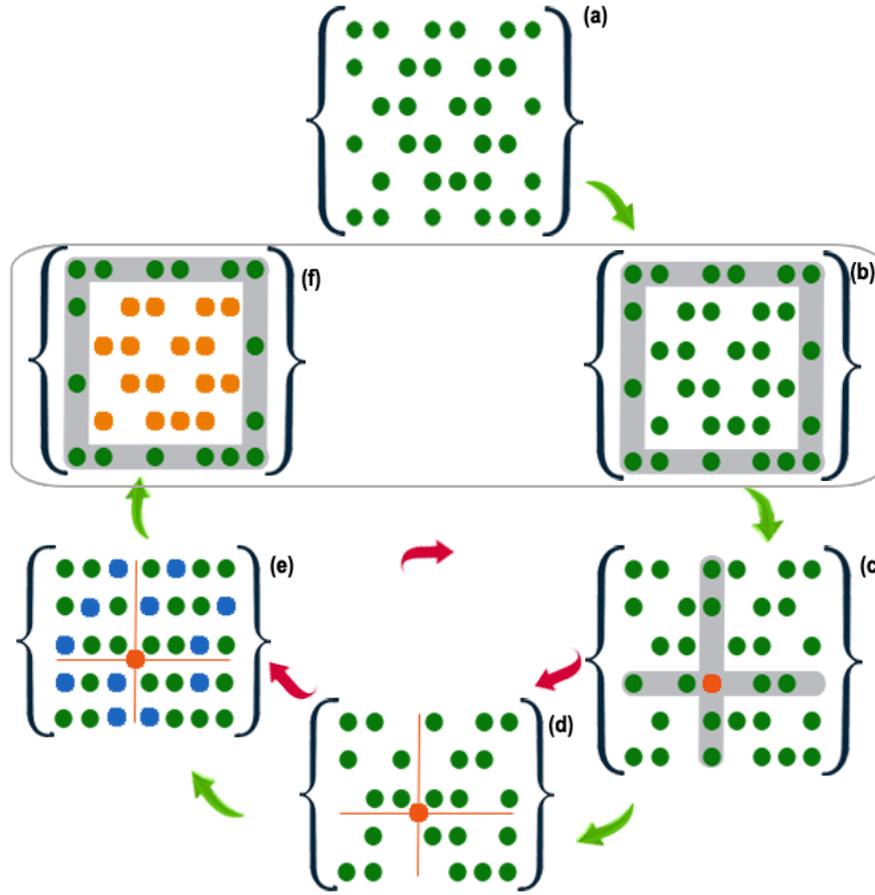


Figura 21: Proceso de validación de la interpolación de splines cúbicos bivariada: (a) Grid de volatilidades implícitas obtenidas de Yahoo Finance, (b) Filtro de filas y columnas exteriores, sólo los valores internos del área sombreada serán validados, (c) Selección del punto naranja por validar. Se seleccionan también los puntos que están en la misma columna y misma fila, (d) Reducción de la matriz de interpolación por eliminación de la fila y columna que interseccionan en el punto por validar, (e) Interpolación de los puntos faltantes (azules) de la matriz reducida mediante splines cúbicos univariados. La volatilidad implícita del punto naranja, será interpolada con esta nueva matriz de puntos mediante splines cúbicos bivariados, (f) Resultado de interpolar todos los puntos de test por splines cúbicos bivariados (puntos naranjas)

Método	ECM	R^2
Bisección/Newton	5.853628e-07	0.9996413
Spline bicúbico	4.688024e-05	0.9712753
Diferencia	4.629487e-05	0.02836607

Cuadro 5: Valores de error cuadrático medio (ECM), y coeficiente de determinación (R^2) para la interpolación de splines cúbicos bivariada

La tabla 5 muestra los resultados del error cuadrático medio (ECM) y coeficiente de determinación R^2 . Los resultados son ligeramente peores que en la validación de splines cúbicos univariados. La diferencia en términos de ECM en relación a los métodos de búsqueda de raíces, es despreciable, apareciendo el error en el quinto decimal. En términos de R^2 obtenemos un valor de 0,97, un ajuste todavía muy próximo al ideal ($R^2 = 1$). Podemos concluir que la estimación de volatilidades implícitas para parejas de strikes y vencimientos no cotizadas en mercados organizados, mediante la interpolación bivariada de splines cúbicos es un método muy preciso ya que nos permite obtener valores muy próximos a los reales.

9. Conclusión

Si la volatilidad implícita fuera constante y si el modelo de Black-Scholes fuera correcto, la sonrisa de volatilidad sería plana, en cambio cada mercado tiene su propia forma de sonrisa resultante de una psicología propia con un comportamiento financiero diferente, en gran medida causada por una aversión al riesgo generalizada en todos los mercados. En los mercados de divisas y renta fija se temen las intervenciones de los bancos centrales, los de renta variable son crash-o-fóbicos [Rubinstein(1994)] y en los de materias primas temen los efectos del cambio climático. Por tanto, la sonrisa de volatilidad nos está brindando información sobre las expectativas que tiene el mercado sobre la volatilidad futura en un momento determinado del tiempo. Las diferentes formas que toma la sonrisa de volatilidad son una manera en que los mercados nos transmiten sus discrepancias con el modelo de Black-Scholes y al mismo tiempo son un nexo de unión entre las irreales hipótesis económicas del modelo y la realidad de los mercados financieros.

En la primera parte del trabajo se ha demostrado que los dos métodos numéricos de búsqueda de raíces, tanto el método de bisección como el de Newton-Raphson son altamente eficaces y perfectamente compatibles para el cálculo de la volatilidad implícita mediante la inversa de Black-Scholes. Es más, los resultados obtenidos son prácticamente idénticos a los valores originales de Yahoo Finance, por lo cual podemos deducir que también emplean el modelo de Black-Scholes. El método de bisección garantiza la convergencia pero por otro lado es muy exigente computacionalmente, mientras que el método de Newton-Raphson es mucho más eficiente por lo que será más recomendable para aquellos profesionales que gestionen grandes carteras de opciones.

En la segunda parte del trabajo se ha querido enseñar el procedimiento para construir la superficie de volatilidad y así poder extraer de ella las volatilidades implícitas para valorar opciones no cotizadas en mercados organizados mediante dos métodos diferentes de interpolación.

La interpolación de splines cúbicos univariada nos ha servido para poder estimar volatilidades implícitas de vencimientos cuyos strikes no cotizan en las bolsas organizadas. Mediante esta interpolación también hemos graficado la volatilidad implícita en sus tres modalidades: sonrisa de volatilidad, scatter plot y superficie de volatilidad y hemos podido verificar que todas las características comentadas en el capítulo 2 coinciden con los resultados obtenidos. La interpolación bivariada de splines cúbicos ha sido necesaria para poder extraer además volatilidades implícitas de la superficie de volatilidad cuyos vencimientos y strikes no cotizan en

bolsas organizadas.

Implementar los dos métodos de interpolaciones es relativamente sencillo dada la cantidad de opciones disponibles mediante las librerías de R , ahora bien lo interesante ha sido comparar los resultados obtenidos mediante interpolación con los datos reales y testear su validez.

Para ello hemos utilizado métodos de validación que se emplean habitualmente en Machine Learning, los cuales nos permiten estimar la capacidad predictiva del modelo a validar.

Los resultados obtenidos mediante las métricas ECM y R^2 , tanto de la validación simple de la interpolación univariada como de la validación cruzada adaptada empleada en la interpolación bivariada han dado resultados excelentes. Ésto nos permite juzgar que ambas interpolaciones de splines cúbicos, realizan generalizaciones eficaces sobre las volatilidades implícitas interpoladas de la superficie de volatilidad y por lo tanto evidencia el motivo por el cual ambas interpolaciones son tan populares entre los quants profesionales.

10. Anexos

10.1. Interpolación univariada de splines cúbicos

Mostramos a continuación la aproximación por tramos de splines cúbicos univariada basándonos en [Bronstein and Semendjajew(2005)].

Si tenemos un conjunto de nodos por interpolar $(x_j, y_j)(j = 0, 1, 2, \dots, n)$, el interpolante de splines cúbicos $s(x)$ para y_j es una función que tiene las siguientes propiedades:

- $s(x_j) = y_j \quad (j = 0, 1, 2, \dots, n)$
- $s(x)$ es para $x \in [x_i, x_{i+1}] \quad (0 \leq i \leq n - 1)$ un polinomio de grado ≤ 3
- $s(x) \in C^2([x_0, x_n])$
- $s''(x_0) = s''(x_n) = 0$.

La función s se compone de polinomios de tercer grado o intervalos, que cumplen las condiciones de interpolación en los nodos. Es continua y dos veces diferenciable en los puntos de interpolación interiores y tiene segundas derivadas con valor cero en los extremos. Para el cálculo numérico del interpolante $s(x)$, representa

$$h_i := x_{i+1} - x_i > 0 \quad (i = 0, 1, 2, \dots, n - 1)$$

la longitud de los subintervalos $[x_i, x_{i+1}]$, para los cuales la expresión de $s(x)$ es

$$s_i(x) = a_i (x - x_i)^3 + b_i (x - x_i)^2 + c_i (x - x_i) + d_i, \quad x \in [x_i, x_{i+1}]$$

Adicionalmente al conjunto de y_i , se necesitan las segundas derivadas y_i'' para obtener los intervalos $s_i(x)$. Los cuatro coeficientes a_i, b_i, c_i, d_i de $s_i(x)$ se calculan con las siguientes expresiones

$$\begin{aligned} a_i &= (y_{i+1}'' - y_i'') / (6h_i), & b_i &= y_i'' / 2 \\ c_i &= (y_{i+1} - y_i) / h_i - h_i (y_{i+1}'' + 2y_i'') / 6, & d_i &= y_i \end{aligned}$$

Con este planteamiento se garantizan las condiciones de interpolación y la continuidad de las segundas derivadas en los puntos interiores. La condición de continuidad de la primera derivada en los $n - 1$ puntos interiores x_i proporciona las $n - 1$ ecuaciones lineales

$$h_{i-1}y_{i-1}'' + 2(h_{i-1} + h_i)y_i'' + h_i y_{i+1}'' - \frac{6}{h_i}(y_{i+1} - y_i) + \frac{6}{h_{i-1}}(y_i - y_{i-1}) = 0$$

para $i = 1, 2, \dots, n - 1$. Junto con las condiciones de contorno para spline cúbicos naturales $y_0'' = y_n'' = 0$ obtendremos un sistema de ecuaciones lineal para las $n - 1$ incógnitas y_1'', \dots, y_{n-1}'' . Su matriz de coeficientes es simétrica, tridiagonal y de diagonal estrictamente dominante, con lo cual el sistema de ecuaciones tiene una solución única.

10.2. Interpolación bivariada de splines cúbicos

La teoría de este apartado está basada en [Bronstein and Semendjajew(2005)].

Dado un grid rectangular en el plano x, y

$$a = x_0 < x_1 < \dots < x_n = b, \quad c = y_0 < y_1 < \dots < y_m = d$$

con un conjunto de puntos $(x_i, y_j), i = 0, 1, \dots, n, \quad j = 0, 1, \dots, m$ y dado un rectángulo $R = \{(x, y) \mid a \leq x \leq b, c \leq y \leq d\}$ con la siguiente función definida en

$$u : R \rightarrow \mathbb{R}, \quad (x, y) \mapsto u(x, y)$$

y sus valores

$$u_{ij} := u(x_i, y_j), \quad i = 0, 1, \dots, n, \quad j = 0, 1, \dots, m$$

en los puntos del grid. Se buscará una función interpolante de splines de dos dimensiones

$$S : R \rightarrow \mathbb{R}, \quad (x, y) \mapsto S(x, y)$$

con la que obtendremos una superficie suavizada $\{(x, y, z) \mid (x, y) \in R, z = S(x, y)\}$

La función de splines bicúbica para S tiene las siguientes propiedades

1. S satisface las siguientes condiciones de interpolación

$$S(x_i, y_j) = u_{ij}, \quad i = 0, 1, \dots, n, \quad j = 0, 1, \dots, m$$

2. S es diferenciable en R

3. $\frac{\partial^2 S}{\partial x \partial y}$ es continua en R

4. En cada subrectángulo R_{ij} con

$$R_{ij} := \{(x, y) \mid x_i \leq x \leq x_{i+1}, y_j \leq y \leq y_{j+1}\}, \quad i = 0, 1, \dots, n-1, \quad j = 0, 1, \dots, m-1$$

S será idéntica al polinomio bicúbico

$$S(x, y) \equiv S_{ij}(x, y) = \sum_{k=0}^3 \sum_{s=0}^3 a_{ijk_s} (x - x_i)^k (y - y_j)^s \tag{16}$$

$$\text{para } (x, y) \in R_{ij}, \quad i = 0, 1, \dots, n-1, \quad j = 0, 1, \dots, m-1$$

Los $16m \cdot n$ coeficientes a_{ijk_s} de la ecuación (16) deben determinarse de tal manera que S cumpla las condiciones (1) y (2). Con las condiciones de interpolación (1) se obtiene $a_{ij00} = u_{ij}$ para $i = 0, 1, \dots, n-1, \quad j = 0, 1, \dots, m-1$. Para el cálculo de los coeficientes a_{ijk_s} tendremos en cuenta las siguientes condiciones de contorno de las derivadas parciales de S

$$\begin{aligned}
\frac{\partial}{\partial x} S(x_i, y_j) &=: p_{ij} = a_{ij10}, & i = 0, 1, \dots, n, & \quad j = 0, 1, \dots, m \\
\frac{\partial}{\partial y} S(x_i, y_j) &=: q_{ij} = a_{ij01}, & i = 0, 1, \dots, n, & \quad j = 0, 1, \dots, m \\
\frac{\partial^2}{\partial x \partial y} S(x_i, y_j) &=: s_{ij} = a_{ij11}, & i = 0, 1, \dots, n, & \quad j = 0, 1, \dots, m
\end{aligned} \tag{17}$$

Una vez obtenidas las derivadas de la función u , calculamos.

$$\begin{aligned}
\frac{\partial}{\partial x} S(x_i, y_j) &= \frac{\partial}{\partial x} u(x_i, y_j), \\
\frac{\partial}{\partial y} S(x_i, y_j) &= \frac{\partial}{\partial y} u(x_i, y_j) \\
\frac{\partial^2}{\partial x \partial y} S(x_i, y_j) &= \frac{\partial^2}{\partial x \partial y} u(x_i, y_j)
\end{aligned}$$

En [De Boor(1962)] se demuestra como para determinados u_{ij} y sus derivadas parciales según (17) existe exactamente una función de splines bicúbica que interpola los u_{ij}

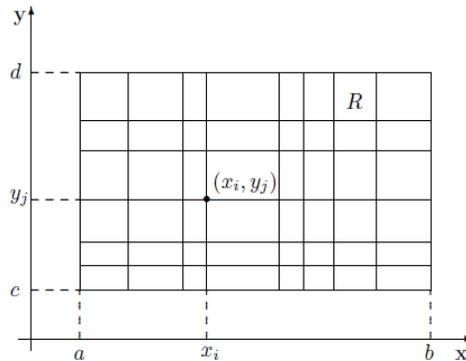


Figura 22: Grid rectangular (*Fuente:* [Engeln-Müllges(2010)])

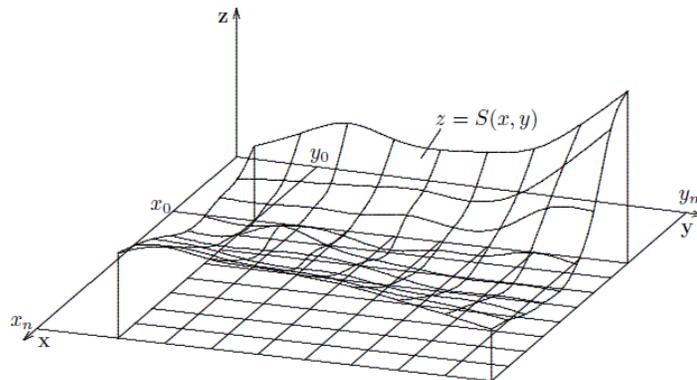


Figura 23: Gráfica de una función de splines bicúbicos (*Fuente:* [Engeln-Müllges(2010)])

10.3. Error cuadrático medio

El error cuadrático medio es una métrica estadística que cuantifica el error de un conjunto de mediciones con respecto a los datos originales, es decir nos otorga una herramienta para observar cuan alejados están los datos estimados por un determinado método, de sus valores ideales. De esta forma podemos comparar el poder de estimación de diferentes métodos en relación al nivel de error cometido, siendo mejor método aquel que obtenga el menor error cuadrático [James and Tibshirani(2013)].

El error cuadrático medio se calcula de la siguiente manera:

- **Primero:** Se calcula la distancia (diferencia) para cada medición, entre la valor obtenido por el método a analizar y el valor real, y esta distancia es elevada al cuadrado.
- **Segundo:** Se suman todas la diferencias obtenidas en el primer paso y luego se divide esta suma entre la cantidad de los datos.

Es necesario observar que la unidad del error cuadrático medio, es en función de el cuadrado de la unidad del valor original, lo que puede dificultar su interpretación, por lo cual es común tomar su raíz cuadrada para poder explicar la distancia que éste representa en relación a las valores de referencia.

La siguiente ecuación representa el cálculo del error cuadrático medio.

$$ECM(\hat{\theta}) = \frac{1}{N} \sum_{i=1}^N (\hat{\theta}_i - \theta_i)^2$$

Donde $\hat{\theta}_i$ representa el i-ésimo valor obtenido por el método a evaluar, θ_i el i-ésimo valor original, y N el numero total de datos.

10.4. Coeficiente de determinación (R cuadrado)

Una forma de evaluar la calidad del ajuste de un determinado método sobre los datos originales es mediante el coeficiente de determinación. Básicamente, este coeficiente es una métrica de la proporción de variabilidad en una variable que se explica por la variabilidad de la otra [Cameron and Windmeijer(1997)].

Definición: Definimos el vector de los datos originales como y_1, \dots, y_n , cada uno asociado con un valor ajustado (o valor estimado por el método a analizar) f_1, \dots, f_n .

Definimos los errores de medición como $e_i = y_i - f_i$.

Si \bar{y} es la media de los datos observados, entonces la variabilidad del conjunto de datos se puede medir con dos fórmulas de sumas de cuadrados:

- La suma total de cuadrados (proporcional a la varianza de los datos):

$$SC_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

- La suma de cuadrados de los errores, también llamada suma de cuadrados de residuos:

$$SC_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

En base a estas dos sumas de cuadrados, la definición del coeficiente de determinación sería:

$$R^2 = 1 - \frac{SC_{\text{res}}}{SC_{\text{tot}}}$$

10.5. Validación cruzada dejando uno fuera

La validación cruzada dejando uno fuera, del inglés *Leave-One-Out Cross-Validation* es un método de validación cruzada que consiste en separar un conjunto de datos de n elementos en dos grupos de $n - 1$ para construir el modelo y 1 elemento para validarlo, realizando este proceso n veces, una por cada elemento, obteniendo de esta forma n elementos estimados para comparar, una vez obtenido los n datos estimados por este proceso se puede comparar con los datos originales siguiendo métricas como el error cuadrático medio o el coeficiente de determinación, entre otras [Sammut and Webb(2010)].

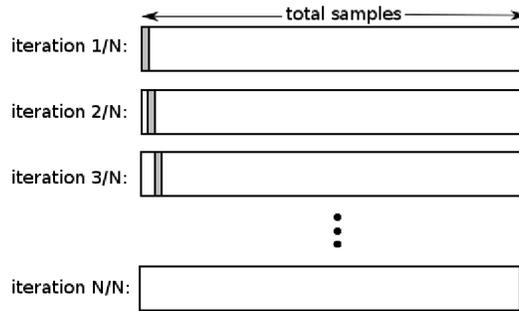


Figura 24: Validación cruzada dejando uno fuera

10.6. Código R

```
1 #cargar packages
2 library(quantmod)
3 library(rvest)
4 library(reshape2)
5 library(plotly)
6 library(akima)
7 # para importar a latex
8 library(xtable)
9 library(mgcv)
10 library(reticulate)
11 library(MLmetrics)
12 library(ggplot2)
13 #####
14 #####Atributos de configuracion
15 config.metodo <- "biseccion" #biseccion newton
16 config.interpolacion <- "spline" # akima, spline
17 config.precio <- "ask" # ask, bid
18 config.tipo <- "calls" # calls, puts
19
20
21 #settings para symbol, moneyness y vencimiento
22 #symbol <- "SPY"
23 symbol <- "^SPX"
24 moneynessRango <- c(0.75,1.25)
25 ttmRango <- c(7, 180)
26
27 #####
28
29 e <- new.env()
30 # hora local
31 #locale <- Sys.getlocale(category = "LC_TIME")
32 #setting local formato US
33 #Sys.setlocale("LC_TIME", "en_US.UTF-8")
34 #obtener info subyacente y precio de cierre
35 infoPrecio <- getQuote(symbol)
36 ultimoPrecio <- infoPrecio$Last
37 dividendos <- getQuote(symbol, what = yahooQF("Dividend Yield"))$'Dividend
  Yield'
38
39 if(is.na(dividendos)){dividendos <- 0}
40
41 fecha <- as.Date(infoPrecio$'Trade Time')
42
43 #webscraping de Yahoo Finance
44 baseUrl <- paste0("https://finance.yahoo.com/quote/",symbol,"/options")
45 baseHTML <- read_html(baseUrl)
46
47 #obtener vencimientos disponibles y convertir a dias
48 vencimientosUNIX <- baseHTML %>% html_nodes("option") %>% html_attr("value")
```

```

49 vencimientos <- as.Date((baseHTML %>% html_nodes("option") %>% html_text()),
  format = "%b %d, %Y")
50 timeToMats <- as.numeric(vencimientos - fecha)
51
52 #acotar vencimientos
53 sel <- timeToMats >= ttmRango[1] & timeToMats <= ttmRango[2]
54
55 vencimientosUNIX <- vencimientosUNIX[sel]
56 vencimientos <- vencimientos[sel]
57 timeToMats <- timeToMats[sel]
58
59 #bucle sobre vencimientos para obtener calls y puts
60 obtener_datos<-function(tipo, pausa = 0){
61   data_cp <- NULL
62   for(i in 1:length(vencimientosUNIX)){
63     vencimientoUrl <- paste0(baseUrl,"?date=",vencimientosUNIX[i])
64     vencimientoHTML <- read_html(vencimientoUrl)
65     tmpData <- vencimientoHTML %>% html_nodes(paste0(".",tipo))%>% html_
      table()
66
67     if(length(tmpData) > 0){
68       tmpData <- tmpData[[1]]
69       #nombres columnas algunas veces en mayuscula
70       colnames(tmpData) <- tolower(colnames(tmpData))
71       #quitar separador de miles y convertir a formato numerico
72       tmpData$strike <- as.numeric(gsub(",","",tmpData$strike))
73       #a adir plazo de vencimiento
74       tmpData$ttm <- timeToMats[i]
75       #calcular moneyness
76       tmpData$moneyness <- tmpData$strike/ultimoPrecio #puts:ultimoPrecio/
      tmpData$strike
77       #convertir volatilidad implicita de YahooFinance a formato numerico
78       tmpData$ivOrig <- as.numeric(gsub("%","",gsub(",","",tmpData$'implied
      volatility')))/100
79       data_cp <- rbind(data_cp, tmpData)
80       print(i)
81       Sys.sleep(pausa)
82     }
83   }
84   return(data_cp)
85 }
86
87 ##### Black-Scholes #####
88 #####
89
90 #generalized black scholes merton
91 gBSM <- function(S, X, sigma, r, q, ttm, tipo){
92   ##S = precio subyacente , X = strike , sigma = volatilidad , r = interes
      libre
93   #de riesgo , q = dividendos , ttm = plazo de vencimiento en dias , tipo =
      tipo de opcion
94

```

```

95 b <- r - q
96 t <- ttm/365.25
97
98 d1 <- (log(S / X) + (b + sigma^2 / 2) * t) / (sigma * sqrt(t))
99 d2 <- d1 - sigma * sqrt(t)
100
101 if(tipo == "calls"){
102   precio <- S * exp((b - r) * t) * pnorm(d1) - X * exp(-r * t) * pnorm(d2)
103 }else if (tipo == "puts"){
104   precio <- (X * exp(-r * t) * pnorm(-d2) - S * exp((b - r) * t) * pnorm
105   (-d1))
106 }
107 return(precio)
108 }
109 #####Biseccion#####
110 #####
111
112 e$iterBiseccion<-c()
113 biseccion <- function(S, K, r, q, ttm, tipo, precio){
114   sig <- 0.20
115   sig.up <- 2
116   sig.down <- 0.001
117
118   vueltas <- 0
119   err <- gBSM(S, K, sig, r, q, ttm, tipo) - precio
120   # print(precio)
121   # print(gBSM(S, K, sig, r, q, ttm, type))
122   # print(err)
123   # repetir hasta que el error sea suficientemente peque o o itineraciones
124   # lleguen a 1000
125   iteraciones <- 0
126   while(abs(err) > 0.00001 && vueltas<1000){
127     if(err < 0){
128       sig.down <- sig
129       sig <- (sig.up + sig)/2
130     }else{
131       sig.up <- sig
132       sig <- (sig.down + sig)/2
133     }
134     err <- gBSM(S, K, sig, r, q, ttm, tipo) - precio
135     vueltas <- vueltas + 1
136     #print (err)
137     iteraciones<-iteraciones + 1
138   }
139   e$iterBiseccion<-c(e$iterBiseccion,iteraciones)
140   # devuelve NA si contador llega a 1000
141   if(vueltas==1000){
142     return(NA)
143   }else{
144     return(sig)
145   }

```

```

145 }
146
147
148 ##### Newton #####
149 #####
150
151 bs_price <- function(cp_flag,S,K,T,r,v,q=0.0) {
152   n = dnorm
153   N = pnorm
154
155   d1 <- (log(S/K)+(r+v*v/2.)*T)/(v*sqrt(T))
156   d2 <- d1-v*sqrt(T)
157   if (cp_flag == 'c') {
158     precio <- S*exp(-q*T)*N(d1)-K*exp(-r*T)*N(d2)
159   }
160   else {
161     precio <- K*exp(-r*T)*N(-d2)-S*exp(-q*T)*N(-d1)
162   }
163   return(precio)
164 }
165
166 bs_vega <- function(cp_flag,S,K,T,r,v,q=0.0) {
167   n = dnorm
168   N = pnorm
169
170   d1 <- (log(S/K)+(r+v*v/2.)*T)/(v*sqrt(T))
171   return(S * sqrt(T)*n(d1))
172 }
173
174 e$iterNewton <- c()
175 newton = function(S, K, r, q = 0, ttm, tipo, precio){
176   #find_vol <- function(valor_objetivo, call_put, S, K, T, r) {
177   #mapping
178   T <- ttm/365.25
179   call_put <- ifelse(tipo=='calls','c','p')
180   valor_objetivo = precio
181
182   MAX_ITERATIONS <- 1000
183   PRECISION <- 1.0e-5
184   call_put <- tolower(call_put)
185   iteraciones <- 0
186   sigma <- 0.5
187   for (i in 1: MAX_ITERATIONS) {
188     precio <- bs_price(call_put, S, K, T, r, sigma)
189     vega <- bs_vega(call_put, S, K, T, r, sigma)
190     diff <- valor_objetivo - precio # our root
191     if (abs(diff) < PRECISION | vega < PRECISION) {
192       e$iterNewton<- c(e$iterNewton,iteraciones)
193       return(sigma)
194     }
195     sigma <- sigma + diff/vega # f(x) / f'(x)
196     iteraciones <- iteraciones + 1

```

```

197   }
198   e$iterNewton<- c(e$iterNewton,iteraciones)
199   return(sigma)
200 }
201 #####
202 ##### configurar newton o biseccion #####
203
204 getIV <- function(x, S, r, q, tipo, tipo_precio, function_app){
205   res <- function_app(S = S,
206                       K = as.numeric(x["strike"]),
207                       r = r,
208                       q = q,
209                       ttm = as.numeric(x["ttm"]),
210                       tipo = tipo,
211                       precio = as.numeric(x[tipo_precio])
212   )
213   return(res)
214 }
215
216 #ivGridCalls<-ivGridCalls_r
217 plotear_puntos<-function(grid){
218
219   pointsGrid <- melt(grid)
220   pointsGrid <- na.omit(pointsGrid)
221   #EJES Y LEYENDAS
222   xaxx <- list(
223     gridcolor='rgb(255, 255, 255)',
224     zerolinecolor='rgb(255, 255, 255)',
225     showbackground=TRUE,
226     backgroundcolor='rgb(230, 230,230)',
227     title = "Moneyness"
228   )
229
230
231   yaxx <- list(
232     gridcolor='rgb(255, 255, 255)',
233     zerolinecolor='rgb(255, 255, 255)',
234     showbackground=TRUE,
235     backgroundcolor='rgb(230, 230,230)',
236     title = "Time To Maturity"
237   )
238
239   zaxx <- list(
240     gridcolor='rgb(255, 255, 255)',
241     zerolinecolor='rgb(255, 255, 255)',
242     showbackground=TRUE,
243     backgroundcolor='rgb(230, 230,230)',
244     tickformat = "%",
245     title = "Implied Volatility"
246   )
247   fig_p <- plot_ly(pointsGrid, x = ~Var2, y = ~Var1, z = ~value, size=0.1)#,
     color = ~am, colors = c('#BF382A', '#0C4B8E'))

```

```

248 fig_p <- fig_p %>% layout(scene = list(xaxis=xaxx, yaxis=yaxx, zaxis =
      zaxx))
249 fig_p
250 }
251
252
253 ##### interpolacion splines cubicos univariada #####
254 #####
255 interpol_spline<-function(grid_input, metodo){
256   grid<-grid_input
257   for (i in 1:nrow(grid)){
258     line_test<-grid[i,]
259     lineXY<-NULL
260     lineXY$X<-as.numeric(names(line_test))
261     lineXY$Y<-line_test
262     lineXY<-data.frame(lineXY)
263     lineXY<-na.omit(lineXY)
264     interpolador <- splinefun(lineXY, method = metodo)
265     valores_interpolador<-as.numeric(names(line_test[is.na(line_test)]))
266     line_test[is.na(line_test)]<-interpolador(valores_interpolador)
267     # if(interpVal < 0 | interpVal > max(interpGrid$value * 1.5)){
268     #   interpVal <- NA
269     # }
270     line_test[line_test < 0]<-0
271     line_test[line_test > max(lineXY$Y)*1.5]<-0
272     grid[i,]<-line_test
273   }
274
275   outs<- boxplot(grid, plot=FALSE)$out
276   grid<- grid[-which(grid %in% outs),]
277   pointsGrid <- melt(grid)
278   pointsGrid <- na.omit(pointsGrid)
279   colnames(pointsGrid)<-c("x","y","z")
280   mod <- gam(z ~ te(x, y), data = pointsGrid)
281   pointsGrid$z <- fitted(mod)
282   return(acast(pointsGrid, x ~ y, value.var = "z"))
283 }
284
285 ##### interpolacion splinefun punto x punto #####
286 #####
287 interpol_spline_ptp<-function(grid_input, metodo)
288   #grid<-ivGridCalls
289   grid<-grid_input
290   i=1
291   for (i in 1:nrow(grid)){
292     while(TRUE){
293       line_test<-grid[i,]
294       lineXY<-NULL
295       lineXY$X<-as.numeric(names(line_test))
296       lineXY$Y<-line_test
297       lineXY<-data.frame(lineXY)
298       valores_interpolador<-as.numeric(names(line_test[is.na(line_test)]))

```

```

299     if(length(valores_interpolar)==0) break
300
301     lineXY<-na.omit(lineXY)
302     interpolador <- splinefun(lineXY, method = metodo)
303     m_p<-valores_interpolar[length(valores_interpolar)%/2 +1]
304     v_mp<-interpolador(m_p)
305     line_test[as.character(m_p)]<-v_mp
306
307     line_test[line_test < 0]<-0
308     line_test[line_test > max(lineXY$Y)*1.5]<-0
309     grid[i,]<-line_test
310   }
311 }
312 return(grid)
313 }
314
315 #####interpolacion splines bicubica#####
316 #####interpolacion splines bicubica#####
317 interpolador_akima<-function(input_grid){
318   grid<-input_grid
319   #obtener coordenadas de NAs en grid
320   toInterpolate <- which(is.na(grid))
321   coordinadass <- cbind(toInterpolate%%dim(grid)[1], toInterpolate%%dim(
322     grid)[1] + 1)
323   coordinadass[coordinadass[,1] == 0, 2] <- coordinadass[coordinadass[,1] ==
324     0, 2] - 1
325   coordinadass[coordinadass[,1] == 0, 1] <- dim(grid)[1]
326
327   # bucle NAs e interpola
328   for(i in 1:nrow(coordinadass)){
329     #coordenadas area 10 x10 sobre punto faltante
330     x1 <- max(coordinadass[i,1] - 10, 1)
331     x2 <- min(coordinadass[i,1] + 10, dim(grid)[1])
332     y1 <- max(coordinadass[i,2] - 10, 1)
333     y2 <- min(coordinadass[i,2] + 10, dim(grid)[2])
334
335     # obtener moneyness / plazo vencimiento del punto faltante
336     x0 <- as.numeric(rownames(grid)[coordinadass[i,1]])
337     y0 <- as.numeric(colnames(grid)[coordinadass[i,2]])
338
339     ## coordenadas del grid usadas para interpolar y eliminar los NAs que
340     aparezcan
341     interpGrid <- grid[x1:x2,y1:y2]
342     interpGrid <- melt(interpGrid)
343     interpGrid <- na.omit(interpGrid)
344
345     #opcion interpolacion lineal desactivar
346     #interpVal <-NA
347     interpVal <-interp(x = interpGrid$Var1, y = interpGrid$Var2, z =
348     interpGrid$value,
349     xo = x0, yo = y0,
350     linear = TRUE, extrapol = TRUE)$z[1,1]

```

```

347
348   #interpolacion splines bicubicos
349   if(is.na(interpVal)){
350     interpVal <- interp(x = interpGrid$Var1, y = interpGrid$Var2, z =
interpGrid$value,
351                       xo = x0, yo = y0,
352                       linear = FALSE, extrap = TRUE)$z[1,1]
353   }
354
355   #si el valor obtenido es erroneo (negativo o un outlier) dejarlo como NA
356   if(interpVal < 0 | interpVal > max(interpGrid$value * 1.5)){
357     interpVal <- NA
358   }
359   #sustituir valor por resultado de la interpolacion
360   grid[coordinadass[i,1],coordinadass[i,2]] <- interpVal
361 }
362
363 return(grid)
364 }
365
366 # plot superficie volatilidad interpolada
367 plot_surface<-function(grid){
368
369   xaxx <- list(
370     # orientation = "h",
371     gridcolor='rgb(255, 255, 255)',
372     zerolinecolor='rgb(255, 255, 255)',
373     showbackground=TRUE,
374     backgroundcolor='rgb(230, 230,230)',
375     title = "Moneyness"
376   )
377
378   yaxx <- list(
379     # orientation = "h",
380     gridcolor='rgb(255, 255, 255)',
381     zerolinecolor='rgb(255, 255, 255)',
382     showbackground=TRUE,
383     backgroundcolor='rgb(230, 230,230)',
384     title = "Time To Maturity"
385   )
386
387   zaxx <- list(
388     # orientation = "v",
389     gridcolor='rgb(255, 255, 255)',
390     zerolinecolor='rgb(255, 255, 255)',
391     showbackground=TRUE,
392     backgroundcolor='rgb(230, 230,230)',
393     tickformat = "%",
394     title = "Implied Volatility"
395   )
396

```

```

397 fig <- plot_ly(x = as.numeric(colnames(grid)), y = as.numeric(rownames(
  grid)), z = grid)
398 fig <- fig %>% add_surface()
399 fig <- fig %>% layout(scene = list(xaxis=xaxx, yaxis=yaxx, zaxis = zaxx))
400 fig <- fig %>% plotly::colorbar(title = "", x = 0.9, y = 0.75, tickformat
  = "%")
401 fig
402 }
403
404 #####
405 #####Estimar volatilidad implicita con grid#####
406 #####
407 estimate_pos<-function(v, ele){
408   inf<-1
409   sup<-length(v)
410   while(TRUE){
411     #posicionarme en la mitad
412     pos<-(inf+sup)%/%2
413     #verificar si esta en el medio
414     if(v[pos] == ele){
415       return(pos)#si existe el elemento
416     }else if((sup-inf) <= 1){#elemento no existe retornar aproximado
417       if((v[sup] - ele) < (v[inf] - ele))
418         return(sup)
419       else
420         return(inf)
421     }
422     else{
423       if(ele < v[pos])
424         sup <- pos - 1
425       else
426         inf <- pos + 1
427     }
428   }
429 }
430 e$cubico <- 0
431 estimate_iv<-function(grid, x0, y0){
432
433   xp<-estimate_pos(as.numeric(rownames(grid)),x0)
434   yp<-estimate_pos(as.numeric(colnames(grid)),y0)
435
436   x1 <- max(xp - 10, 1)
437   x2 <- min(xp + 10, dim(grid)[1])
438   y1 <- max(yp - 10, 1)
439   y2 <- min(yp + 10, dim(grid)[2])
440
441
442 #coordenadas del grid usadas para interpolar y eliminar NAs que aparezcan
443 interpGrid <- grid[x1:x2,y1:y2]
444 #interpGrid <- grid
445 interpGrid <- melt(interpGrid)
446 interpGrid <- na.omit(interpGrid)

```

```

447
448 #interpolacion lineal FALSE
449 #interpVal <-NA
450 # interpVal <-interp(x = interpGrid$Var1, y = interpGrid$Var2, z =
      interpGrid$value,
451 #           xo = x0, yo = y0,
452 #           linear = TRUE, extrap = TRUE)$z[1,1]
453 #
454 #interpolacion splines bicubica
455 # if(is.na(interpVal)){
456   e$cubico <- e$cubico + 1
457   interpVal <- interp(x = interpGrid$Var1, y = interpGrid$Var2, z =
      interpGrid$value,
458 #           xo = x0, yo = y0,
459 #           linear = FALSE, extrap = TRUE)$z[1,1]
460 # }
461
462 #si el valor obtenido es erroneo ( negativo o un outlier ) dejarlo como NA
463 if(interpVal < 0 | interpVal > max(interpGrid$value * 1.5)){
464   interpVal <- NA
465 }
466 return(interpVal)
467 }
468
469 estimar<-function(grid,x){
470   res <- estimate_iv (grid = grid,
471                       x0 = as.numeric(x["ttm"]),
472                       y0 = as.numeric(x["moneyness"])
473   )
474   return(res)
475 }
476
477 #####Obtener los datos#####
478 #####
479
480 data_cp <-obtener_datos(config.tipo)
481
482 back_data<-data_cp
483 #data_cp <- back_data
484 # filtrar calls que esten dentro del rango moneyness
485 data_cp <- data_cp[data_cp$moneyness >= moneynessRango[1] & data_cp$
      moneyness <= moneynessRango[2],]
486 ## filtrar calls que hayan cotizado durante ultimo dia de cotizacion
487 data_cp <- data_cp[strptime(data_cp$'last trade date',format = "%Y-%m-%d %I
      :%M%p") >= strptime(paste0(fecha," 9:00AM EDT"), format = "%Y-%m-%d %I:%M
      %p"),]
488
489
490 metodo = ifelse(config.metodo == "biseccion", biseccion, newton)
491 data_cp$iv <- apply(data_cp,
492                    1,
493                    getIV,

```

```

494         S = ultimoPrecio ,
495         r = 0.0011,
496         q = dividendos ,
497         tipo = config.tipo,
498         tipo_precio=config.precio,
499         function_app = metodo)
500 #crear grids
501 ivGrid <- acast(data_cp, ttm ~ moneyness, value.var = "iv")
502
503
504 if(config.interpolacion == "akima")
505     ivGridInterpolada <- interpolar_akima(ivGrid)
506
507 if(config.interpolacion == "spline")
508     ivGridInterpolada <- interpolar_spline(ivGrid, metodo = "natural")
509
510
511 ff<-row.names(ivGrid) %in% row.names(ivGridInterpolada)
512 ivGrid<-ivGrid[ff,]
513 ### Plotear
514 plotear_puntos(ivGrid)
515 plotear_puntos(ivGridInterpolada)
516
517 plot_surface(ivGridInterpolada)
518
519 data_cp_all <- data_cp
520 data_cp <- data_cp[data_cp$ttm %in% row.names(ivGridInterpolada),]
521 # importar tablas a latex
522 xts<-xtable(data_cp[1:10,], caption = config.tipo)
523 print.xtable(xts, type="latex", file="tables.tex", append = T)
524
525 #####
526 #####Validacion#####
527
528 #calcular puntos de test con la interpolacion
529
530 #dividir datos en entrenamiento y test
531
532 data_cp_bck <- data_cp
533 ## 90 % de la muestra de observaciones
534 smp_size <- floor(0.90 * nrow(data_cp))
535
536 ## semilla
537 set.seed(1234)
538 train_ind <- sample(seq_len(nrow(data_cp)), size = smp_size)
539 data_cp <- data_cp_bck[train_ind, ]
540 test_data_cp <- data_cp_bck[-train_ind, ]
541
542 # crear grid con solo los datos de entrenamiento
543 ivGrid_t <- acast(data_cp, ttm ~ moneyness, value.var = "iv")
544 #interpolar
545

```

```

546 #if(config.interpolacion == "akima")
547 #   int_t <- interpolar_akima(ivGrid_t)
548
549 #if(config.interpolacion == "spline")
550   int_t <- interpolar_spline(ivGrid_t, metodo = "natural")
551
552 test_data_cp$ivInter <- apply(test_data_cp, 1, estimar, grid = int_t)
553 #   estimate_iv(grid = int_t,
554 #               test_data_cp$ttm,
555 #               test_data_cp$moneyness)
556 # estimate_iv(grid = int_t, 98, 1.0176782)
557 #
558 # warning()
559 # apply(test_data_cp, 1, estimar, grid = int_t)
560
561 test_data_cp <- test_data_cp[ !is.na(test_data_cp$ivInter),]
562
563 erroIv <- MSE(test_data_cp$iv, test_data_cp$ivOrig)
564 erroInter <- MSE(test_data_cp$ivInter, test_data_cp$ivOrig)
565 diff_errInter <- erroInter - erroIv
566
567 r2iv<-R2_Score(test_data_cp$iv, test_data_cp$ivOrig)
568 r2Int<-R2_Score(test_data_cp$ivInter, test_data_cp$ivOrig)
569 diff_inter<-r2iv-r2Int
570
571 ###diferencias
572 #par(mfrow=c(1,1))
573
574 data_cp$diff=abs(data_cp$ivOrig - data_cp$iv)
575 ggplot(data=data_cp, aes(x=ttm,y=diff))+
576   geom_point()+
577   ylab(label = "Diferencia")+
578   xlab(label = "ttm")
579 #plot(x=calls$ttm, y=calls$diff,type="p")
580
581 #par(mfrow=c(1,3))
582
583 df<-data.frame(ttm=test_data_cp$ttm ,
584               diff=abs(test_data_cp$ivOrig - test_data_cp$iv),
585               valor="original")
586 df<-rbind(df,data.frame(ttm=test_data_cp$ttm,
587                       diff=abs(test_data_cp$ivOrig - test_data_cp$ivInter)
588                       ,
589                       valor="interpolado"))
590 ggplot(data = df, aes(x = ttm, y = diff))+
591   geom_point()+
592   facet_grid(~valor)+
593   ylab(label = "Diferencia")+
594   xlab(label = "ttm")
595
596

```

```

597 data_cp<-data_cp_bck
598
599 #####
600 #####
601 library("tidyverse")
602
603 pointsGrid <- melt(ivGridInterpolada)
604 pointsGrid <- na.omit(pointsGrid)
605 colnames(pointsGrid)<-c("ttm","strike","iv")
606 pointsGrid$ttm <- as.character(pointsGrid$ttm)
607
608 data_cp$ttm <- as.character(data_cp$ttm)
609
610 p1 <- ggplot(data = pointsGrid, aes(x = strike, y = iv, group = ttm, color=
    ttm))+
611   #geom_point(size = 0.1)+
612   geom_line()+
613   theme(axis.title.x = element_blank(), axis.text.x = element_blank(),legend
    .position = "top")
614
615 data_cp$'open interest' <- as.numeric(gsub(",","",data_cp$'open interest'))
616 p2 <- ggplot(data_cp, aes(x = strike, y = 'open interest')) +
617   geom_bar(stat="identity") +
618   theme_minimal() +
619   theme(axis.text.x = element_text(angle=90))
620
621 p3 <- ggplot(data = data_cp, aes(x = strike, y = iv, group = ttm, color=ttm)
    )+
622   #geom_point(size = 0.1)+
623   geom_line()+
624   theme(axis.title.x = element_blank(), axis.text.x = element_blank(),legend
    .position = "top")
625
626 p4 <- ggplot(data = data_cp, aes(x = strike, y = iv, group = ttm, color=ttm)
    )+
627   #geom_point(size = 0.1)+
628   geom_line()+
629   theme(axis.text.x = element_text(angle=90), legend.position = "none")
630
631
632 library(grid)
633 grid.newpage()
634 grid.draw(rbind(ggplotGrob(p1), ggplotGrob(p2), size = "last"))
635
636 grid.newpage()
637 grid.draw(rbind(ggplotGrob(p3), ggplotGrob(p2), size = "last"))
638
639 grid.newpage()
640 grid.draw(rbind(ggplotGrob(p1), ggplotGrob(p4), size = "last"))
641
642 ##### Escojer metodo de busqueda de raiz#####
643 #####

```

```

644 data_cp_bck<-data_cp
645
646 e$iterNewton <- c()
647 data_cp$ivNewton <- apply(data_cp,
648                             1,
649                             getIV,
650                             S = ultimoPrecio,
651                             r = 0.0011,
652                             q = dividendos,
653                             tipo = config.tipo,
654                             tipo_precio=config.precio,
655                             function_app = newton)
656 e$iterBiseccion <- c()
657 data_cp$ivBiseccion <- apply(data_cp,
658                               1,
659                               getIV,
660                               S = ultimoPrecio,
661                               r = 0.0011,
662                               q = dividendos,
663                               tipo = config.tipo,
664                               tipo_precio=config.precio,
665                               function_app = biseccion)
666
667 nb<-data.frame(iteraciones=e$iterNewton, metodo = "Newton")
668 nb<-rbind(data.frame(iteraciones=e$iterBiseccion, metodo = "Biseccion"), nb
669 )
670 ggplot(data = nb,aes(metodo,iteraciones))+
671   geom_boxplot()+
672   xlab(label = "M  todo")+
673   ylab(label = "N  de iteraciones")
674
675 mediaNew <- mean(e$iterNewton)
676 sdNew <- sd(e$iterNewton)
677
678 mediaBis <- mean(e$iterBiseccion)
679 sdBis <- sd(e$iterBiseccion)
680
681 #####comparar biseccion y newton
682
683 erro_b <- MSE(data_cp$ivBiseccion,data_cp$ivOrig)
684 erro_n <- MSE(data_cp$ivNewton,data_cp$ivOrig)
685 diff_err <- erro_b - erro_n
686
687 r2_b <- R2_Score(data_cp$ivBiseccion,data_cp$ivOrig)
688 r2_n <- R2_Score(data_cp$ivNewton,data_cp$ivOrig)
689 diff_r2 <- r2_b - r2_n
690
691 #####Comparacion metodos#####
692 #####Histograma#####
693 library("tidyverse")
694

```

```

695 df <- rownames_to_column(data_cp, var="timeToMats") %>%
696   select(ttm, timeToMats, strike, ivOrig, ivNewton, ivBisection) %>%
697   gather(key = "Metodo", value = "IV", -strike, -timeToMats, -ttm)
698
699 p1 <- ggplot(df, aes(x = strike, y = IV)) +
700   geom_line(aes(color = Metodo, linetype = Metodo)) +
701   scale_color_manual(values = c("darkred", "springgreen", "steelblue"))+
702   #theme_minimal() +
703   theme(axis.title.x = element_blank(), axis.text.x = element_blank(), legend
704         .position = "top")
705
706 data_cp$'open interest' <- as.numeric(gsub(",", "", data_cp$'open interest'))
707
708 p2 <- ggplot(data_cp, aes(x = strike, y = 'open interest')) +
709   geom_bar(stat="identity") +
710   #theme_minimal() +
711   theme(axis.text.x = element_text(angle=90))
712
713 library(grid)
714 grid.newpage()
715 grid.draw(rbind(ggplotGrob(p1), ggplotGrob(p2), size = "last"))
716
717 #####Validacion Interp Bivariada#####
718 #####
719
720 data_cp_i <- data_cp
721 data_cp <- data_cp_i
722 df_base <- data_cp[,c("ttm", "moneyness", "iv")]
723 df_base$ttm <- as.numeric(df_base$ttm)
724
725 estimar_cross <- function(ttm, moneyness, df){
726   df_c <- data.frame(df)
727   if (ttm == min(df_c$ttm) | ttm == max(df_c$ttm))
728     #print("na")
729     return(NA)
730   if (moneyness == min(df_c$moneyness) | moneyness == max(df_c$moneyness))
731     #print("mn")
732     return(NA)
733   #filtrar datos
734   df_c <- df_c[df_c$ttm != ttm,]
735   df_c <- df_c[df_c$moneyness != moneyness,]
736   #contruir grid
737   ivGrid_t <- acast(df_c, ttm ~ moneyness, value.var = "iv")
738   #rellenar grid con interpolacion de splines cubicos
739   ivGrid_t <- interpolar_spline(ivGrid_t, "natural")
740   #estimar IVs
741   return(estimate_iv(ivGrid_t, ttm, moneyness))
742 }
743
744 ivBi <- c()
745 for (i in 1:nrow(data_cp)){

```

```

746 result = tryCatch({
747   estimar_cross(df_base[i,"ttm"], df_base[i,"moneyness"], df=df_base)
748 }, error = function(error_condition) {
749   return(NA)
750 })
751 ivBi <- c(ivBi,result)
752 }
753 #estimar_cross(df_base[2,"ttm"], df_base[2,"moneyness"], df=df_base)
754
755 data_cp$ivBi <- ivBi
756
757 data_cp <- data_cp[!is.na(data_cp$ivBi),]
758
759 erroIv <- MSE(data_cp$iv, data_cp$ivOrig)
760 erroIvBi <- MSE(data_cp$ivBi, data_cp$ivOrig)
761 diff_errBi <- erroIvBi - erroIv
762
763 r2iv<-R2_Score(data_cp$iv, data_cp$ivOrig)
764 r2Bi<-R2_Score(data_cp$ivBi, data_cp$ivOrig)
765 diff_bi<-r2iv-r2Bi
766
767 ###diferencias
768 #par(mfrow=c(1,1))
769
770 data_cp$diff_Bi=abs(data_cp$ivOrig - data_cp$ivBi)
771 ggplot(data=data_cp, aes(x=ttm,y=diff_Bi))+
772   geom_point()+
773   ylab(label = "Diferencia")+
774   xlab(label = "ttm")
775 #plot(x=calls$ttm, y=calls$diff,type="p")
776
777 #par(mfrow=c(1,3))
778
779 df<-data.frame(ttm=data_cp$ttm ,
780               diff=abs(data_cp$ivOrig - data_cp$iv),
781               valor="Bisecci n/Newton")
782 df<-rbind(df,data.frame(ttm=data_cp$ttm,
783                       diff=abs(data_cp$ivOrig - data_cp$ivBi),
784                       valor="Bivariado"))
785
786 ggplot(data = df, aes(x = ttm, y = diff))+
787   geom_point()+
788   facet_grid(~valor)+
789   ylab(label = "Diferencia")+
790   xlab(label = "ttm")
791
792 data_cp<-data_cp_bck

```

Listing 1: Código Fuente

11. Bibliografía

Referencias

- [Analytics(2021)] Analytics, G., 2021. Machine Learning Crash Course — Google Developers. URL: <https://developers.google.com/machine-learning/crash-course>. (Accessed on 01/18/2021).
- [Bronstein and Semendjajew(2005)] Bronstein, I.N., Semendjajew, 2005. Taschenbuch der Mathematik ((6. Auflage) ed.). Verlag Harri Deutsch, Frankfurt am Main.
- [Burden and Faires(2011)] Burden, R., Faires, J., 2011. Numerical analysis, 9th international edition. Brooks/Cole, Cencag Learning .
- [Cameron and Windmeijer(1997)] Cameron, A.C., Windmeijer, F.A., 1997. An r-squared measure of goodness of fit for some common nonlinear regression models. *Journal of econometrics* 77, 329–342.
- [De Boor(1962)] De Boor, C., 1962. Bicubic spline interpolation. *Journal of mathematics and physics* 41, 212–218.
- [Derman and Miller(2016)] Derman, E., Miller, M.B., 2016. The volatility smile. John Wiley & Sons.
- [Engeln-Müllges(2010)] Engeln-Müllges, 2010. Numerik-Algorithmen: Verfahren, Beispiele, Anwendungen. Springer-Verlag.
- [Hull et al.(2009)] Hull, J., et al., 2009. Options, futures and other derivatives/John C. Hull. Upper Saddle River, NJ: Prentice Hall,.
- [James and Tibshirani(2013)] James, Hastie, T., Tibshirani, 2013. An introduction to statistical learning. volume 112. Springer.
- [Manaster and Koehler(1982)] Manaster, S., Koehler, G., 1982. The calculation of implied variances from the black-scholes model: A note. *The Journal of Finance* 37, 227–230.
- [Merk(2011)] Merk, A., 2011. Optionsbewertung in Theorie und Praxis: Theoretische und empirische Überprüfung des Black/Scholes-Modells. Springer-Verlag.
- [Natenberg and Cohen(1994)] Natenberg, S., Cohen, J.M., 1994. Option volatility & pricing: advanced trading strategies and techniques. volume 192. McGraw-Hill New York.
- [Neftci(2008)] Neftci, S.N., 2008. Principles of financial engineering. Academic Press.
- [Pan(2019)] Pan, J., 2019. Empirical Asset Pricing Notes. Shanghai Advanced Institute of Finance, Shanghai Jiao Tong University.

- [Roch(2020)] Roch, O., 2020. Stochastic Finance Notes. Facultad de Economía y Empresa, Universidad de Barcelona.
- [Rubinstein(1994)] Rubinstein, M., 1994. Implied binomial trees. *The journal of finance* 49, 771–818.
- [Sammut and Webb(2010)] Sammut, C., Webb, G.I. (Eds.), 2010. Leave-One-Out Cross-Validation. Springer US, Boston, MA. pp. 600–601. URL: https://doi.org/10.1007/978-0-387-30164-8_469, doi:10.1007/978-0-387-30164-8_469.