



# הנדסה פיננסית באמצעות שפת Python: סימטריה-על בין אופציית רכש לאופציית מכר (Put-Call Supersymmetry)

סימטריה שימושית בין אופציית רכש (Call) ואופציית מכר (Put) נתונה על ידי:

$$c(S, X, T, r, b, \sigma) = p(-S, -X, T, r, b, -\sigma)$$

כאשר:

- $b = r$  עבור המודל של Black and Scholes (1973) לתמחור אופציות על מניות שאינן מחלקות דיבידנד.
- $b = r - q$  עבור המודל של Merton (1973) לתמחור אופציות על מניות שמחלקות תשואת דיבידנד רציפה.
- $b = 0$  עבור המודל של Black (1976) לתמחור אופציות על חוזים עתידיים.
- $b = r - rf$  עבור המודל של Garman and Kohlhagen (1983) לתמחור אופציות על מטבעות.

כלומר, הזנת מחיר נכס בסיס שלילי, מחיר מימוש שלילי ותנודתיות שלילית עבור אופציית מכר תניב שווי זהה לזה של אופציית רכש. אם נניח את טרנספורמציה מצב-מרחב הבאה:

$$k \times c(S, X, T, r, b, \sigma) = c(k \times S, k \times X, T, r, b, \sigma)$$

ביחד עם הסימטריה שלעיל, נקבל את סימטריה-העל שבין אופציית רכש לאופציית מכר:

$$c(S, X, T, r, b, \sigma) = -p(S, X, T, r, b, -\sigma)$$

ובאופן טבעי:

$$p(S, X, T, r, b, \sigma) = -c(S, X, T, r, b, -\sigma)$$

שווייה של אופציית רכש שקול אפקטיבית מכל הבחינות המימוניות המהותיות לשווי של מינוס אופציית מכר עם תנודתיות שלילית, ולהיפך. התוצאה מפסדת את הפיתוח והיישום של חישובי אופציות רבים.



למעשה, אין עוד צורך לפתח נוסחאות נפרדות עבור אופציות רכש ומכר. ניתן להרחיב את סימטריית-העל גם לאופציות אקזוטיות רבות וגם לאופציות אמריקאיות.

לפרטים נוספים על סימטריית-העל כמו גם על תנודתיות שלילית, ראו Adamchuk (1998), Peskir and Shiryaev (2001), Haug (2002) ו-Aase (2004).

(2004) (2001), Haug (2002), and Aase

## קוד ה-Python שפיתח האקטואר רועי פולניצר עבור אופציית רכש (Call)

```
import numpy as np
import scipy.stats as si
def PolanitzerSupersymmetryCall1(S,X,r,T,v):
    d1 = (np.log(S/X)+(r+0.5*(-v)**2)*T)/(-v*np.sqrt(T))
    d2 = (np.log(S/X)+(r-0.5*(-v)**2)*T)/(-v*np.sqrt(T))
    PolanitzerSupersymmetryCall1 = -(X*si.norm.cdf(-d2,0.0,1.0)*np.exp(-r*T)\
        -S*si.norm.cdf(-d1,0.0,1.0))
    print(PolanitzerSupersymmetryCall1)
```

```
PolanitzerSupersymmetryCall1(100,80,0.06,4,0.2)
```

```
38.98377035690456
```

```
PolanitzerBlackScholesCall(100,80,0.06,4,0.2)
```

```
38.98377035690456
```



## קוד ה-Python שפיתח האקטואר רועי פולניצר עבור אופציית מכר (Put)

```
import numpy as np
import scipy.stats as si
def PolanitzerSupersymmetryPut1(S,X,r,T,v):
    d1 = (np.log(S/X)+(r+0.5*(-v)**2)*T)/(-v*np.sqrt(T))
    d2 = (np.log(S/X)+(r-0.5*(-v)**2)*T)/(-v*np.sqrt(T))
    PolanitzerSupersymmetryPut1 = -(S*si.norm.cdf(d1,0.0,1.0)\
        -X*si.norm.cdf(d2,0.0,1.0)*np.exp(-r*T))
    print(PolanitzerSupersymmetryPut1)
```

```
PolanitzerSupersymmetryPut1(100,80,0.06,4,0.2)
```

1.9139992422288366

```
PolanitzerBlackScholesPut(100,80,0.06,4,0.2)
```

1.9139992422288366



פירמת הייעוץ שווי פנימי מסייעת ללקוחותיה לפתח וליישם מודלים מתקדמים הדורשים הבנה עמוקה בתהליכים סטוכסטיים, ידע בשיטות נומריות ושליטה ברמה גבוהה בשפות תכנות כגון: Python ו-R.

הצוות שלנו כולל מומחה לשוק ההון וניהול סיכונים בעל תארים בכלכלה ומימון (BA ו-MBA) עם ניסיון רב הן בפיתוח, יישום ותיקוף מודלים כמותיים.

האקטואר רועי פולניצר, בעל הסמכות מתקדמות בניהול סיכונים פיננסיים (CRM ו-FRM), מייעץ לחברות בניתוחים כמותיים מתקדמים בתחומים של הנדסה פיננסית, יישום מודל מונטה-קרלו, תהליכים סטוכסטיים ופתרון בעיות כמותיות באמצעות שיטות נומריות מתקדמות.

לאקטואר פולניצר שליטה בשפת התכנות וניתוח הנתונים Python, השלטת כיום בעולמות ה-Data, הכוללת את יסודות השפה (מנושאי תחביר פשוטים ועד מודולים ייחודיים לשפה זו), מה שהופך אותו למפתח Python לכל דבר ועניין, ברמה הנדרשת בתעשייה בכלל ובעולמות ה-Data בפרט. בנוסף, האקטואר פולניצר הינו מרצה בקורסים והשתלמויות מקצועיות של לשכת מעריכי השווי והאקטוארים הפיננסיים בישראל (IAVFA) בשפת Python.

