**Main Equation**

The main equation to estimate the implied equity premium (IEP) as is *Equation (10)* is:

=

This equation relates the expected market excess return to risk-neutral moments of market returns.

1. **Source and Derivation of the Main Equation**

Martin’s measure equity premium by focusing primarily on the SVIX index, which is derived from the risk-neutral variance of returns. The key formula is:

*(), or*  =

And from this equation, Tetlock gives a more generalized formula as *Equation (7)*:

=

This equation estimates the equity premium by summing over multiple risk-neutral moments (up to a chosen finite number (K). While Martin’s simply consider the impact from variance as K=2, the use of these higher-order moments in Tetlock’s is intended to capture complex risk factors that influence the expected return, considering more than just the volatility implied by market prices.

The main equation is given the circumstance that K = 4, as *Equation (10)* is a practical approximation of Equation (7), where the summation is limited to a finite number of moments (K) instead of infinity. By limiting K to a specific number (K = 4), the equation focuses on capturing the most important aspects of the market return distribution that have a significant impact on the equity premium, such as variance (volatility), skewness (asymmetry), and kurtosis (fat tails). Therefore, the main equation can be rewrite as:

*= ( + + + )*

**Definition, Calculation, Data and Code of the Main Equation by Components**

The main equation constituted of 4 main components, which are the Equity Premium Estimate (the left side of the equation, regard as ), the Discount Factor (the right side of the equation,), the Risk-neutral Moments Estimate (the right side of the equation, ), and the Risk Adjustment Coefficients (the right side of the equation, regard as weights).

**2. Discount Factor: Risk-free Inverse** ***(******)***

***2.1 Definition***

Denotes the inverse of the risk-free rate at time t. It acts as a discount factor to adjust the present value of future returns or premiums. Here, the SDF MT is simplified to (Rf,t)-1 because the risk-adjustment is handled separately through the sum of weighted risk-neutral moments.

***2.2 Data Acquisition*** *(FRED, Ken French’s website)*

As some equity premium calculations require risk-free rates ranging from monthly to annual horizons and dividend yields, used constant maturity market yields of Treasury bills of 1, 3, 6, and 12 months from the Federal Reserve Economic Database (FRED) to match equity premium horizons. Adopted the average of the one-month and three-month rates as the 60-day rate. Meanwhile, used the one-month risk-free rate (Rf) from Ken French's website before July 31, 2001, when FRED data on the one-month yield became available.

***2.3 Calculation***

With the risk-free rate extracted from the database, we can therefore find the risk-free inverse with the following formula:

=

***2.4 Data Cleaning***

The first data-cleaning method for risk-free inverse is handling missing data. When risk-free rate data is missing for specific dates, the paper recommends using the nearest available data points for interpolation. This approach ensures continuity in the risk-free rate time series.

And the second step is extrapolation for specific maturities. For dates when the maturity structure of risk-free rates does not perfectly align with options maturity, the approach involves extrapolating rates by using trends from nearby dates. Formula example for interpolating the 60-day rate given as:

=

***2.5 Suggested Pseudo Code***

*# Required Libraries*

*install.packages("quantmod") # For data acquisition*

*install.packages("zoo") # For time series manipulation*

*install.packages("dplyr") # For data manipulation*

*library(quantmod)*

*library(zoo)*

*library(dplyr)*

*# Step 1: Acquire Data from FRED*

*# Example: Retrieve 1-month and 3-month Treasury yields*

*getSymbols(c("DGS1MO", "DGS3MO"), src = "FRED")*

*# Step 2: Combine Data into a Data Frame*

*# Convert to a data frame and handle missing values*

*risk\_free\_data <- data.frame(*

*Date = index(DGS1MO),*

*OneMonth = coredata(DGS1MO),*

*ThreeMonth = coredata(DGS3MO)*

*)*

*# Step 3: Interpolate Missing Values for Continuity*

*# Use linear interpolation to fill any missing values*

*risk\_free\_data <- risk\_free\_data %>%*

*mutate(*

*OneMonth = na.approx(OneMonth, maxgap = 5, rule = 2), # Interpolate up to 5-day gaps*

*ThreeMonth = na.approx(ThreeMonth, maxgap = 5, rule = 2)*

*)*

*# Step 4: Calculate the 60-Day Risk-Free Rate (Interpolation)*

*# Interpolate between 1-month and 3-month yields to estimate a 60-day rate*

*risk\_free\_data <- risk\_free\_data %>%*

*mutate(SixtyDay = (OneMonth + ThreeMonth) / 2)*

*# Step 5: Handle Data for Specific Maturities*

*# Extrapolate if necessary, using trends from nearby rates*

*# For example, fill missing 60-day rates using last observed data point*

*risk\_free\_data$SixtyDay <- na.locf(risk\_free\_data$SixtyDay)*

*# Step 6: Calculate the Risk-Free Inverse*

*# Calculate the inverse of the risk-free rate for discounting*

*risk\_free\_data <- risk\_free\_data %>%*

*mutate(*

*SixtyDayRateDecimal = SixtyDay / 100, # Convert percentage to decimal*

*RiskFreeInverse = 1 / (1 + SixtyDayRateDecimal)*

*)*

*# Step 7: Output the Cleaned Data*

*# The cleaned risk-free inverse data is now ready for use*

*head(risk\_free\_data)*

**3. Q-Measure: Risk-neutral Moments Estimate***( for k = 1, 2, 3, 4* ***)***

***3.1 Definition***

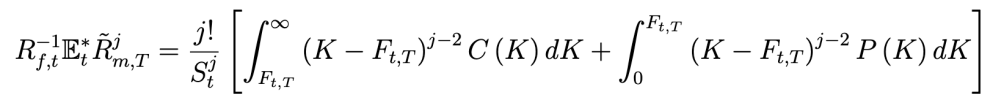
Risk-neutral moments estimate refers to the expectations of higher-order moments of market returns under the risk-neutral probability measure (Q). These estimates are derived from market prices of options and represent how the market prices various dimensions of risk, such as volatility, skewness, and kurtosis.

***3.2 Data Acquisition*** *(OptionMetrics)*

Actively traded options on the S&P 500 index (ticker SPX) from OptionMetrics, only with cash-settled European options with a.m. settlements that do not expire at quarter ends.

***3.3 Calculation for Risk-Neutral Moments ()***

According to general formula of Bakshi and Madan (2000), as specialized in Carr and Madan (2001), the formula shows that risk-neutral expected excess market returns raised to the *j*th power are (*equation 20 in Tetlock’s*):



where j = 2, 3, 4, 5, 6 for relevant moments, St is the market index value, K is the strike price of call and put options priced at C(K) and P (K), respectively, and Ft,T = Rf,tSt is the futures price of the market for maturity T.

***3.4 Calculation for High-Frequency Identification of Expected Variance ()***

As the first step, the realized variance is calculated using high-frequency intra-day data (e.g., minute-by-minute returns). For a given trading day t, the daily realized variance is:

=

where represents the log return during the i-th interval of day t, and Nt is the total number of high-frequency intervals.

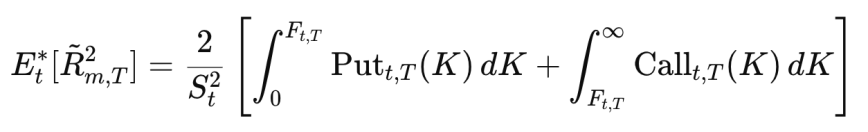
To forecast realized variance over longer horizons, a fractionally integrated (FI) model is used, which accounts for the long-memory property of volatility. A expanded form can be expressed as:

= → =

where L is the lag operator, d is the order of fractional differencing, is a white noise error term and is the gamma function used for fractional differencing. And therefore, the long-term variance can be forecast as:

[] =

For the third step, the risk-neutral variance can be derived from option prices:



where St is the current spot price, Ft,T is the forward price at the t. Putt,T(K) and Callt,T(K) are prices of European put and call options with strike price K. Once both the realized variance and the risk-neutral variance are estimated, the variance premium is calculated as:

*Variance Premium = [] - []*

where [] represents the market's expectation of variance derived from option prices (Q-measure) and [] represents the expected variance based on high-frequency realized variance (P-measure).

***3.5 Data Cleaning***

For the high-frequency calculation of realized variance, using high-frequency data like minute-by-minute or tick data, data cleaning involves handling noise, market microstructure effects, and missing data. Here’s the typical data cleaning process.

|  |  |  |
| --- | --- | --- |
| ***Component*** | ***Data Cleaning Method*** | ***Description*** |
| *Risk-Neutral Moments* | *Options Data Filtering* | *Remove iliquid options, smooth volatility surface, adjust for dividends* |
| *Maturity Alignment* | *Interpolate between maturities to match desired horizons* |
| *Data Adjustment for Dividends* | *Adjust underlying prices for expected dividends* |
| *Smoothing Volatility Surface* | *Use local volatility models for consistent risk- neutral estimates* |
| *High-Frequency Variance* | *Handling Microstructure Noise* | *Remove trade outliers and erroneous trades* |
| *Filtering Illiquid Periods* | *Focus on active trading periods and exclude low volume times* |
| *Synchronizing Data* | *Align timestamps and handle overnight returns* |
| *Data Resampling* | *Resample high-frequency data to 5-minute intervals to reduce noise* |

***3.6 Suggested Pseudo Code***

*# Step 1: Acquire Options Data*

*# Data Source: OptionMetrics or any database providing European-style option prices*

*# Input: Spot price (S\_t), strike prices (K), call prices (C(K)), put prices (P(K)), maturity (T), risk-free rate (r)*

*function get\_options\_data(start\_date, end\_date, T):*

*# Fetch call and put prices across strike prices (K) for maturity T*

*options\_data = fetch\_data(source="OptionMetrics", start\_date=start\_date, end\_date=end\_date, maturity=T)*

*return options\_data*

*# Step 2: Calculate Risk-Neutral Variance*

*# Formula for risk-neutral variance using put and call prices*

*function calculate\_risk\_neutral\_variance(options\_data, S\_t, T):*

*# Extract forward price from spot price and risk-free rate*

*F\_t = S\_t \* exp(r \* T)*

*# Calculate the risk-neutral variance as the integral over put and call prices*

*rn\_variance = (2 / S\_t^2) \* (*

*integrate(options\_data["puts"], from=0, to=F\_t, integrand="put\_price \* dK") +*

*integrate(options\_data["calls"], from=F\_t, to=Inf, integrand="call\_price \* dK"))*

*# Step 3: Calculate Risk-Neutral Skewness*

*# Formula for risk-neutral skewness using put and call prices*

*function calculate\_risk\_neutral\_skewness(options\_data, S\_t, T):*

*# Calculate the risk-neutral skewness using a similar integration process*

*rn\_skewness = (3 / S\_t^3) \* (*

*integrate(options\_data["puts"], from=0, to=F\_t, integrand="put\_price \* (K - F\_t) \* dK") +*

*integrate(options\_data["calls"], from=F\_t, to=Inf, integrand="call\_price \* (K - F\_t) \* dK"))*

*# Step 4: Calculate Risk-Neutral Kurtosis*

*# Formula for risk-neutral kurtosis using put and call prices function calculate\_risk\_neutral\_kurtosis(options\_data, S\_t, T): # Calculate the risk-neutral kurtosis using the following formula*

*rn\_kurtosis = (4 / S\_t^4) \* (*

*integrate(options\_data["puts"], from=0, to=F\_t, integrand="put\_price \* (K - F\_t)^2 \* dK") +*

*integrate(options\_data["calls"], from=F\_t, to=Inf, integrand="call\_price \* (K - F\_t)^2 \* dK") )*

*# Step 5: Combine the Moments into a Data Frame*

*# Output: Data frame with risk-neutral variance, skewness, and kurtosis for further analysis*

*function calculate\_risk\_neutral\_moments(start\_date, end\_date, S\_t, T):*

*# Get options data*

*options\_data = get\_options\_data(start\_date, end\_date, T)*

*# Calculate individual moments*

*rn\_variance = calculate\_risk\_neutral\_variance(options\_data, S\_t, T)*

*rn\_skewness = calculate\_risk\_neutral\_skewness(options\_data, S\_t, T)*

*rn\_kurtosis = calculate\_risk\_neutral\_kurtosis(options\_data, S\_t, T)*

*# Combine moments into a data structure*

*rn\_moments = {*

*"variance": rn\_variance,*

*"skewness": rn\_skewness,*

*"kurtosis": rn\_kurtosis }*

**4. Risk Adjustment Coefficients: Weights Estimate***, , , , )*

***4.1 Definition***

The weights corresponding to each of the risk-neutral moments. The weights indicate the relative importance or influence of each risk-neutral moment on the expected excess return. They are often time-varying and are estimated using statistical models or regressions. The weights are determined based on how sensitive the market's expected returns are to each moment (e.g., variance, skewness).

***4.2 Calculation***

The weights are estimated using recursive regressions of the variance premium on risk-neutral moments. The recursive regression technique involves applying exponentially declining weights to older data, allowing the coefficients to adjust over time and better capture time-varying relationships between the variance premium and risk-neutral moments.

**5. P-Measure: Equity Premium Estimate** ***(******)***

***5.1 Definition***

The equity premium estimate represents the expected market excess return at the time t for a future horizon T. It is the equity premium, which is the expected return of the market over the risk-free rate, as anticipated by investors.

***5.2 Data Acquisition*** *(TAQ Database)*

The paper used exchange-traded fund (ETF) with ticker SPY, SPDR S&P 500 ETF Trust to measure the intraday and daily market returns from 1994 to 2021. Using SPY ETF trades in the Trades and Quotes (TAQ) database, the paper tackled daily realized variance (RV) based on SPY ETF returns over 78 intraday intervals spaced equally in business (i.e., transaction) time throughout regular trading hours: 9:30am to 4:00pm Eastern time.

***5.3 Calculation***

The calculation of equity premium is based on equation 10, as:

=

However, to extract the weights allocation, we need the assumption that the GO portfolio weights on each market-related security are equal across all horizons up to one year, as:

, Ɐ = 30, ..., 360, = 30, ..., 360

After that, as the discount factor, risk-neutral moments estimate and weights have been shown the calculation from section 2 to 4, we can hence take them into equation 10 for equity premium estimate.

***5.4 Suggested Pseudo Code***

*# Step 1: Estimate Time-Varying Weights Using Recursive Regression*

*# Input: Historical variance premium data, risk-neutral moments data*

*function estimate\_weights(start\_date, end\_date, rn\_moments):*

*weights\_data = perform\_recursive\_regression(rn\_moments, variance\_premium\_data)*

*return weights\_data*

*# Step 2: Calculate the Equity Premium Estimate*

*# Formula: Equity Premium = Risk-Free Inverse \* Sum(weights \* risk-neutral moments)*

*function calculate\_equity\_premium(rf\_data, rn\_moments, weights\_data, T):*

*# Initialize a list to store the equity premium estimates*

*equity\_premium\_estimates = []*

*# Loop through each time period to compute the premium*

*for t in time\_periods(start\_date, end\_date):*

*# Get the risk-free inverse for time t*

*rf\_inverse\_t = calculate\_rf\_inverse(rf\_data[t])"]*

*rn\_skewness = rn\_moments[t]["skewness"]*

*rn\_kurtosis = rn\_moments[t]["kurtosis"]*

*# Get the weights for each moment*

*w1 = weights\_data[t]["w1"]*

*w2 = weights\_data[t]["w2"]*

*w3 = weights\_data[t]["w3"]*

*# Calculate the weighted sum of risk-neutral moments*

*weighted\_sum = w1 \* rn\_variance + w2 \* rn\_skewness + w3 \* rn\_kurtosis*

*# Calculate the equity premium for time t*

*equity\_premium\_t = rf\_inverse\_t \* weighted\_sum*

1. **Inspiration for Further Research Directions**

One direction inspired by this paper (Tetlock, 2023) is expanding the framework beyond equity markets to include other asset classes like commodities, currencies, or fixed income markets. Such an approach could reveal differences in how risks are priced across various asset classes, including newer markets like cryptocurrencies. The methodology also suggests potential applications using high-frequency data to analyze intraday patterns of risk premiums, offering insights for high-frequency trading strategies and understanding the dynamics of how information is incorporated into prices throughout the trading day. Additionally, the time-varying weights used in the estimation process indicate that investor risk preferences change over time.