# Estimation of model based Volatility Indexes (VIXs)

**Akhil Behl**
Indira Gandhi Institute of Development Research, Mumbai

**Rohini Grover**
Indira Gandhi Institute of Development Research, Mumbai

**Abstract**

This paper describes the implementation of the model based volatility indexes discussed in Grover and Thomas (2012) in R. The volatility indexes available are: spread adjusted VIX (SVIX), volume adjusted VIX (TVVIX), vega weighted VIX (VVIX), elasticity weighted VIX (EVIX), and the old CBOE VIX (VXO). A novel method for confidence interval estimation of these indexes using bootstrap resampling, introduced in Grover and Shah (2013), is provided along with a first implementation (in R) for calculation of VXO.

*Keywords*: volatility indexes, implied volatility, confidence intervals, bootstrap.

## 1. Introduction

A volatility index (VIX) measures the market's expectation of volatility computed from a chain of option prices at different strikes and maturities. VIX has been shown (Blair, Poon, and Taylor 2001; Jiang and Tian 2005; Corrado and Miller 2005; Giot 2005) to provide superior forecasts compared to historical volatility. Recent work by Grover and Thomas (2012) provides refinements of model based approaches to calculation of VIX by incorporating option liquidity. Grover and Shah (2013) add to the discussion by emphasizing the *uncertainty* in the estimation of VIX. This paper presents an implementation of these model based approaches to point and confidence interval estimation of these methods. We also provide the first implementation in R (R Core Team 2013) for calculation of the old CBOE VIX (VXO) proposed by Whaley (1993).

The paper is organized as follows: Section 2 discusses the steps involved in point estimation of the old CBOE VIX and the model based VIXs. Section 3 describes the procedure employed to estimate confidence bands for the VIXs. Section 4 describes the implementation of the functions developed for point and interval estimation of the VIXs alongwith illustrations. Section 5 concludes.

## 2. Point estimation of VIXs

The methods for calculation of VIX used in these routines are detailed in Whaley (1993);

Grover and Thomas (2012). In this section we provide a brief overview of these methods.

## 2.1. VXO

In 1993, the CBOE introduced the first volatility index. This index is computed from eight near-the-money implied volatility (IV) estimates for the two nearest maturities. The following steps summarise the computation of VXO:

1. Estimation of implied volatilities for a call and put with strike price ($K_t$) immediately below the current index level, S and strike price ($K_u$) immediately above S for the two nearest maturities. These are denoted by:

$$\mathrm{IV}^{K_t}_{c,near},\quad \mathrm{IV}^{K_t}_{p,near},\quad \mathrm{IV}^{K_u}_{c,near},\quad \mathrm{IV}^{K_u}_{p,near},$$
$$\mathrm{IV}^{K_t}_{c,next},\quad \mathrm{IV}^{K_t}_{p,next},\quad \mathrm{IV}^{K_u}_{c,next},\quad \mathrm{IV}^{K_u}_{p,next}.$$

2. Averaging the call and put implied volatilities for each strike and maturity:

$$\mathrm{IV}^{K_t}_{near} = (\mathrm{IV}^{K_t}_{c,near} + \mathrm{IV}^{K_t}_{p,near})/2$$
$$\mathrm{IV}^{K_t}_{next} = (\mathrm{IV}^{K_t}_{c,next} + \mathrm{IV}^{K_t}_{p,next})/2$$
$$\mathrm{IV}^{K_u}_{near} = (\mathrm{IV}^{K_u}_{c,near} + \mathrm{IV}^{K_u}_{p,near})/2$$
$$\mathrm{IV}^{K_u}_{next} = (\mathrm{IV}^{K_u}_{c,next} + \mathrm{IV}^{K_u}_{p,next})/2$$

3. Linear interpolation between nearby implied volatilities and second nearby implied volatilities to create "at-the-money" implied volatilities given by:

$$\mathrm{IV}_i = \mathrm{IV}^{K_t}_i \frac{K_u - S}{K_u - K_t} + \mathrm{IV}^{K_u}_i \frac{S - K_u}{K_u - K_t}$$

where $i = \{near, next\}$ for the two nearest maturities.

4. Trading-day conversion of the interpolated IVs using the formula:

$$\mathrm{IV}_i = \mathrm{IV}_i \frac{\sqrt{N_{c,i}}}{\sqrt{N_{t,i}}}$$

where $N_{c,i}$ and $N_{t,i}$ are number of calender and trading days to expiry for the two nearest maturities, $i = \{near, next\}$.

5. Interpolation of the near and next IVs to create a 22 trading-day implied volatility.

$$\mathrm{VXO} = 100 \times \left[\mathrm{IV}_{near}\left(\frac{N_{t,next} - 22}{N_{t,next} - N_{t,near}}\right) + \mathrm{IV}_{next}\left(\frac{22 - N_{t,near}}{N_{t,next} - N_{t,near}}\right)\right]$$

where $\mathrm{IV}_{near}$ and $\mathrm{IV}_{next}$ are the trading-day implied volatility rates and $N_{near}$ and $N_{next}$ are the number of trading days to expiration for the two nearest maturities. Rollover to the next expiration occurs eight calendar days prior to the expiry of the nearby option.

## 2.2. Model based VIXs

The volatility indexes are computed from all available option prices as follows:

1. Estimation of implied volatilities for a cross-section of options from the two nearest maturities using an option pricing model.

2. Computation of the average implied volatility for each maturity $i$:

$$\mathrm{IV}_i = \frac{\sum_{j=1}^{n} w_{ij}\mathrm{IV}_{ij}}{\sum_{j=1}^{n} w_{ij}}$$

   where $\mathrm{IV}_{ij}$ refers to a vector of implied volatilities for a chain of options $j = \{1 \ldots n\}$ and the two nearest maturities $i = \{near, next\}$, and $w_{ij}$ refers to the weight for the corresponding option $j$. The weights assigned to each option corresponds to one of the following attributes of the options employed: spread, volume, vega, and elasticity.

3. Linear interpolation of the weighted average implied volatilities to compute the 30 day expected volatility. Rollover to the next expiration occurs eight calendar days prior to the expiry of the nearby option.

$$\text{VIX} = 100 \times \left[ IV_{near} \left( \frac{N_{next} - 30}{N_{next} - N_{near}} \right) + IV_{next} \left( \frac{30 - N_{near}}{N_{next} - N_{near}} \right) \right]$$

   where $IV_{near}$ and $IV_{next}$ are weighted average implied volatilities and $N_{near}$ and $N_{next}$ are the number of calendar days to expiration for the two nearest maturities.

*Data filtering*

The computation of a VIX requires vetting of options for the following criteria:

1. Non-positive spreads

2. Zero bid/ask prices.

3. Option prices that violate the no-arbitrage bounds of the option pricing model.

   For example, options violating the lower and upper bounds of the Black and Scholes option pricing model, where the limits are given by:

| | Upper limit | Lower limit |
|---|---|---|
| Call | $S - K \times e^{-rt}$ | $S$ |
| Put | $K \times e^{-rt} - S$ | $K \times e^{-rt}$ |

   where $S$ refers to the underlying price, $K$ refers to the strike price, $r$ refers to the risk-free rate, and $t$ refers to the time to maturity of an option.

   Similarly, for options on currency and futures respectively, one needs to check for violation of Garman & Kohlhagen's, and Black's options pricing models.

# 3. Interval estimation of VIXs

Grover and Shah (2013) introduce the importance of precision of a volatility index and a method to measure it using a confidence interval. Briefly, the confidence band is estimated based on the bootstrapped sampling distribution of a volatility index computed as follows:

1. Sample with replacement from a cross-section of available options at each maturity to construct a bootstrap replicate.
2. Estimate IVs and weighted average IV for each maturity from the corresponding bootstrap replicate.
3. Compute the final VIX estimate by interpolating the weighted average IVs for the two nearest maturities.
4. Steps 1-3 are repeated to approximate the sampling distribution of VIX.
5. The confidence band is computed from this distribution using the adjusted bootstrap percentile method (Efron 1987).

# 4. Implementation and examples

This section describes the functions and example datasets provided in the R package **ifrogs** (Group 2013) for calculation of VXO and the VIXs.

## 4.1. Computing VXO

The function that implements the old CBOE VIX is:

```
vxo(maturity, riskfree, carry, type, strike,
underlying, bid=NULL, ask=NULL, value=NULL)
```

This function requires eight at-the-money options for the two nearest maturities. The eight options are selected in the following fashion: there should be a pair of call and put options for two unique strikes, one directly above and the other directly below the underlying value. The function `vxo` also validates these inputs and stops if they do not match these requirements. The arguments passed are the annualized risk-free rates, annualized carry rates, strike prices, underlying prices, bid/ask prices or option prices. The object returned is a point estimate of VXO.

We demonstrate this by computing VXO for a sample of SPX and NIFTY options. These datasets are provided as `vxo_spx` and `vxo_nifty` respectively. Each dataset is a `data.frame` that includes time to maturity, risk-free rates, type of option ("c" for call,"p" for put), strike prices, underlying prices, bid/ask prices for eight at-the-money options chosen as described above.

*Example 1: Using SPX options*

```
> library(ifrogs)
> data(vxo_spx)
> str(vxo_spx)
```

```
'data.frame':        8 obs. of  7 variables:
 $ maturity  : num  0.0795 0.0795 0.1753 0.1753 0.0795 ...
 $ riskfree  : num  0.0012 0.0012 0.0016 0.0016 0.0012 0.0..
 $ type      : chr  "p" "c" "p" "c" ...
 $ strike    : num  1125 1125 1125 1125 1130 ...
 $ underlying: num  1126 1126 1126 1126 1126 ...
 $ bid       : num  23.2 22 37.1 33.8 24.2 17.4 39.2 30.4
 $ ask       : num  25.9 23.5 42.3 36.3 28.4 21.3 43.9 35.5
```

```
> vxo(maturity=vxo_spx$maturity,
+     riskfree=vxo_spx$riskfree,
+     carry=vxo_spx$riskfree,
+     type=vxo_spx$type,
+     strike=vxo_spx$strike,
+     underlying=vxo_spx$underlying,
+     bid=vxo_spx$bid,
+     ask=vxo_spx$ask)
```

```
[1] 21.92531
```

*Example 2: Using NIFTY options*

```
> data(vxo_nifty)
> str(vxo_nifty)
```

```
'data.frame':        8 obs. of  7 variables:
 $ maturity  : num  0.0795 0.0795 0.1562 0.1562 0.0795 ...
 $ riskfree  : num  0.0629 0.0629 0.0695 0.0695 0.0629 0.0..
 $ type      : chr  "p" "c" "p" "c" ...
 $ strike    : int  5400 5400 5400 5400 5500 5500 5500 5500
 $ underlying: num  5465 5465 5465 5465 5465 ...
 $ bid       : num  68.5 133.7 120 186 72.5 ...
 $ ask       : num  69 134.2 120.7 187 72.6 ...
```

```
> vxo(maturity=vxo_nifty$maturity,
+     riskfree=vxo_nifty$riskfree,
+     carry=vxo_nifty$riskfree,
+     type=vxo_nifty$type,
+     strike=vxo_nifty$strike,
+     underlying=vxo_nifty$underlying,
+     bid=vxo_nifty$bid,
+     ask=vxo_nifty$ask)
```

```
[1] 17.5125
```

## 4.2. Model based VIXs

The point estimation of VIX involves three steps: 1) checking/filtering invalid options at each maturity, 2) computing weighted average implied volatility at each maturity, and 3) interpolating the weighted IVs to compute the final VIX estimate. The VIXs may be computed for the various weighting schemes simultaneously. We demonstrate these three steps and the involved functions in this subsection.

### Preparing the options for a maturity

The function:

```
prep_maturity(maturity, riskfree, carry, type, strike, underlying, schemes,
              bid=NULL, ask=NULL, value=NULL, traded_vol=NULL,
              tv_filter=FALSE, verbose=TRUE)
```

consumes a cross-section of options for a maturity along with a set of weighting schemes.

The inputs comprise of the time to maturity in years, the annualized risk-free rate, the annualized carry rate, the call or put type of option, the strike prices, the underlying prices, the option prices, and the desired schemes.

The optional arguments `bid` and `ask` are used to compute the option spreads for calculation of spread adjusted VIX. All other VIXs can be computed by providing either of the bid/ask prices or the option prices directly via the `value` argument.

Traded volume is an optional argument used for the calculation of the traded volume adjusted VIX, or otherwise, for filtering options with zero traded volume with the optional flag `tv_filter = TRUE`.

The optional flag `verbose=TRUE` may be used to raise messages if any options are dropped.

#### *Carry*

The implied volatilities for an option type are computed using the corresponding option pricing model. This choice can be made using the optional argument `carry`.

Setting `carry = riskfree` gives the Black and Scholes' stock option model, setting `carry = riskfree-div` switches to Merton's stock option model with continuous dividend yield 'div'. Further details on the use of this option are available in the documentation of `GBSVolatility` from **fOptions** (Wuertz, many others, and see the SOURCE file 2013).

#### *Output*

The function returns a `list` of three elements: `maturity`, `schemes`, and `out` which is a `data.frame` of the filtered inputs, implied volatilities, and vega values (based on the schemes). This `list` object is fed to `weighted_iv`.

#### *Illustration*

To illustrate, we prepare data for the S&P 500 index options from the dataset `vix_spx` and the NIFTY index options from the dataset `vix_nifty`.

Each dataset is a list of two data frames corresponding to two nearest maturities. Each list includes: time to maturity (in years), risk-free rate, type of option ("c" for call, "p" for put), strike prices, underlying prices, bid/ask prices, and traded volumes (only for NIFTY) for a cross-section of all available option contracts.

```
> data(vix_spx)
> str(vix_spx)

List of 2
 $ opt_near:'data.frame':       239 obs. of  7 variables:
  ..$ maturity  : num [1:239] 0.0795 0.0795 0.0795 0.0795 ..
  ..$ riskfree  : num [1:239] 0.0012 0.0012 0.0012 0.0012 ..
  ..$ type      : chr [1:239] "p" "p" "p" "p" ...
  ..$ strike    : num [1:239] 675 680 690 700 710 715 720 ..
  ..$ underlying: num [1:239] 1126 1126 1126 1126 1126 ...
  ..$ bid       : num [1:239] 0.05 0.05 0.05 0.1 0.1 0.1 0..
  ..$ ask       : num [1:239] 0.1 0.1 0.1 0.15 0.15 0.15 0..
 $ opt_next:'data.frame':       242 obs. of  7 variables:
  ..$ maturity  : num [1:242] 0.175 0.175 0.175 0.175 0.17..
  ..$ riskfree  : num [1:242] 0.0016 0.0016 0.0016 0.0016 ..
  ..$ type      : chr [1:242] "p" "p" "p" "p" ...
  ..$ strike    : num [1:242] 600 620 625 630 640 650 660 ..
  ..$ underlying: num [1:242] 1126 1126 1126 1126 1126 ...
  ..$ bid       : num [1:242] 0.05 0.1 0.1 0.1 0.15 0.2 0...
  ..$ ask       : num [1:242] 0.2 0.25 0.25 0.35 0.35 0.3 ..

> data(vix_nifty)
> str(vix_nifty)

List of 2
 $ opt_near:'data.frame':       42 obs. of  8 variables:
  ..$ maturity  : num [1:42] 0.0795 0.0795 0.0795 0.0795 0..
  ..$ riskfree  : num [1:42] 0.0629 0.0629 0.0629 0.0629 0..
  ..$ type      : chr [1:42] "p" "p" "p" "c" ...
  ..$ strike    : int [1:42] 4400 4300 4600 6100 4500 6000..
  ..$ underlying: num [1:42] 5465 5465 5465 5465 5465 ...
  ..$ bid       : num [1:42] 2.1 2 2.65 1.2 2.1 1.8 1.65 3..
  ..$ ask       : num [1:42] 2.15 2.1 2.7 1.3 2.15 1.9 1.7..
  ..$ traded_vol: int [1:42] 67000 8400 102200 35550 68300..
 $ opt_next:'data.frame':       28 obs. of  8 variables:
  ..$ maturity  : num [1:28] 0.156 0.156 0.156 0.156 0.156..
  ..$ riskfree  : num [1:28] 0.0695 0.0695 0.0695 0.0695 0..
  ..$ type      : chr [1:28] "c" "c" "c" "p" ...
  ..$ strike    : int [1:28] 6100 6000 5900 4900 5800 5200..
  ..$ underlying: num [1:28] 5465 5465 5465 5465 5465 ...
  ..$ bid       : num [1:28] 3.25 4.55 9.3 29.15 18.9 ...
  ..$ ask       : num [1:28] 3.3 4.6 9.45 29.35 19 ...
  ..$ traded_vol: int [1:28] 86550 211100 481650 462550 83..
```

*Example 1: Vega VIX using SPX options*

```
> spx_near <- prep_maturity(maturity=vix_spx$opt_near$maturity[[1]],
+                           riskfree=vix_spx$opt_near$riskfree[[1]],
+                           carry=vix_spx$opt_near$riskfree[[1]],
+                           type=vix_spx$opt_near$type,
+                           strike=vix_spx$opt_near$strike,
+                           underlying=vix_spx$opt_near$underlying,
+                           schemes="vega",
+                           bid=vix_spx$opt_near$bid,
+                           ask=vix_spx$opt_near$ask,
+                           tv_filter=FALSE)
> str(spx_near)

List of 3
 $ maturity: num 0.0795
 $ schemes : chr "vega"
 $ out     :'data.frame':        186 obs. of  11 variables:
  ..$ maturity  : num [1:186] 0.0795 0.0795 0.0795 0.0795 ..
  ..$ riskfree  : num [1:186] 0.0012 0.0012 0.0012 0.0012 ..
  ..$ carry     : num [1:186] 0.0012 0.0012 0.0012 0.0012 ..
  ..$ type      : Factor w/ 2 levels "c","p": 1 1 1 1 1 1 ..
  ..$ strike    : num [1:186] 965 970 975 980 985 ...
  ..$ underlying: num [1:186] 1126 1126 1126 1126 1126 ...
  ..$ bid       : num [1:186] 158 153 148 143 138 ...
  ..$ ask       : num [1:186] 163 158 153 149 144 ...
  ..$ value     : num [1:186] 161 156 151 146 141 ...
  ..$ iv        : num [1:186] 0.18 0.219 0.212 0.223 0.216..
  ..$ vega      : num [1:186] 1.18 6.36 6.51 10.35 10.61 ...

> spx_next <- prep_maturity(maturity=vix_spx$opt_next$maturity[[1]],
+                           riskfree=vix_spx$opt_next$riskfree[[1]],
+                           carry=vix_spx$opt_next$riskfree[[1]],
+                           type=vix_spx$opt_next$type,
+                           strike=vix_spx$opt_next$strike,
+                           underlying=vix_spx$opt_next$underlying,
+                           schemes="vega",
+                           bid=vix_spx$opt_next$bid,
+                           ask=vix_spx$opt_next$ask,
+                           tv_filter=FALSE)
> str(spx_next)

List of 3
 $ maturity: num 0.175
 $ schemes : chr "vega"
 $ out     :'data.frame':        201 obs. of  11 variables:
```

```
  ..$ maturity  : num [1:201] 0.175 0.175 0.175 0.175 0.17..
  ..$ riskfree  : num [1:201] 0.0016 0.0016 0.0016 0.0016 ..
  ..$ carry     : num [1:201] 0.0016 0.0016 0.0016 0.0016 ..
  ..$ type      : Factor w/ 2 levels "c","p": 1 1 1 1 1 1 ..
  ..$ strike    : num [1:201] 920 925 930 935 940 945 950 ..
  ..$ underlying: num [1:201] 1126 1126 1126 1126 1126 ...
  ..$ bid       : num [1:201] 204 199 194 189 185 ...
  ..$ ask       : num [1:201] 208 204 199 195 190 ...
  ..$ value     : num [1:201] 206 201 197 192 187 ...
  ..$ iv        : num [1:201] 0.203 0.21 0.227 0.235 0.234..
  ..$ vega      : num [1:201] 10.1 14.1 22.6 28.8 31.5 ...
```

*Example 2: Spread, elasticity, and vega VIX using NIFTY options*

The datasets are prepared for near and next month options by filtering out options with zero traded volume.

```
> nifty_near <- prep_maturity(maturity=vix_nifty$opt_near$maturity[[1]],
+                             riskfree=vix_nifty$opt_near$riskfree[[1]],
+                             carry=vix_nifty$opt_near$riskfree[[1]],
+                             type=vix_nifty$opt_near$type,
+                             strike=vix_nifty$opt_near$strike,
+                             underlying=vix_nifty$opt_near$underlying,
+                             schemes=c("spread", "elasticity", "vega"),
+                             bid=vix_nifty$opt_near$bid,
+                             ask=vix_nifty$opt_near$ask,
+                             traded_vol=vix_nifty$opt_near$traded_vol,
+                             tv_filter=TRUE)
> nifty_next <- prep_maturity(maturity=vix_nifty$opt_next$maturity[[1]],
+                             riskfree=vix_nifty$opt_next$riskfree[[1]],
+                             carry=vix_nifty$opt_next$riskfree[[1]],
+                             type=vix_nifty$opt_next$type,
+                             strike=vix_nifty$opt_next$strike,
+                             underlying=vix_nifty$opt_next$underlying,
+                             schemes=c("spread", "elasticity", "vega"),
+                             bid=vix_nifty$opt_next$bid,
+                             ask=vix_nifty$opt_next$ask,
+                             traded_vol=vix_nifty$opt_next$traded_vol,
+                             tv_filter=TRUE)
```

### Computing the weighted implied volatility

The `list` object returned by the `prep_maturity` routine is to be fed to

```
weighted_iv(prepped)
```

which computes the weighted implied volatilities for the contained `schemes`.

This function also returns a `list` containing three elements: `maturity`, `schemes`, and the weighted average implied volatilities.

*Example 1: Vega VIX using SPX options*

```
> spx_near_iv <- weighted_iv(prepped=spx_near)
> spx_next_iv <- weighted_iv(prepped=spx_next)
> spx_near_iv

$maturity
[1] 0.07945205

$schemes
[1] "vega"

$iv
     vega
0.2150128
```

*Example 2: Spread, elasticity, and vega VIX using NIFTY options*

```
> nifty_near_iv <- weighted_iv(prepped=nifty_near)
> nifty_next_iv <- weighted_iv(prepped=nifty_next)
> nifty_near_iv

$maturity
[1] 0.07945205

$schemes
[1] "spread"     "elasticity" "vega"

$iv
    spread elasticity       vega
 0.1675416  0.2511321  0.1779081
```

## Computing point estimates for VIX

The third and final step in the calculation is implemented in the function

```
vix_pt(iv_near, iv_next)
```

which consumes the output of `weighted_iv` and returns the point estimates of the included schemes. Continuing with our illustration:

The point estimates for the VIXs are given by:

```
> spx_vix <- vix_pt(iv_near=spx_near_iv, iv_next=spx_next_iv)
> spx_vix

    vega
21.53219

> nifty_vixes <- vix_pt(iv_near=nifty_near_iv, iv_next=nifty_next_iv)
> nifty_vixes

    spread elasticity       vega
  16.78193   24.85004   17.81540
```

### 4.3. Estimating confidence bands for VIXs

The confidence bounds for the volatility indexes: SVIX, TVVIX, VVIX, and EVIX are implemented through the function:

```
vix_ci(prep_near, prep_next=NULL, n_samples=1e3, conf=0.95,
      verbose=TRUE, ...)
```

The inputs to `vix_ci` are again the prepared data for the two maturities alongwith arguments required for bootstrap sampling. More arguments to `boot` (**boot** Canty and Ripley (2013)) may be passed through '`...`'.

If options are not available for the next maturity, `prep_next=NULL` may be used to compute the confidence interval for weighted average IV for only the near maturity. In this case the weighted average implied volatility is converted to a percentage.

`vix_ci` returns a `list` with three elements: the point estimates, the confidence bands, and the bootstrap replicates of the volatility index(es).

We demonstrate below the computation of confidence bands for the model based VIXs using SPX and NIFTY options for the same datasets and inputs discussed under point estimation of these VIXs.

*Example 1: For vega VIX using SPX options*

```
> spx_ci <- vix_ci(prep_near=spx_near,
+                  prep_next=spx_next,
+                  n_samples=1e3, conf=0.95,
+                  verbose=TRUE)
> str(spx_ci)

List of 3
 $ point  : Named num 21.5
  ..- attr(*, "names")= chr "vega"
 $ ci     : num [1:2, 1] 20.8 22.3
  ..- attr(*, "dimnames")=List of 2
```

```
  .. ..$ : chr [1:2] "lower" "upper"
  .. ..$ : chr "vega"
 $ samples: num [1:1000, 1] 21.1 21.5 21.6 22.1 21.8 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr "vega"
```

*Example 2: For vega VIX using NIFTY options*

```
> nifty_ci <- vix_ci(prep_near=nifty_near,
+                    prep_next=nifty_next,
+                    n_samples=1e3, conf=0.95,
+                    verbose=TRUE)
> str(nifty_ci)
```

```
List of 3
 $ point  : Named num 17.8
  ..- attr(*, "names")= chr "vega"
 $ ci     : num [1:2, 1] 16 19.9
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "lower" "upper"
  .. ..$ : chr "vega"
 $ samples: num [1:1000, 1] 17 19.2 14.9 16.1 17 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr "vega"
```

# 5. Conclusion

This paper presents an implementation of the model based approaches to point and confidence interval estimation of these volatility indexes. We also provide the first implementation in R for calculation of the old CBOE VIX (VXO) proposed by Whaley (1993).

## References

Blair BJ, Poon SH, Taylor SJ (2001). "Forecasting S&P 100 volatility: The incremental information content of implied volatilities and high-frequency index returns." *Journal of Econometrics*, **105**, 5–26.

Canty A, Ripley B (2013). *boot: Bootstrap Functions (originally by Angelo Canty for S)*. R package version 1.3-9.

Corrado CJ, Miller TWJ (2005). "The forecast quality of CBOE implied volatility indexes." *Journal of Futures Markets*, **25**, 339–373.

Efron B (1987). "Better bootstrap confidence intervals." *Journal of the American Statistical Association*, **82**(397), 171–185.

Giot P (2005). "Implied volatility indexes and daily Value-at-Risk models." *Journal of Derivatives*, **12**, 54–64.

Group FR (2013). *ifrogs: Finance routines developed by the IGIDR Finance Research Group.* R package version 0.1-3, URL `http://ifrogs.r-forge.r-project.org`.

Grover R, Shah A (2013). "The imprecision of volatility indexes."

Grover R, Thomas S (2012). "Liquidity considerations in estimating implied volatility." *Journal of Futures Markets*, **32**(8), 714–741.

Jiang GJ, Tian YS (2005). "The model-free implied volatility and its information content." *The Review of Financial Studies*, **18**, 1305–1342.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org/`.

Whaley RE (1993). "Derivatives on market volatility: Hedging tools long overdue." *Journal of Derivatives*, **1**, 71–84.

Wuertz D, many others, see the SOURCE file (2013). *fOptions: Basics of Option Valuation*. R package version 3010.83, URL `http://CRAN.R-project.org/package=fOptions`.

### Affiliation:

Akhil Behl & Rohini Grover
Indira Gandhi Institute of Development Research
Goregaon East, Mumbai
400 065, India
E-mail: `akhilsbehl@gmail.com, grover.rohini@gmail.com`