CERTIFICATE IN QUANTITATIVE FINANCE

Statistical Arbitrage Using Time Series Analysis

by

Koundinya Vajjha

July 2017

Contents

1 CVA calculation for an Interest Rate Swap **1**

1.1 Theory . . . 1

1.1.1 CVA . . . 1

1.1.2 IRS . . . 2

1.2 Calculation of CVA for an IRS . . . 3

1.2.1 Step 1: Estimating probability of default for each tenor . . . 3

1.2.2 Step 2: Extracting future LIBOR rates through HJM model. . . . 6

1.2.3 Step 3: Getting the LIBOR-OIS discounting curve. . . . 8

1.2.4 Step 4: Finding Mark-to-Market position and Exposures of the IRS. 9

1.2.5 Step 5: Repeating the above for many simulations of the forward curve. . . . 9

1.2.6 Step 6: CVA calculation. . . . 10

1.2.7 Step 7: Median and 97.5 Percentile Exposures . . . 11

2 Statistical Arbitrage Using Time Series Analysis **13**

2.1 VAR(p) models: Theory . . . 13

2.1.1 Matrix form estimation of a VAR(p) model. . . . 14

2.2 VAR(p) models: Implementation. . . . 16

2.2.1 Results and Comments . . . 18

2.3 Cointegration Analysis and Estimation : Theory . . . 19

2.3.1 Step 1: Fitting a regression between the levels data. . . . 19

2.3.2 Step 2: Checking stationarity of the residual: ADF test . . . 19

2.3.3 Step 3: Engle-Granger Two Step procedure. . . . 20

2.3.4 Step 4: Fitting an Ornstien-Uhlenbeck process to the spread. . . . 20

2.4 Cointegration Analysis and Estimation : Implementation . . . 21

3 Backtesting **27**

3.1 Introduction: the 'Quantstrat' library . . . 27

3.1.1 Installing Quantstrat . . . 27

3.1.2 Overview of Quantstrat . . . 28

3.1.3 Quantstrat backtesting workflow. . . . 28

3.2 Implementation . . . 29

A About the code and data **33**

Bibliography **35**

Chapter 1

CVA calculation for an Interest Rate Swap

In this chapter we start off with the working and implementation of CVA calculation for an Interest Rate Swap, which is a mandatory addition for the CQF final project. In the following sections, we present theory and outline, in detail, the calculation implemented in the code. The relevant code file names containing the R implementation will be found in the Appendix.

1.1 Theory

1.1.1 CVA

Credit Valuation Adjustment (CVA) is defined to be the adjustment in the values of a risk-free portfolio and the true portfolio value when the possibility of counterparty credit default is taken into account.

$$\text{Risky value} = \text{Risk-free value} - CVA$$

According to [1], "CVA has become a key topic for banks in the recent years due to the volatility of credit spreads and the associated accounting and capital requirements. However, whilst CVA calculations are a major concern for banks, they are also relevant for other financial institutions and corporations that have significant amounts of OTC derivatives to hedge their economic risks."

Most of this section is a slight adaptation of Chapter 14 of [1].

The standard and most general formula for calculating the CVA for an OTC derivative is the following:

$$\text{CVA} = \text{LGD} \int_0^T \text{EE}^*(s) d\text{PD}(s)$$

where

1. *LGD* stands for the loss given default. This is the percentage amount of the exposure expected to be lost if the counterparty defaults. Most often, this quantity is set to be equal to $(1 - RR)$, where RR is the recovery rate.
2. EE^* is the discounted expected exposure for the relevant dates s between now $(s = 0)$ and maturity $(s = T)$. The discounting carried out in computing EE is the risk-free discounting.
3. $dPD(s)$ is the density function of the probability of default of the counterparty.

Thus, the CVA is proportional to the Probability of Default, the Expected Exposure and the Loss Given Default.

1.1.2 IRS

According to [2], an interest rate swap's (IRS's) effective description is a derivative contract, agreed between two counterparties, which specifies the nature of an exchange of payments benchmarked against an interest rate index. The most common IRS is a fixed for floating swap, whereby one party will make payments to the other based on an initially agreed fixed rate of interest, to receive back payments based on a floating interest rate index. Each of these series of payments is termed a 'leg', so a typical IRS has both a fixed and a floating leg. The floating index is commonly an interbank offered rate (IBOR) of specific tenor in the appropriate currency of the IRS, (for example, LIBOR,EURIBOR etc.)

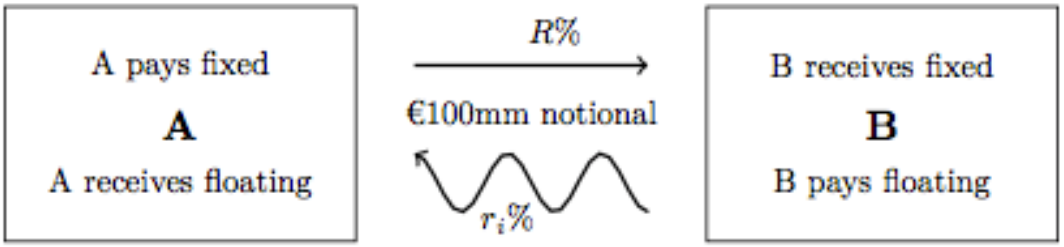


FIGURE 1.1: Graphical depiction of IRS cashflows between two counterparties.

According to its December 2014 statistics release, the Bank for International Settlements reported that interest rate swaps were the largest component of the global OTC

derivative market representing 60% of it, with the notional amount outstanding in OTC interest rate swaps of $381 trillion, and the gross market value of $14 trillion.

1.2 Calculation of CVA for an IRS

We are given an Interest Rate Swap written on the 6M LIBOR over 5Y between a Counterparties A and B. Counterparty B is a potentially risky. In order to set up our problem, we start with a list of data which we are provided with, or assume.

- We assume the following CDS spread values (in basis points) across years 1 to 5 for the counterparty B. Also given are the discount factors for each year. We bootstrap implied probability of default for counterparty B from these CDS spreads and discount factors.

Maturity	CDS spreads B	$D(0;T)$
1Y	50.00	0.97
2Y	77.00	0.94
3Y	94.00	0.92
4Y	125.00	0.86
5Y	133.00	0.81

- In order to obtain a reasonable discounting curve for calculating the expected exposure, we assume a static LIBOR-OIS spread of 80 *bps*.
- Recovery rate is assumed to be 40%.
- Notional is taken to be 1.
- Time period is half-a-year 0.5.
- Fixed-leg rate is taken to be 0.03 or 30 *bps*.

1.2.1 Step 1: Estimating probability of default for each tenor

Here is a list of notation we use for calculation.

1. We are given that the recovery rate is $RR = 4000$ *bps*. Set $L = 1 - RR$.
2. The time points $\{t_n\}_{n=0}^{5}$ are 0,1Y,2Y,3Y,4Y,5Y respectively.
3. Using the formula $\Delta t_n = t_n - t_{n-1}$ for $n = 1, \ldots, 5$, we get the sequence of time differences $\{\Delta t_n\}_{n=1}^{5}$to be 1,1,1,1,1.

4. Set the spreads $\{S_n\}_{n=1}^{5}$ to be 50,77,94,125,133. S_n is the observed spread at time t_n.

5. Denote by P_i the probability of survival from time period t_{i-1} until t_i.

6. $D(0, t_n)$ is the value of the discount factor at time t_n.

7. λ_i is the hazard rate in the period (t_{i-1}, t_i)

Next, we bootstrap the implied probabilities by using the following recursive scheme.

$$
\begin{aligned}
P_1 &= 1 \\
P_2 &= \frac{L}{L + \Delta t_1 S_1} \\
P_n &= \frac{\sum_{i=1}^{n-1} D(0, t_i) \left(L P_{i-1} - P_i (L + \Delta t_i S_n) \right)}{D(0, t_n)(L + S_n \Delta t_n)} + \frac{L P_{n-1}}{(L + S_n \delta t_n)} \qquad n \geq 3
\end{aligned}
$$

From the above we get the hazard rates through the following formula.

$$\lambda_i = \frac{1}{\Delta_i} \log \left(\frac{P_i}{P_{i-1}} \right)$$

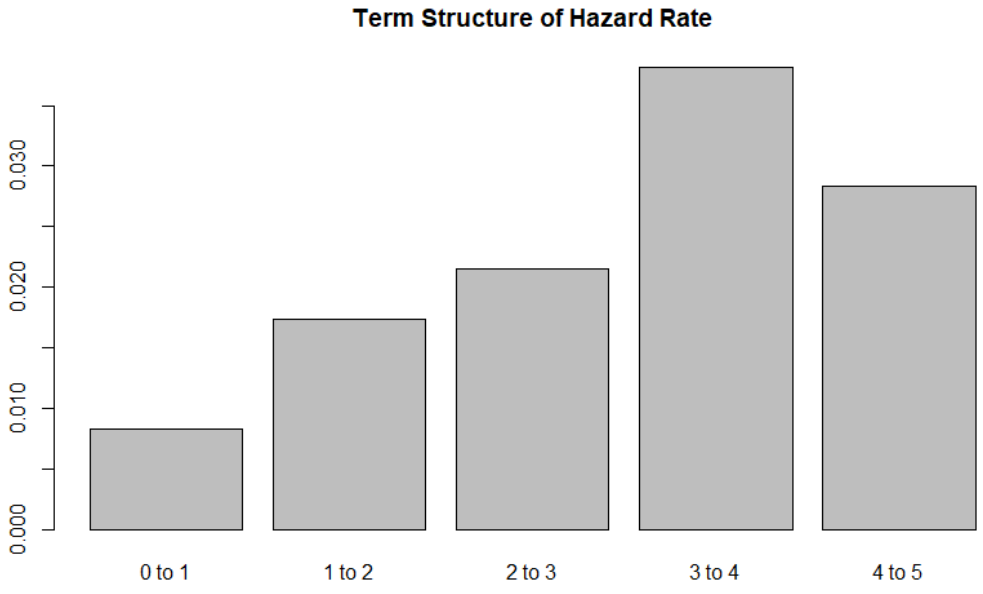


FIGURE 1.2: Term structure of hazard rates implied from CDS spreads.

The above a plot of the term structure of λ across each period. Below is the table of computed values.

t_i	P_i	λ_i	S_i	$D(0,t_i)$
0	1.00000	-	-	-
1Y	0.99174	0.00830	50	0.97000
2Y	0.97462	0.01741	77	0.94000
3Y	0.95389	0.02150	94	0.92000
4Y	0.91821	0.03813	125	0.86000
5Y	0.89259	0.02830	133	0.81000

We interpolate the values of λ_i and obtain a linear interpolation function $\lambda(t)$. Next we compute the Survival Probability function from the following formula.

$$\text{PrSrv}(t) = e^{-\int_0^t \lambda(s)ds}$$

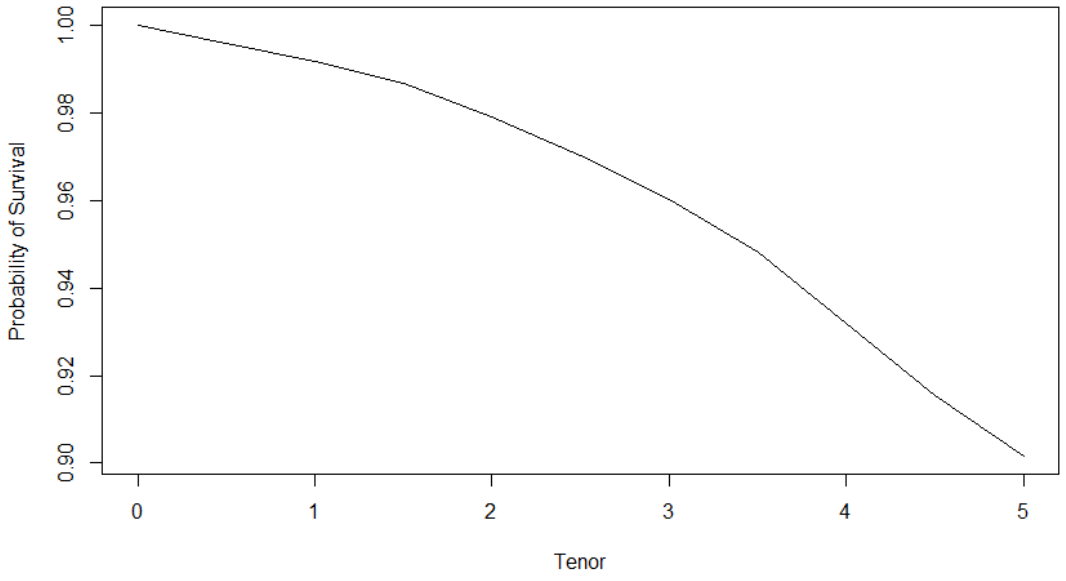


FIGURE 1.3: Survival Probability Function.

Using the above function, we can derive the probability of default for each tenor through the following formula.

$$\text{PD}_i = \text{PrSurv}(t_i) - \text{PrSurv}(t_{i+1}) = P_i - P_{i+1}$$

Tenor	Probability of Default
0	0
1	0.004123658
1.5	0.005230369
2	0.007434312
2.5	0.008980192
3	0.009880491
3.5	0.012238286
4	0.015965548
4.5	0.016475360
5	0.013971912

TABLE 1.1: Probability of Default for each tenor

1.2.2 Step 2: Extracting future LIBOR rates through HJM model.

At time t, the notation $L(t, T_i, T_{i+1})$ stands for the LIBOR rate fixed at time T_i and which matures at time T_{i+1}, which is when the cashflow is paid.

1. **Step 1**: First we define the volatility functions already derived in the *HJM Model MC.xlsm* file provided. Those functions are as follows.

   $v_1(t,\tau) = 0.0064306548$
   $v_2(t,\tau) = -0.0035565431 + -0.0005683999\tau + 0.0001181915\tau^2 + -0.0000035939\tau^3$
   $v_3(t,\tau) = -0.0047506715 + 0.0017541783\tau + -0.0001415249\tau^2 + 0.0000031274\tau^3$

   Here t is the time parameter and τ is the tenor. Next I define the drift function which goes into the SDE. This has the formula:

   $$m(t,\tau) = \sum_{i=1}^{3} v_i(t,\tau) \int_0^\tau v_i(t,s)ds$$

   But for our numerical purposes, we have to integrate numerically each of the three functions v_1,v_2 and v_3 and then sum each of them up to get a numerically tractable version of m.

2. **Step 2**: We work with the Musiela parameterization of the three parameter HJM SDE. This is given by changing the maturity variable T to work with fixed tenors $\tau = t - T$. The final SDE is given by

   $$df(t,\tau) = \left(\sum_{i=1}^{3} v_i(t,\tau) \int_0^\tau v_i(t,s)ds \right) dt + \sum_{i=1}^{3} v_i(t,\tau) dX_i + \frac{\partial F(t,\tau)}{\partial \tau} dt \qquad (1.1)$$

To descretize the above SDE, we denote by $f_i(\tau_j)$ the forward rate for tenor τ_j on date i. Say we have N tenors. We assume that the current days forward rates, $\{f_1(\tau_j\}_{j=1}^{N}$ are known. Now pick an appropriate discretized time scale δt and generate the forward rates for appropriate tenors through the following recurrence relation, which is got from discretizing the above SDE: For j in $1, \ldots, N-1$ and for i in $1, \ldots, M$

$$f_{i+1}(\tau_j) = f_i(\tau_j) + m(t, \tau_j)\delta t + \sum_{k=1}^{3} v_k(t, \tau_j)\phi_k^{(i)}\sqrt{\delta t} + \frac{f_i(\tau_{j+1}) - f_i(\tau_j)}{\tau_{j+1} - \tau_j}\delta t$$

Here $\phi_1^{(i)}, \phi_2^{(i)}, \phi_3^{(i)}$ are standard normal variates chosen for date i. So if δt is chosen to be 1 day, then the above recurrence relation gives the forward rate curves for tenors 1 to $N-1$ for the next M days. For tenor N, we use the same formula but take the *backward* derivative instead of the forward derivative.

3. **Step 3**: Once we have the forward curves for each of the next M days, we can then take the simulated LIBOR rate for a date and a tenor in the future. For example, to get the 6M LIBOR rate a year from now, we choose the value in the 6M tenor column and for the 1 year date in the rows. Figure 1.4 shows simulated forward curves for times 0,6M,1Y,1.5Y,2Y,2.5Y,3Y,3.5Y,4Y,4.5Y and 5Y, which is what we need for the Expected Exposure calculation for the CVA calculation.

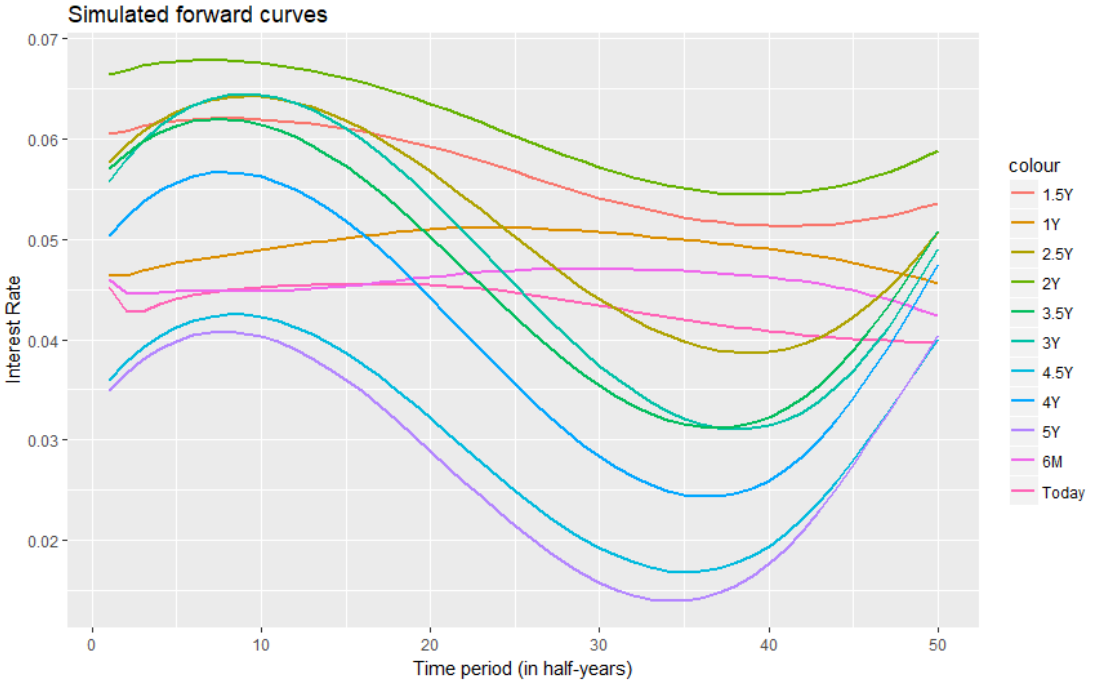


FIGURE 1.4: Simulated forward curves for every 6 months from now till 5 years.

1.2.3 Step 3: Getting the LIBOR-OIS discounting curve.

In fact, for the CVA calculation, we just need the elements in the *diagonal* of the simulated HJM output, i.e., we just need the 6M LIBOR 6 months from now, 1Y LIBOR 1 year from now and so on.

	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0.5	0.0460	0.0447	0.0446	0.0447	0.0448	0.0448	0.0448	0.0448	0.0448	0.0448
1	0.0464	0.0464	0.0468	0.0473	0.0476	0.0479	0.0482	0.0485	0.0487	0.0490
1.5	0.0604	0.0608	0.0613	0.0616	0.0618	0.0620	0.0620	0.0621	0.0620	0.0620
2	0.0664	0.0668	0.0673	0.0676	0.0678	0.0679	0.0679	0.0678	0.0677	0.0676
2.5	0.0577	0.0593	0.0607	0.0618	0.0627	0.0634	0.0638	0.0641	0.0642	0.0642
3	0.0558	0.0579	0.0598	0.0613	0.0625	0.0634	0.0640	0.0644	0.0645	0.0644
3.5	0.0570	0.0585	0.0597	0.0606	0.0613	0.0618	0.0620	0.0620	0.0618	0.0614
4	0.0504	0.0522	0.0536	0.0548	0.0557	0.0563	0.0566	0.0567	0.0566	0.0562
4.5	0.0360	0.0377	0.0392	0.0403	0.0412	0.0419	0.0423	0.0425	0.0425	0.0422
5	0.0349	0.0366	0.0380	0.0391	0.0399	0.0404	0.0407	0.0408	0.0406	0.0403

TABLE 1.2: Simulated HJM output indicating the entries needed for EE calculation.

Once we have this rate f from the simulation, we need to convert it to a simple annualized rate L, by the formula $L = 0.5(e^{2f} - 1)$. The 0.5 is the accrual factor. Now from the above values, we subtract a constant spread of 80 *bps* in order to obtain the OIS curve. Then we find the value of a zero-coupon bond by integrating over these forward rates, and this value will be our discounting factor.

$$\mathrm{DF}_{\mathrm{OIS}} = e^{-\int_0^\tau \mathrm{OIS}_s ds}$$

Tenor	Forward rates	$\mathrm{DF}_{\mathrm{OIS}}$
0.5	0.045995	0.831618
1	0.046397	0.835909
1.5	0.061254	0.806242
2	0.067572	0.811292
2.5	0.062704	0.846009
3	0.063353	0.869107
3.5	0.061969	0.898002
4	0.056691	0.929847
4.5	0.042457	0.966234
5	0.040294	0.983983

TABLE 1.3: Results for this step

1.2.4 Step 4: Finding Mark-to-Market position and Exposures of the IRS.

As mentioned earlier, the fixed leg rate is chosen to be $K = 0.03$. Now the mark-to-market of the Interest Rate Swap at tenor t is defined to be equal to

$$\text{MtM}_t = N\tau \text{DF}_{\text{OIS}}(L(t) - K)$$

Tenor	MtM
0	0.10752
0.5	0.10245
1	0.09644
1.5	0.08403
2	0.06967
2.5	0.05518
3	0.04134
3.5	0.02610
4	0.01350
4.5	0.00665
5	0.00000

TABLE 1.4: Mark-to-Markets of the IRS across tenors.

The exposure is defined as $\text{Exposure}_t = \max(\text{MtM}_t, 0)$.

1.2.5 Step 5: Repeating the above for many simulations of the forward curve.

Note that in the above, we carried out the calculations based on *one* simulation of the forward curve. We now repeat the same calculations for many different simulations of the forward curve through the HJM output. Figure 1.5 is a plot of Exposure values for 100 simulations of the forward curve.

If one takes the mean across each tenor and averages it and plots the average Exposure across tenors, we get the plot in Figure 1.6.

The plot in figure 1.6 shows the expected exposure for the IRS. Taking a slightly higher value of K will result in the familiar "hump" shape for the exposure profile. But we stick to the value of $K = 0.03$.

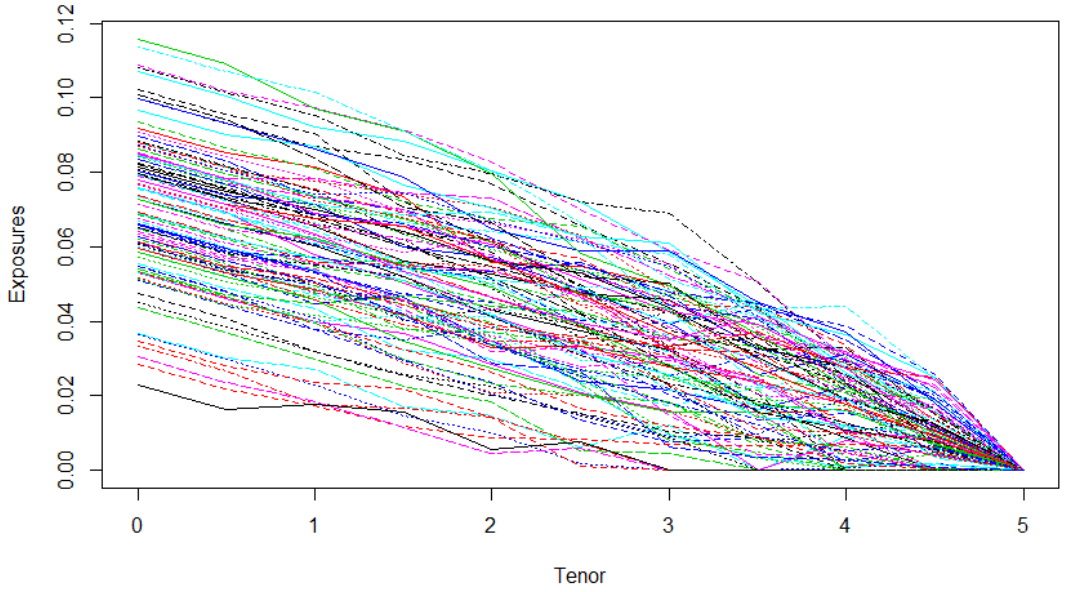


FIGURE 1.5: Exposure profiles for 100 simulations of the forward curve, for $K = 0.03$

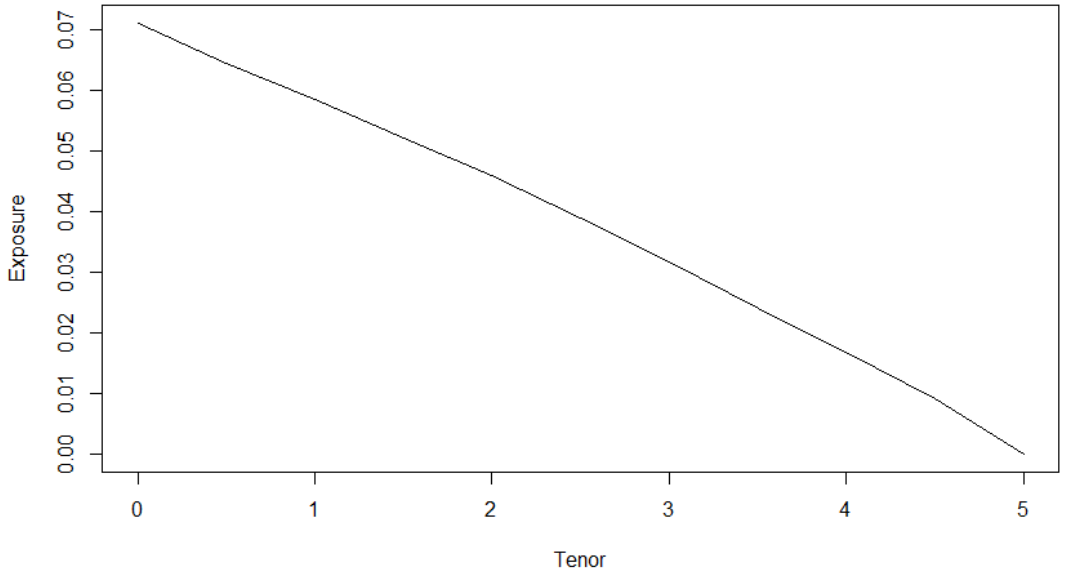


FIGURE 1.6: Average exposure across tenors, for $K = 0.03$

1.2.6 Step 6: CVA calculation.

Finally, once we have the expected exposure from the previous step, we can calculate the Credit Valuation Adjustment for the Interest Rate Swap as follows.

One notes that instead of calculating the exposure at the end points of each tenor, we take the average across the end points of each tenor to obtain the Expected Exposure in the "middle" of each tenor, i.e., at time points 0.25,0.75 and so on up till 4.75. In order to do this, we note the following steps.

For each $t = 0.25, 0.75, \ldots, 4.75$,

- Once we have the expected exposure (EE) from the previous step, we take the rolling mean over 2 periods to obtain the expected exposure in the "middle" of each tenor. Call this EE_t^*
- On the output of DF_{OIS} in Table 1.3, we perform log-linear interpolation and exponentitate to obtain a function $\text{DF}_{\text{OIS}}(t)$.
- Once we have this function, evaluate this function at points 0.25,0.75 and so on up till 4.75. Call this DF_t
- Get the Probability of Default values in each tenor from Table 1.1 and call it PD_t. Finally calculate the CVA as

$$\text{CVA} = \sum_{t=0.25}^{4.75} \text{PD}_t \text{EE}_t^* \text{DF}_t$$

Plugging in our values, we find that the CVA value is approximately 14.146\$ on a notional of 1\$.

1.2.7 Step 7: Median and 97.5 Percentile Exposures

Figures 1.7 and 1.8 show the Median and 97.5 percentile exposure plots for each tenors. One notes that the maximum exposures all occur at tenor 0. This is surely because of the small fixed leg interest rate we have considered. Taking a slightly larger interest rate would have resulted in the fimilar "hump" shape.

Also, if fixed leg interest rates are small, then the exposures and mark-to-market are equal and almost linear.

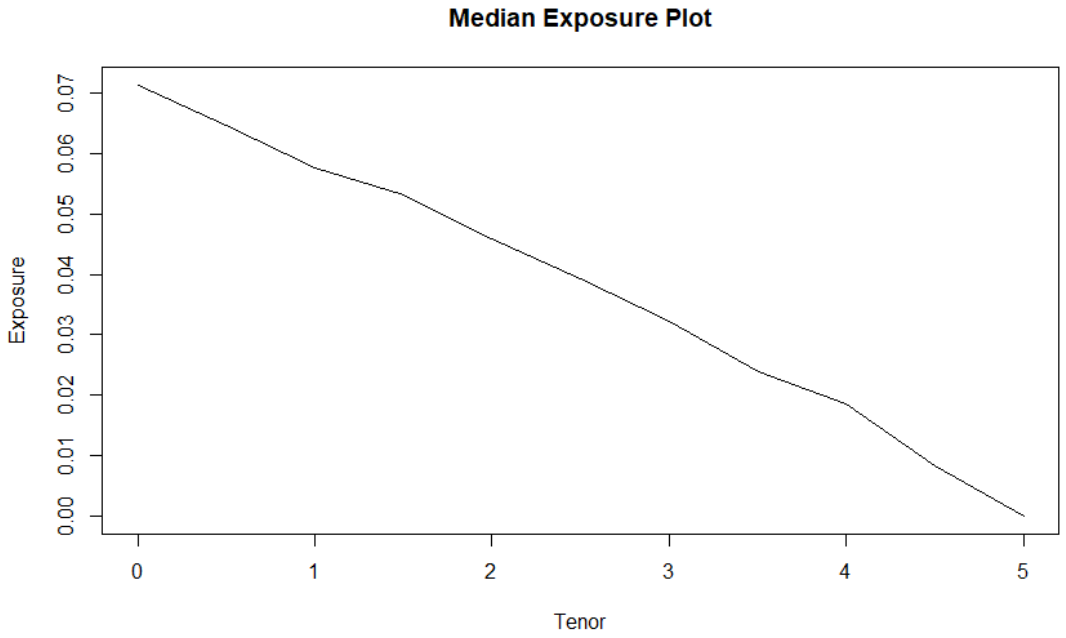


FIGURE 1.7: Median exposure across tenors, for $K = 0.03$

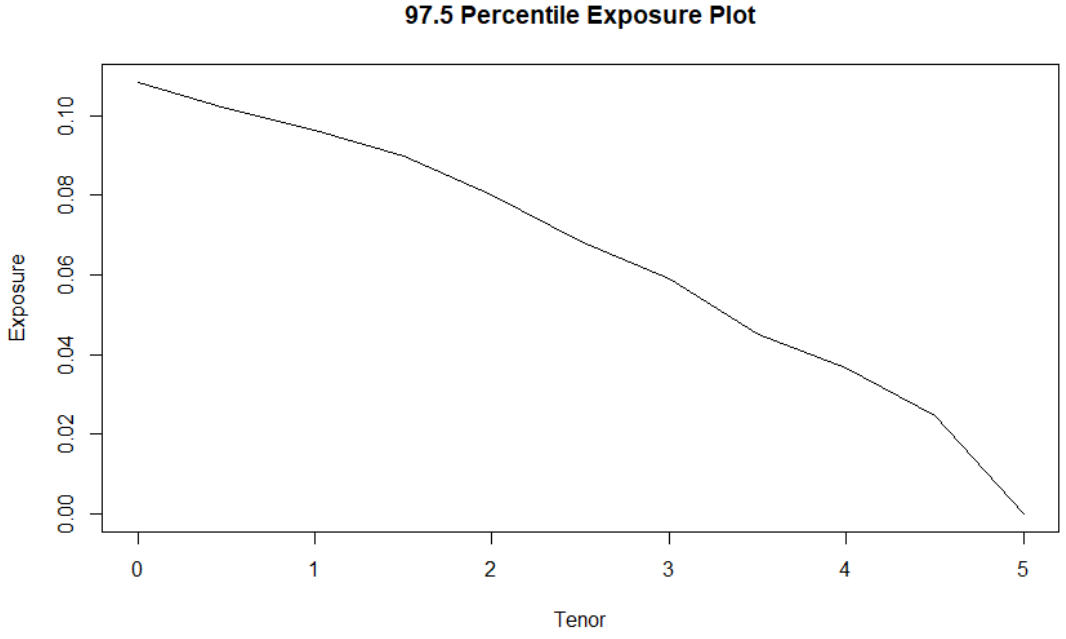


FIGURE 1.8: 97.5 percentile exposure across tenors, for $K = 0.03$

Chapter 2

Statistical Arbitrage Using Time Series Analysis

As part of the final project, it is asked to identify cointegration and causality between two or more time series, with the series considered being both returns and levels data.

In this regard, for the returns data, we implemented a Vector Auto Regression model (VAR) and for the levels data, we implemented Error Correction Models (ECM), more specifically, the Engle-Granger procedure and other stationarity tests.

This chapter is organized as follows: Section 2.1 concerns itself with the theory and implementation of a VAR(p) model for a basket of returns data. Section 2.2 concerns itself with the theory and implementation of VECM models to parameterize cointegration for levels data. The final section contains the implementation and backtesting of a pairs trading strategy which utilizes cointegrated pairs.

2.1 VAR(p) models: Theory

This section will closely follow [3]. Say we are given a univariate time series $\{y_t\}$, whose forecasts are what we are intersted in. It makes sense to begin forecasts that are linear functions of a number p, of past observations.

$$\hat{y_{T+1}} = \nu + \alpha_1 y_T + \alpha_2 y_{T+2} + \cdots + \alpha_p y_{T-p+1}$$

Now since the true value y_{T+1} is generally not equal to the value $\hat{y}_{T+1}$, there is a forecast error u_{T+1}. So our equation can be written as

$$\hat{y_{T+1}} = \nu + \alpha_1 y_T + \alpha_2 y_{T+2} + \cdots + \alpha_p y_{T-p+1} + u_{T+1}$$

Now assuming our numbers are realizations of random variables and that the same data generation law prevails in each time period T, the above has the form of an *autoregressive* process.

$$y_{t+1} = \nu + \alpha_1 y_t + \alpha_2 y_{t+2} + \cdots + \alpha_p y_{t-p+1} + u_t$$

where the quantitites y and u are now random variables. In order to get a true autoregressive process, we assume that the forecast errors u_t and u_s ins different time periods are uncorrelated.

Now, for multiple time series, (say k in number) we have the easy generalization of having the same autoregressive relationship hold for each time series separately. Writing it in vector notation,

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + u_t$$

where the l-periods back observation y_{t-l} is called the l-th lag of y, c is a $k \times 1$ vector of constants, A_i is a time-invariant $k \times k$ matrix and u_t is a $k \times 1$ vector of error terms called the K dimensional white noise process, which satisfies $\mathsf{E}[u_t] = 0$, $\mathsf{E}[u_t u_t^{'}] = \Sigma_u$, a non-singular matrix independent of time and $\mathsf{E}[u_t u_s^{'}] = 0$ for $t \neq s$. This is the VAR(p) process.

An important condition for a VAR(p) model is it's *stability*. A VAR model is stable if the *reverse characteristic polynomial* has no roots on and inside the unit circle. This is equivalent to saying that no eigenvalue of each matrix A_p is of modulus greater than 1.

2.1.1 Matrix form estimation of a VAR(p) model.

It is possible to express a VAR(p) model in concise matrix form so that the matrices can be estimated in one go. This section outlines this matrix form.

Given the VAR(p) model with k endogenous variables,

$$\mathbf{y}_{t+1} = \mathbf{c} + \alpha_1 \mathbf{y}_{t-1} + \alpha_2 \mathbf{y}_{t-2} + \cdots + \alpha_p \mathbf{y}_{t-p+1} + \mathbf{e}_t$$

we can expand it to look like,

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,k} \\ a^1_{2,1} & a^1_{2,2} & \cdots & a^1_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{k,1} & a^1_{k,2} & \cdots & a^1_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,k} \\ a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{k,1} & a^p_{k,2} & \cdots & a^p_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix}$$

Now one can rewrite this in a general way which includes $T+1$ observations y_0 to y_T.

$$\mathbf{Y} = \mathbf{BZ} + \mathbf{U}$$

where

$$\mathbf{Y} = \begin{bmatrix} y_p & y_{p+1} & \cdots & y_T \end{bmatrix} = \begin{bmatrix} y_{1,p} & y_{1,p+1} & \cdots & y_{1,T} \\ y_{2,p} & y_{2,p+1} & \cdots & y_{2,T} \\ \vdots & \vdots & \vdots & \vdots \\ y_{k,p} & y_{k,p+1} & \cdots & y_{k,T} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} c & A_1 & A_2 & \cdots & A_p \end{bmatrix} = \begin{bmatrix} c_1 & a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,k} & \cdots & a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,k} \\ c_2 & a^1_{2,1} & a^1_{2,2} & \cdots & a^1_{2,k} & \cdots & a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ c_k & a^1_{k,1} & a^1_{k,2} & \cdots & a^1_{k,k} & \cdots & a^p_{k,1} & a^p_{k,2} & \cdots & a^p_{k,k} \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_{p-1} & y_p & \cdots & y_{T-1} \\ y_{p-2} & y_{p-1} & \cdots & y_{T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_0 & y_1 & \cdots & y_{T-p} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_{1,p-1} & y_{1,p} & \cdots & y_{1,T-1} \\ y_{2,p-1} & y_{2,p} & \cdots & y_{2,T-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,p-1} & y_{k,p} & \cdots & y_{k,T-1} \\ y_{1,p-2} & y_{1,p-1} & \cdots & y_{1,T-2} \\ y_{2,p-2} & y_{2,p-1} & \cdots & y_{2,T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,p-2} & y_{k,p-1} & \cdots & y_{k,T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1,0} & y_{1,1} & \cdots & y_{1,T-p} \\ y_{2,0} & y_{2,1} & \cdots & y_{2,T-p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,0} & y_{k,1} & \cdots & y_{k,T-p} \end{bmatrix}$$

and

$$U = \begin{bmatrix} e_p & e_{p+1} & \cdots & e_T \end{bmatrix} = \begin{bmatrix} e_{1,p} & e_{1,p+1} & \cdots & e_{1,T} \\ e_{2,p} & e_{2,p+1} & \cdots & e_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ e_{k,p} & e_{k,p+1} & \cdots & e_{k,T} \end{bmatrix}$$

From the above, we can derive the following:

- An OLS estimate for the matrix $\mathbf{B}$.

$$\hat{\mathbf{B}} = \mathbf{Y}\mathbf{Z}' \left(\mathbf{Z}\mathbf{Z}'\right)^{-1}$$

- The Regression Residuals

$$\hat{\epsilon} = \mathbf{Y} - \hat{\mathbf{B}}\mathbf{Z}$$

- Estimator of the residual covariance matrix

$$\hat{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} \hat{\epsilon}_t \hat{\epsilon}_t'$$

- Covariance matrix of regression coefficients. Here $\otimes$ refers to the Kronecker Product, and *Vec* denotes vectorization.

$$\widehat{\mathrm{Cov}}(\mathrm{Vec}(\hat{\mathbf{B}})) = (\mathbf{Z}\mathbf{Z}')^{-1} \otimes \hat{\Sigma}$$

So we have outlined enough theory to code up our own version of the concise matrix regression estimation of the VAR(p) model. The relevant code file names containing the R implementation will be found in the Appendix.

2.2 VAR(p) models: Implementation.

For the implementation part, we chose a basket of 5 stocks traded on the National Stock Exchange (NSE) of India. According to a report made by the Times of India, these four stocks were the largest gainers in the past two months i.e., from May till July of 2017. We decided to download 10 minute candle data from a broker in India called Zerodha. Zerodha offers it's clients an API whereby it is possible to download historical data of options,futures and common equity traded across the Bombay Stock Exchange, the National Stock Exchange and the Mercentile Exchange of India.

As a test case, we decided to implement a VAR(p) model on the five stocks with ticker names CIPLA, ITC, DRREDDY, TATASTEEL and RELIANCE. The data frequency was 10 minutes and the data was for two months in duration. Figures 2.1 and 2.2 show the time series and returns for each of the above series.

Matrix form regression estimation and optimal lag selection.

In order to fit a VAR(p) model, we have to estimate the value of p, which is the optimal lag to consider. common method to do this is to consider various lags for p and select

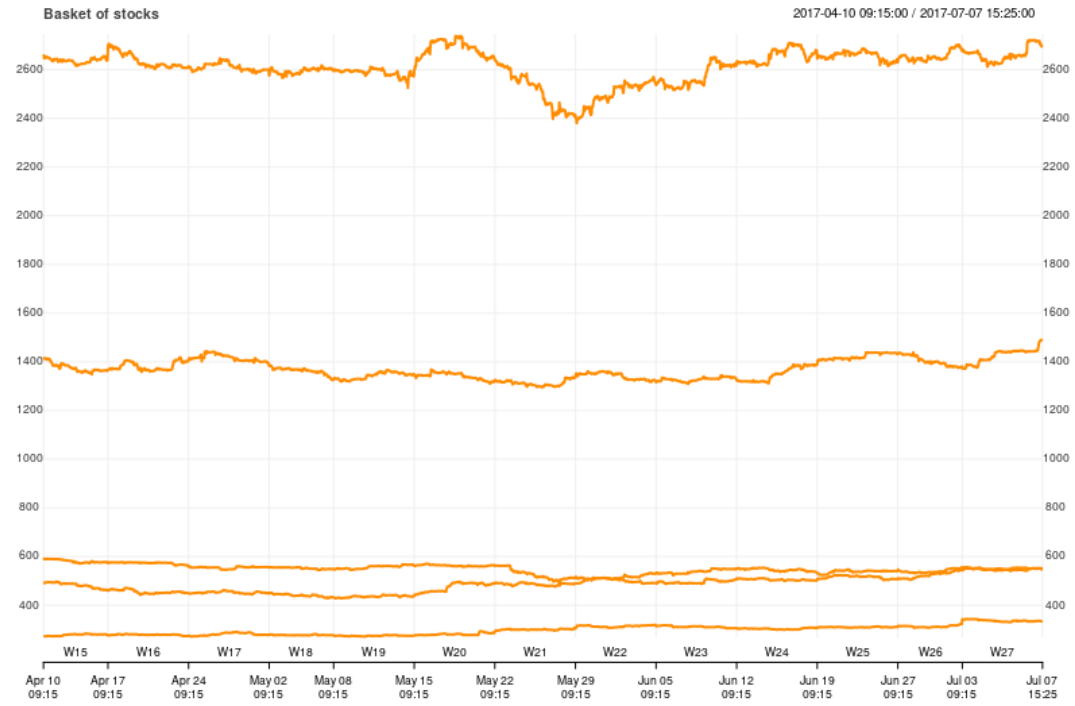


FIGURE 2.1: Plot of RELIANCE, DRREDDY, CIPLA, TATASTEEL and ITC (from top to bottom)

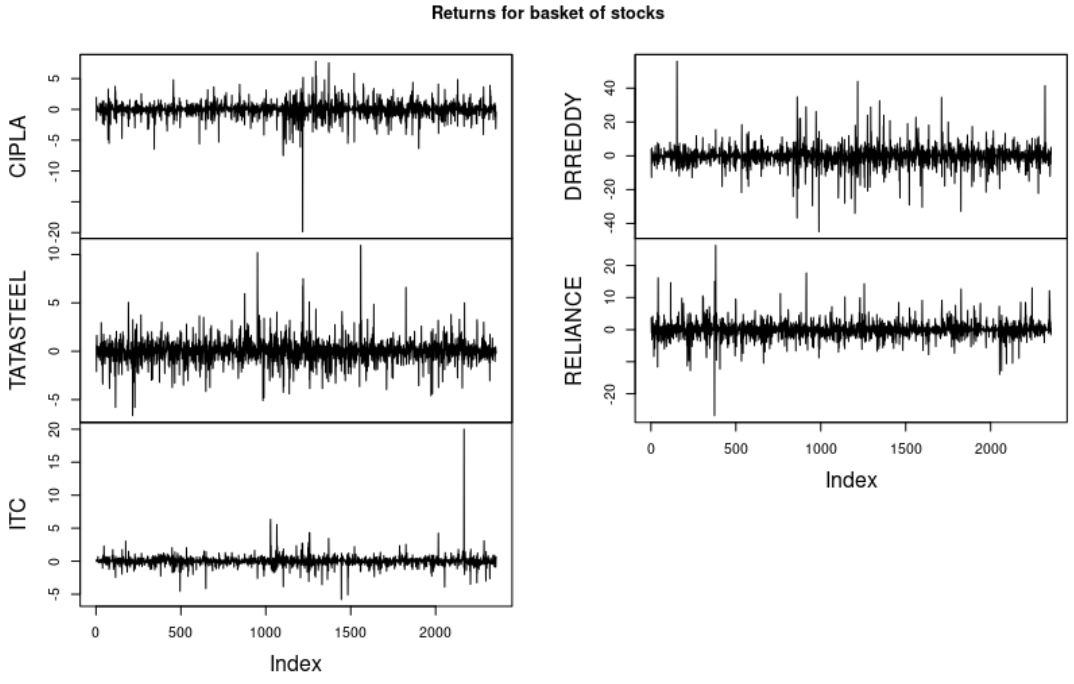


FIGURE 2.2: Plot of returns of RELIANCE, DRREDDY, CIPLA, TATASTEEL and ITC

the value of p which minimizes a particular function called the "Information Criterion". There are many information criteria available, and which one to consider isn't quite

clear. For this exercise, we chose that value of p which minimizes the Akaike Information Criterion, which is given by the following:

$$\text{AIC}(p) = \log|\hat{\Sigma}| + \frac{n(np+1)}{T}$$

where n is the total number of stocks in the basket and T is the total number of observations.

2.2.1 Results and Comments

We use the 'vars' package in R to minimize the AIC and we get the optimal p to be 3. The same package also shows us that the fitted VAR model is *stable* because all the roots of the characteristic polynomial have absolute value less than 1.

The following is the output for the matrix form regression which we have coded from scratch (function 'myVAR()'). The actual estimates for the beta coefficients can also be accessed from the function. I have not printed them here.

	Cipla	Tata Steel	ITC	Reddy	Reliance
Cipla	$5.627x10^{-06}$	$2.244x10^{-07}$	$-1.638x10^{-07}$	$8.635x10^{-07}$	$1.208x10^{-07}$
Tata Steel	$2.244x10^{-07}$	$7.372x10^{-06}$	$1.102x10^{-06}$	$8.109x10^{-07}$	$1.023x10^{-06}$
ITC	$-1.638x10^{-07}$	$1.102x10^{-06}$	$7.572x10^{-06}$	$-2.076x10^{-07}$	$3.967x10^{-07}$
Reddy	$8.635x10^{-07}$	$8.109x10^{-07}$	$-2.076x10^{-07}$	$4.555x10^{-06}$	$3.173x10^{-07}$
Reliance	$1.208x10^{-07}$	$1.023x10^{-06}$	$3.967x10^{-07}$	$3.173x10^{-07}$	$3.845x10^{-06}$

TABLE 2.1: Residual covariance matrix for the VAR(3) model

From Table 2.1 we can derive the Correlation Matrix of Residuals (dividing each column by the leading covariance).

	Cipla	Tata.Steel	ITC	Reddy	Reliance
Cipla	1.00	0.03	-0.03	0.17	0.03
Tata.Steel	0.03	1.00	0.15	0.14	0.19
ITC	-0.03	0.15	1.00	-0.04	0.07
Reddy	0.17	0.14	-0.04	1.00	0.08
Reliance	0.03	0.19	0.07	0.08	1.00

TABLE 2.2: Residual correlation matrix for the VAR(3) model

From the above table we can see that there is little, if any, correlation between the returns of the basket of stocks. As this project concerns itself with cointegration, this basket of stocks would serve as a bad example for a cointegrated relationship. However, one can see that the above analysis shows that this basket can be a good portfolio for diversification. The simple VAR model can be recalibrated again at timely intervals to check if there is any more correlation between the stocks as time progresses.

Such a basket of stocks can be a good source of income in non-turbulent times in the market. However, as is commonly known, during a crash, all equities tend to move together and there is high correlation. So as long as one has a hedge against sudden movements, this basket of stocks can be thought of as a good investment. So this would suggested that the VAR model is a good tool at identifying baskets of stocks which are uncorrelated and thus shows possibility for diversification.

Since matters of correlation are subtle, this idea definitely warrants more serious and rigorous study, which is unfortunately outside the scope of the current project.

2.3 Cointegration Analysis and Estimation : Theory

A system $\mathbf{Y}_t$ of k time series is called *cointegrated* if there exists a vector β_{Coint} of weights such that the process $\beta'_{\text{Coint}}\mathbf{Y}_t = e_t$ is integrated of order 0, i.e., stationary. In this section we outline the theory of how to estimate cointegration in time series.

To keep our analysis simple, we focus on the case $k = 2$, that is, estimating cointegration in two time series, although it is possible to extend this analysis to more than two time series as well.

2.3.1 Step 1: Fitting a regression between the levels data.

Given two time series $\{x_t\}$ and $\{y_t\}$, the first step is to fit a regression between y_t and x_t. We then obtain a coefficient $\hat{\beta}$ of the regression. Set e_t to be the residuals

$$e_t = y_t - \hat{\beta}_1 x_t - \hat{\beta}_0 = \beta'_{\text{Coint}}\mathbf{Y}_t - \mu_e$$

where $\beta_{\text{Coint}} = (1, -\hat{\beta}_1)'$ and $\mathbf{Y}_t = (y_t, x_t)$ and $\mu_e = \hat{\beta}_0$

The idea is to check if the introduction of the coefficient $\hat{\beta}$ results in the *elimination* of a common stochastic process between x_t and y_t so the resulting residual will be stationary.

2.3.2 Step 2: Checking stationarity of the residual: ADF test

To check if the residual series is stationary, we can perform the Augmented Dickey-Fuller (ADF) test on it. The ADF test is applied to the model

$$\Delta e_t = \gamma e_{t-1} + \delta_1 \Delta e_{t-1} + \cdots + \delta_{p-1}\Delta e_{t-p+1} + \varepsilon_t$$

Note that there are three main versions of the test, the other versions apart from the one described above offer to add a constant "drift" and "trend". However we don't focus on these. In the model described above, the ADF test tests the null hypothesis that $\gamma = 0$ contrast to the alternate hypothesis of $\gamma < 0$.

The test statistic for the ADF test $\mathrm{DF}_\tau = \frac{\hat{\gamma}}{SE(\hat{\gamma})}$ is compared to a critical value corresponding to the confidence level. For a confidence level of 95%, the critical value is -1.95. The ADF test also requires an estimation of the number of lags to be used, p. To do this, as before, we estimate an optimal p based on the Akaike Information Criterion.

2.3.3 Step 3: Engle-Granger Two Step procedure.

The first step of the Engle-Granger procedure to test for cointegration was the last two steps. Once we have a stationary residual for the regression we have performed, the next step is to plug the residual into the ECM equation:

$$\Delta y_t = \beta_1 \Delta x_t - (1 - \alpha) e_{t-1}$$

And estimate the coefficients β_1 and $(1 - \alpha)$ through another regression. It is then required to confirm the significance of $(1 - \alpha)$. If this is not significant, check the same for the regression

$$\Delta x_t = \beta_1 \Delta y_t - (1 - \alpha) e_{t-1}$$

This step is required to identify the leading variable.

These two steps confirm the cointegration in a pair of time series.

2.3.4 Step 4: Fitting an Ornstien-Uhlenbeck process to the spread.

Once the residuals pass the Engle-Granger procedure we have evidence of a cointegrating relationship between the stocks, and so the residuals are mean-reverting. This spread can now be used to trade. However, there remains the question of entry and exit.

In order to find out the optimal entry and exit points, we fit an Ornstien-Uhlenbeck process to the spread and infer the entry and exit points. The solution for the OU SDE has an autoregressive term:

$$e_{t+\tau} = (1 - e^{-\theta\tau})\mu_e + e^{-\theta\tau} e_t + \varepsilon_{t,\tau}$$

Where τ is observation frequency. To estimate the coefficients, we fit an $AR(1)$ process to it via Ordinary Least Squares:

$$e_t = C + Be_{t-1} + \varepsilon_t$$

From this we get, by comparing the above two equations, that

$$\theta = -\frac{\log(B)}{\tau}$$

and

$$\mu_e = \frac{C}{1-B}$$

The scatter of the OU process is given by

$$\sigma_{OU} = \sqrt{\frac{2\theta}{1-e^{-2\theta\tau}}\mathsf{V}(e_{t,\tau})}$$

To plot the trading bounds, we use

$$\sigma_{eq} \approx \frac{\sigma_{OU}}{\sqrt{2\theta}}$$

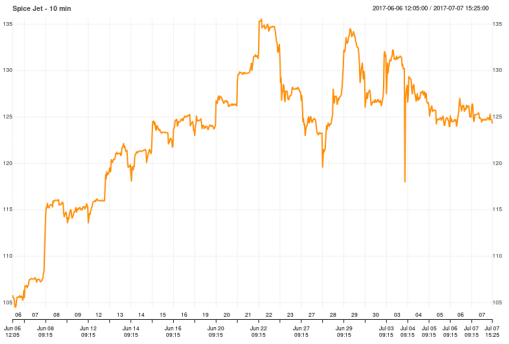
For potential entry/exit signals, we use $\mu_e \pm \sigma_{eq}$. For getting an idea of half-life, or the speed of mean reversion, we take the following: $\hat{\tau} = \frac{\log(2)}{\theta}$

2.4 Cointegration Analysis and Estimation : Implementation

In order to implement and test for cointegration between a pair of time series, we chose two stocks of two companies Spice Jet and Indigo airlines, which are the two largest private airlines in India.

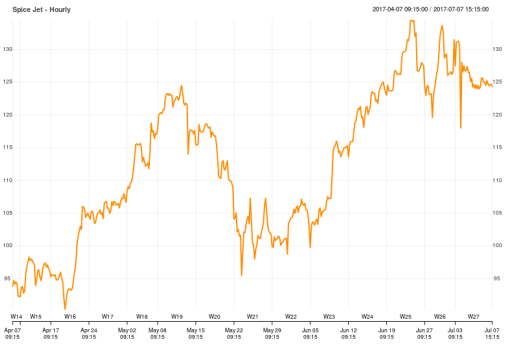
Since a majority of the market share in the airlines business is held by one of these two companies, it is reasonable to assume that their stock prices would be cointegrated. In order to test this asssumption, we set out to implement the Engle-Granger procedure. The idea was if there is indeed evidence of cointegration, then we could set up a trading strategy which trades the mean-reverting spread. We consider time-series data of INDIGO and SPICEJET at frequencies of daily, hourly and 10 minutes. Figure 2.3 shows the stock prices at different frequencies.

(A) Indigo 10 minute data

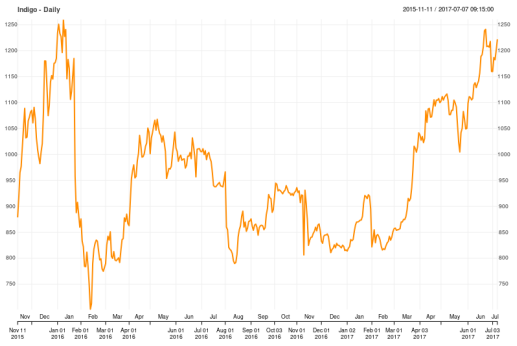


(B) Spice Jet 10 minute data

(C) Indigo hourly data



(D) Spice Jet hourly data



(E) Indigo daily data

(F) Spice Jet daily data

FIGURE 2.3: Comparison of Indigo and Spice Jet stock prices over different frequencies.

The data for both series shows pretty noticeable cointegration in that both series tend to move in the same direction across long periods of time. It is an interesting question to ask if cointegration becomes more pronounced or less pronounced as frequency of measurement increases.

- For the time series data we proceed to step 1 and fit a regression between the two price series. Table 2.5 shows the results. Note that the coefficients are more or less similar in each case.

  So we can conclude, to a reasonable degree, that the frequency of observation does not matter in the regression as all three give more or less the same regression

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	667.4543	6.9245	96.39	0.0000
SPICEJET	4.1136	0.0609	67.59	0.0000

TABLE 2.3: 10 minute data

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	658.0440	15.6271	42.11	0.0000
spice.zoo[, 3]	4.1820	0.1384	30.23	0.0000

TABLE 2.4: 60 min data

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	643.8760	18.6062	34.61	0.0000
spice.zoo[, 3]	4.2857	0.2462	17.41	0.0000

TABLE 2.5: Daily data

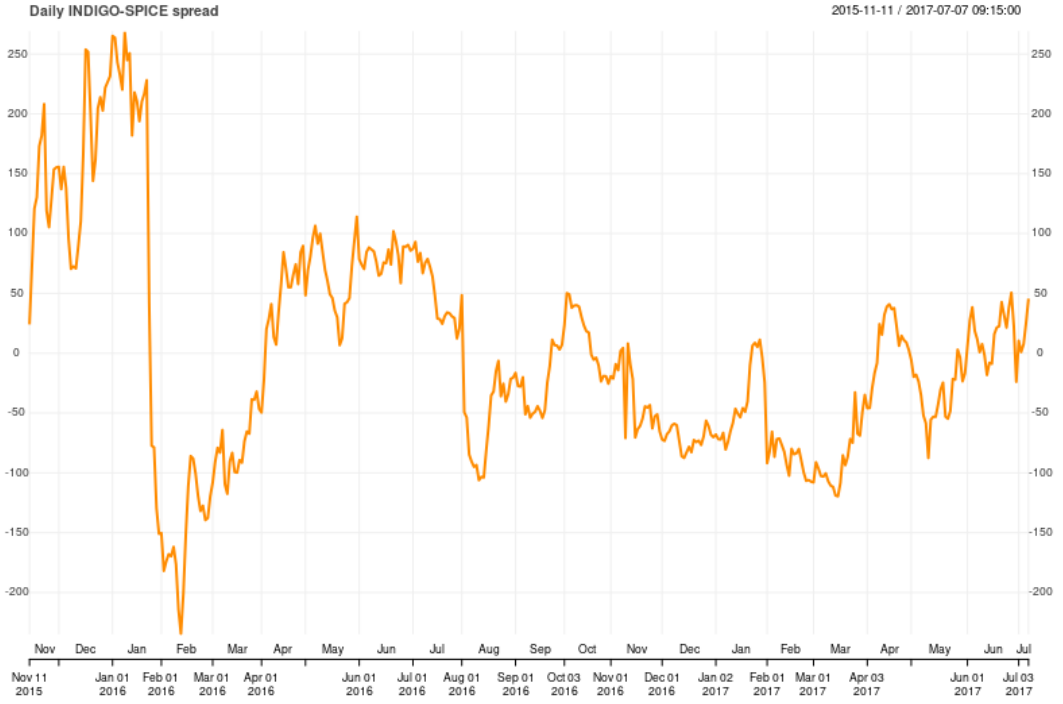


FIGURE 2.4: Plot of daily spread between INDIGO-SPICEJET

coefficients and hence the same spread. We work with daily data from now onwards.The residuals of the regression on daily data is plotted in Figure 2.4. We call this the "spread".

- Now that we have the spread, our next task is to check if the spread is stationary. For this, we look at the Augmented Dickey-Fuller test. Before performing the test, we compute the optimal lag value through the AIC criterion via the "`ur.df`" function in R. The optimal lag value was found to be 1.

Our recoded ADF test function is implemented in the function '`myADF()`'. Running the '`myADF`' function on the residuals of the previous regression estimates that the coefficient of e_{t-1} divided by the standard error of estimation is -2.90181, which is nothing but the ADF test statistic. This value is consistent with the output from the '`ur.df`' function as well. The latter is printed below.

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################
Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max
-179.206   -8.491    0.098    9.756   88.444

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
z.lag.1    -0.03453    0.01188  -2.907  0.00384 **
z.diff.lag  0.15810    0.04876   3.242  0.00128 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 22.05 on 406 degrees of freedom
Multiple R-squared:  0.04001,Adjusted R-squared:  0.03528
F-statistic:  8.46 on 2 and 406 DF,  p-value: 0.0002513



Value of test-statistic is: -2.9074

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Note that -2.9 is smaller than -2.58 so we reject the null hypothesis that $\gamma = 0$. So there is no unit root so the series is stationary.

- Now that we've confirmed that the residuals are stationary, we move to step 2 of the Engle-Granger procedure. So we perform a regression:

$$\Delta y_t = \beta_1 \Delta x_t - (1 - \alpha) e_{t-1}$$

Where y_t is the price series for INDIGO, x_t is the price series for SPICEJET and e_t is the residual time series for the regression fitted in the previous steps.

Here is the output from the R code:

```
Call:
lm(formula = DeltaY ~ DeltaX + e)

Residuals:
     Min       1Q   Median       3Q      Max
-184.306   -8.616    0.913   10.602   79.815

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01193    1.11258   0.011   0.9915
DeltaX       4.51672    0.39375  11.471   <2e-16 ***
e            0.02947    0.01202   2.451   0.0147 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 22.45 on 406 degrees of freedom
Multiple R-squared:  0.2587,Adjusted R-squared:  0.2551
F-statistic: 70.86 on 2 and 406 DF,  p-value: < 2.2e-16
```

So we see that the coefficient $(1 - \alpha)$ is significant at the 95% confidence level. So we pass the Engle-Granger procedure for cointegration for the spread.

- Next, we try to trade around the spread. Since the spread is mean reverting, we fit an Ornstein-Uhlenbeck process to it to find out the entry/exit trade points. As explained in section 2.3.4, we fit an AR(1) process to the spread residuals and estimate the coefficients. The results are shown in Table 2.6:

One immediately notices that the maximum and minimum values of the spread are both within than the prescribed bounds given by $\mu_e \pm \sigma_{eq}$. So in order to optimize the bounds, we chose the bounds of $\mu_e \pm \sigma_{OU}$. Figure 2.5 shows the spread along with these new bounds.

		Value
1	θ	0.0299402858353395
2	μ_e	1.67143910977679
3	σ_{OU}	93.925191805238
4	σ_{eq}	383.830181784731
5	$\max(e_t)$	268.596623259263
6	$\min(e_t)$	-234.374998780455
7	Half-Life	23.15

TABLE 2.6: Results of fitting the Ornstein-Uhlenbeck process to the spread.

To trade the spread, a strategy enter at the bounds and exit at the mean μ_e. Note that since

$$e_t = y_t - \hat{\beta}_1 x_t - \hat{\beta}_0 = \beta'_{\text{Coint}} \mathbf{Y}_t - \mu_e$$

where $\beta_{\text{Coint}} = (1, -\hat{\beta}_1)'$ and $\mathbf{Y}_t = (y_t, x_t)$,

1. Entering a long position on the spread is the same thing as going long 1 unit of stock y_t and going short $\hat{\beta}_1$ units of stock x_t
2. Entering a short position on the spread is the same thing as going long $\hat{\beta}_1$ units of stock x_t and going short 1 unit of stock y_t.

The next chapter concerns itself with the backtesting of the strategy outlined in this section.

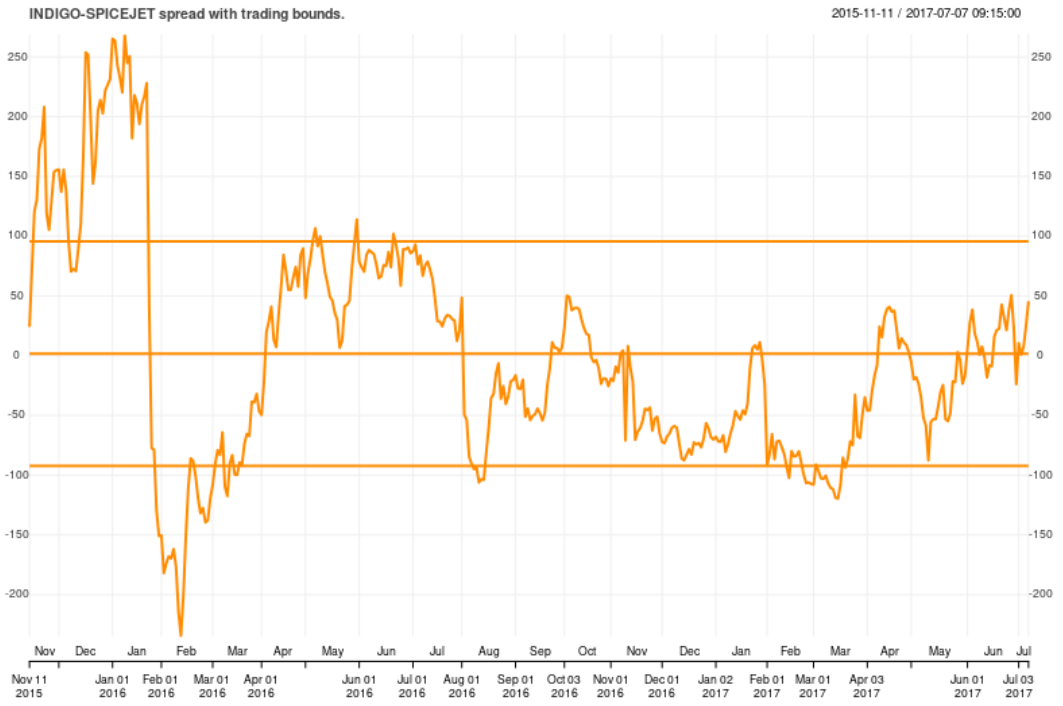


FIGURE 2.5: Plot of the spread along with the trading bounds.

Chapter 3

Backtesting

3.1 Introduction: the 'Quantstrat' library

For backtesting of the strategy outlined in the previous chapter, we use the R library 'Quantstrat'. In this section we outline how to install Quantstrat and it's general features and methods. Most of this section is an adaptation of Guy Yollin's excellent lectures on Quantstrat and Blotter [4].

3.1.1 Installing Quantstrat

As of the time of writing this report, Quantstrat isn't available in the offical R repositories. So in order to get working with it, it is recommended to work with a github version of it. Here are the steps to install Quantstrat.

1. Install packages 'dplyr','devtools' and 'githubinstall' through the usual R command

   ```
   install.packages()
   ```

2. Run the command

   ```
   githubinstall("blotter")
   ```

3. Run the command

   ```
   githubinstall("quantstrat")
   ```

   In both cases choose the repositories of user 'braverock'.

3.1.2 Overview of Quantstrat

Quantstrat is an R package which provides a generic infrastructure to model and backtest signal-based quantitative strategies. It is a high-level abstraction layer (built on xts, FinancialInstrument, blotter, etc.) that allows you to build and test strategies in very few lines of code.

Key features:

- Supports strategies which include indicators, signals, and rules.
- Allows strategies to be applied to multi-asset portfolios.
- Supports market, limit, stoplimit, and stoptrailing order types.
- Supports order sizing, parameter optimization, transaction costs and much more.

3.1.3 Quantstrat backtesting workflow.

Here is a quick overview of the backtesting workflow of Quantstrat.

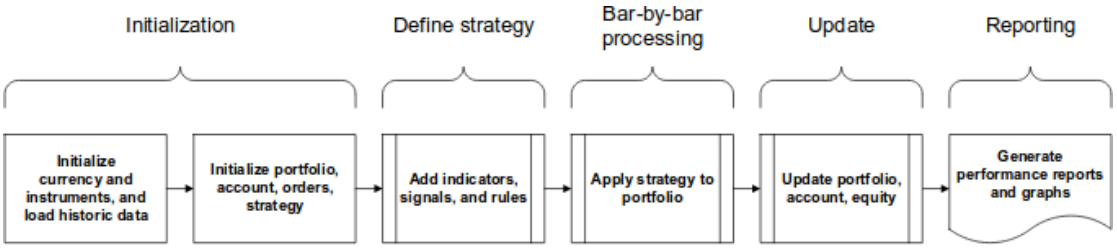


FIGURE 3.1: Basic strategy backtesting workflow of quantstrat.

1. Initialize a *currency* and instruments that contain market historical data.
2. Initialize a *portfolio*, *account*, *orders* and a *strategy*.
3. Add *indicators*. Indicators are quantitative values derived from market data. In our case, indicators are the trading bounds derived from the OU process.
4. Interaction between indicators and market data are used to generate *signals* (e.g. crossovers, thresholds).
5. *Rules* use market data, indicators, signals, and current account/portfolio characteristics to generate *orders*.
6. Interaction between *orders* and market data generates transactions. These transactions are then stored in the portfolio, account and orders objects.

3.2 Implementation

To implement the backtest, we first implement the indicators, signals and rules for the Quantstrat system to analyze.

FIGURE 3.2: Plot of the spread along with the trading bounds.

1. *Indicators*: For our strategy, the indicators will be the mean spread and the first standard deviations away from the mean, got from fitting the Ornstein-Uhlenbeck process to the spread. These are denoted by the red, blue and green lines in Figure 3.2 respectively.

2. *Signals*: The signals of our strategy, in keeping with the dictum of "enter at the bounds and exit at the mean", will be

   - Enter into a short position when there is a spread crossover with the upper bound (green line) from above to below.
   - Exit the short position as soon as there is a spread crossover with the mean (red line) from above to below.
   - Enter into a long position when there is a spread crossover with the lower bound (blue line) from above to below.
   - Exit the long position as soon as there is a spread crossover with the mean (red line) from below to above.

3. *Rules*: Finally, the rules of our strategy is to go long/short 3 units of the spread with a transaction cost of 20 rupees. (These reflect actual numbers which the author himself is comfortable trading! Also, the broker Zerodha in India charges a flat transaction fee of 20 rupees.)

4. *Initial Equity*: We start with an initial equity of 1500 in our portfolio.

Running the backtest, here are the overall trade statistics.

	Trading the spread
Num.Txns	21
Num.Trades	9
Net.Trading.PL	3420.692
Avg.Trade.PL	406.7436
Med.Trade.PL	417.1538
Largest.Winner	675.2896
Largest.Loser	-20
Gross.Profits	3660.692
Gross.Losses	0
Std.Dev.Trade.PL	138.9632
Percent.Positive	100
Percent.Negative	0
Avg.Win.Trade	406.7436
Med.Win.Trade	417.1538
Avg.Daily.PL	406.7436
Med.Daily.PL	417.1538
Std.Dev.Daily.PL	138.9632
Ann.Sharpe	46.46451
Max.Drawdown	-1186.922
Profit.To.Max.Draw	2.881985
Max.Equity	3959.582
Min.Equity	-808.7863
End.Equity	3420.692

This shows an Annualized Sharpe Ratio of 46! However, there are many factors to consider which make this slightly unreasonable.

1. This backtest is not an out-of-sample backtest. To perform that we shall need more data, which is unfortunately not available since INDIGO only went public in 2015.

2. As the old economist's joke goes, there cannot be a 10$ bill lying on the street, for if there was, someone would have already picked it up. Such a big profit may be unrealizable because there might be other players in the market who have already profited from such a trade. Which is why trading at a higher frequency than daily is preferable, which comes with it's own problems.

Here are other statistics.

Trade	1	2	3	4	5
Start	2015-12-09 00:00:00	2016-03-02 00:00:00	2016-06-02 09:15:00	2016-12-16 09:15:00	2017-02-16 09:15:00
End	2016-02-29 00:00:00	2016-05-12 09:15:00	2016-09-26 09:15:00	2017-01-30 09:15:00	2017-05-02 09:15:00
Init.Qty	-3	3	-3	3	3
Init.Pos	-3	3	-3	3	3
Max.Pos	-6	9	-12	3	6
End.Pos	0	0	0	0	0
Closing.Txn.Qty	3	-3	9	-3	-6
Num.Txns	4	6	6	2	3
Max.Notional.Cost	-141.2148	-729.8046	-1006.4567	-218.6253	-472.4139
Net.Trading.PL	509.0721	1282.4376	1012.4433	193.4288	423.3104
MAE	-808.78630	-193.21445	-390.47164	-31.29293	-146.53476
MFE	1401.1361	1554.1873	2168.0725	265.0001	846.0957
Pct.Net.Trading.PL	3.6049497	1.7572343	1.0059483	0.8847502	0.8960583
Pct.MAE	-5.7273497	-0.2647482	-0.3879667	-0.1431350	-0.3101830
Pct.MFE	9.922023	2.129594	2.154164	1.212120	1.791005
tick.Net.Trading.PL	8484.535	14249.307	8437.028	6447.626	7055.174
tick.MAE	-80878.630	-19321.445	-39047.164	-3129.293	-14653.476
tick.MFE	140113.61	155418.73	216807.25	26500.01	84609.57
duration	7084800 secs	6167700 secs	10022400 secs	3888000 secs	6480000 secs

TABLE 3.1: Per-trade Statistics of the backtest.

FIGURE 3.3: Plot of consolidated account equity (initial equity is 1500).

From the above we can get the drawdown and returns plots.

Figure 3.5 shows the entire backtesting summary.

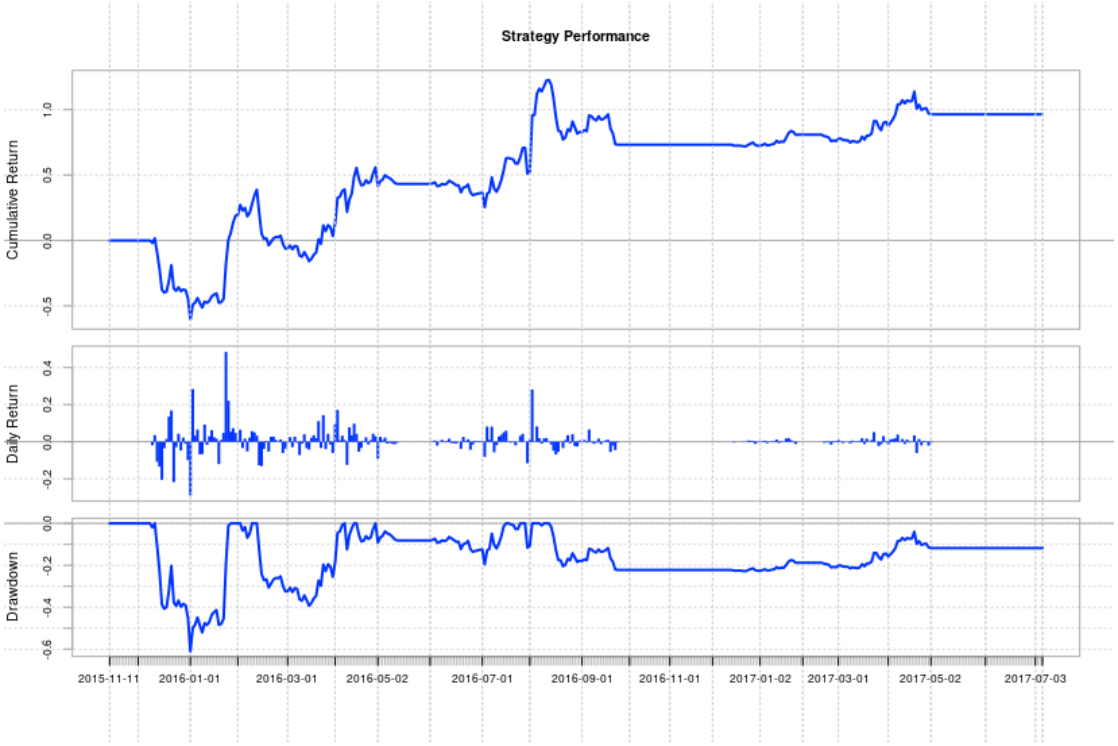


FIGURE 3.4: Cumulative return, Daily Return and Drawdown waterfall plots.

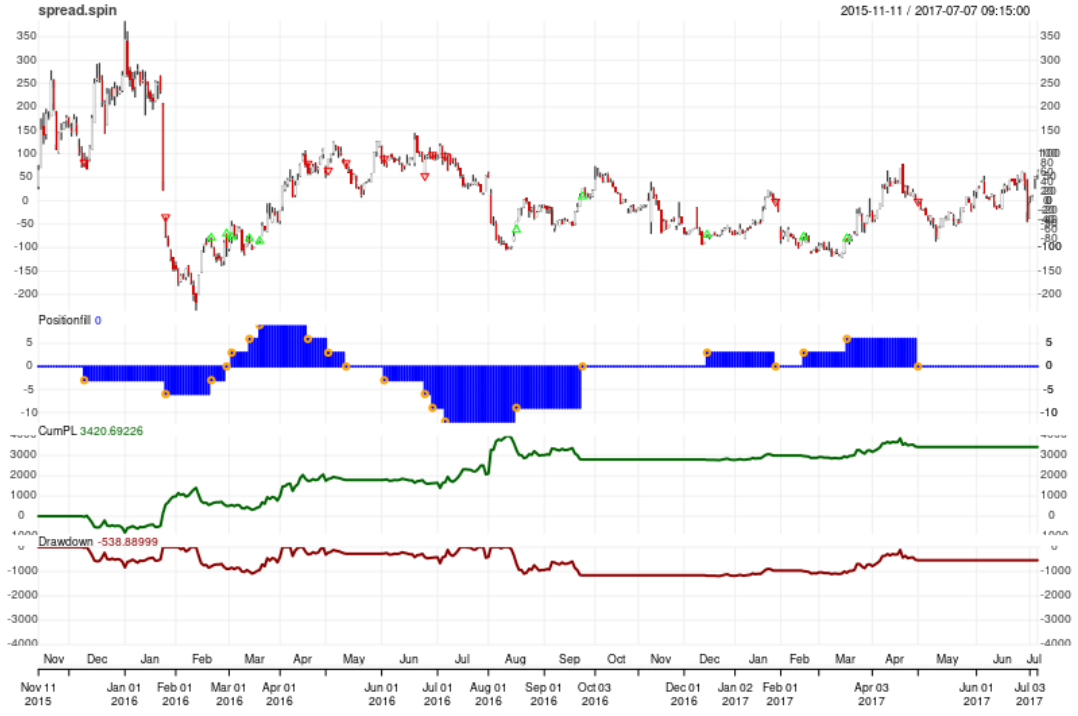


FIGURE 3.5: Spread, Position, Cumulative PnL and Drawdown plot.

Appendix A

About the code and data

This appendix is intended to be a reference for parts of the code which correspond to the sections outlined above.

Most of the code was written in R, with the exception of downloading the data, which was done in python. The data was downloaded by using the Kite Connect API provided by Zerodha, a discount brokerage in India.

Here is a list of code used:

- Chapter 1 : CVA calculation for an Interest Rate Swap. The entire code for this chapter is given in the file '`cva_calculation.R`' file provided.
- For sections 2.1 and 2.2, the estimation and implementation of VAR(p) models, the relevant code is in `time_series_1.R` file.
- For sections 2.3 and 2.4, the Engle-Granger procedure,ADF tests and fitting an OU process to the spread are in the file '`time_series_2.R`'
- For the final chapter on Backtesting, the relevant file is `time_series_3_backtesting.R`.Note that to run this file, one needs to install the `quantstrat` package. Instructions on how to install this package are given in section 3.1.1.

Here is the list of data provided along with the project code:

- 10 minute candles data for RELIANCE,DRREDDY,ITC, CIPLA and TATASTEEL from 2017-04-10 till 2017-07-07. This is used for the VAR(p) model implementation. However we use the close prices for the modelling.

- Daily candles data for INDIGO and SPICEJET from 2015-11-11 till 2017-07-07. The close prices for each day are used for the modelling. The whole candles data is needed for backtesting, because it involves converting to OHLC format.
- Also provided is 10mins and hourly data for INDIGO and SPICEJET from 2017-04-10 till 2017-07-07.

All files are in `.csv` format.

Bibliography

[1] J Gregory. *The XVA Challenge, Counterparty Credit Risk, Funding, Collateral and Capital, 2015*. John Wiley & Sons Ltd, 2015.

[2] J.H.M. Darbyshire. *The Pricing and Trading of Interest Rate Derivatives: A Practical Guide to Swaps*. Unknown Publisher, 2016. ISBN 9780995455511. URL https://books.google.co.in/books?id=ilNbvgAACAAJ.

[3] Helmut Ltkepohl. *New introduction to multiple time series analysis*. Springer, Berlin [u.a.], 2005. ISBN 3540262393. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+366296310&sourceid=fbw_bibsonomy.

[4] Guy Yollin. *Computational Finance and Risk Management:Introduction to Trading Systems*. Unknown Publisher, 2014. URL http://www.r-programming.org/papers.