Javier Zamora
fja2117
Modeling and Market
Making in Foreign Exchange
IEOR E4722
June 5, 2017

Lecture 2 (Forwards Markets) Assignment

Due start of class, Wednesday June 7, 2017

Question 1 (4 marks)

Why are correlations of daily returns of spot vs daily returns of forward prices so high in the FX markets? What are the two requirements a market must support to enforce a high correlation across the forward curve?

The theoretical forward price of a currency pair is a mathematical construction that defines an investment strategy: go long in one of the currencies, short in the other one and pay the applicable interest rates. Any deviation between the "theoretical forward price" and the "market forward price" would create a money-free arbitrage opportunity by means of the execution of the outlined strategy that would correct the deviation. Thus, the market forward price" follows the spot price.

The arbitrage execution relies on the applicability of the strategy, so it is needed:

1. buy to store currencies (and receive an interest rate for them)
2. borrow and short the counterparty currency of the pair (and pay an interest rate)

Question 2 (2 marks)

Why is risk management more complex for an FX forwards risk manager than for an FX spot risk manager?

Because a FX forward risk manager needs to evaluate the exposure to risk of having issued contracts with different settlement times.

Intuitively, let's imagine a cantilever beam. Punctual loads can appear at any length of the beam, pushing upwards or downwards. If a load were the risk of a forward contract, and the load location was the tenor. The challenge for the forwards risk manager is keeping the beam balanced by allocating contracts/hegdes along time that would prevent the beam to move. (More on this on Q4).

Question 3 (2 marks)

Explain why risk to FX forward points can be expressed as risk to non-USD interest rates.

A FX forward point is the difference in forward rates between the forward and spot. The risk exposure of the spot is managed through the portfolio's delta position; and the USD interest rate risk is also normally managed separately. The only exposed dimension of risk in the forward contract is the non-USD interest rate risk.

Question 4 (4 marks)

Assume a portfolio has just one FX forward position in it, settling on a date T which lies between two benchmark settlement dates T_1 and T_2. Derive the notionals N_1 and N_2 of the benchmark forwards which hedge the portfolio risk assuming triangle shocks to the benchmark non-USD interest rates, as shown on page 21 of the lecture notes.

To hedge the portfolio, it is needed to allocate forward contracts at T_1 and T_2 that oppose to the position taken at T. The notionals N_1 and N_2 have to be such that the portfolio is balanced, i.e. the risk of the portfolio is zero.

Since the risk of a forward contract depends of the non-USD interest rate, the parameter to balance is the variation of the price of the forward contract respect to the non-USD interest rate. Then for the contract at T,

$$V_Q = \frac{\partial V}{\partial Q} = \frac{\partial}{\partial Q}(S \cdot e^{-QT} - K \cdot e^{-RT}) = -S \cdot T \cdot e^{-QT}$$

For the contracts at T_1 and T_2 we have V_1 and V_2, and V_{Q1} and V_{Q2}, respectively.
Assuming triangle rate shocks allows to continue with the analogy of the cantilever beam. Then, the portfolio will be balanced if 2 conditions occur:

i) $N_1 V_{Q1} + N_2 V_{Q2} = V_Q$
ii) $N_1 V_{Q1} T_1 + N_2 V_{Q2} T_2 = V_Q T$

The first condition says that the total amount of risk taken long and short have to be the balanced, and the second that the total amount of risk taken long and short *measured at the spot*, have to be also the balanced. Then, from (ii)

$$N_1 V_{Q1} = \frac{V_Q T - N_2 V_{Q2} T_2}{T_1}$$

Inserting this expression into (i)

$$\frac{V_Q T - N_2 V_{Q2} T_2}{T_1} + N_2 V_{Q2} = V_Q => N_2 V_{Q2}(T_2 - T_1) = V_Q(T - T_1) => N_2 V_{Q2} = V_Q \frac{T - T_1}{T_2 - T_1}$$

And coming back to (ii)

$$N_1 V_{Q1} T_1 + V_Q \cdot \frac{T - T_1}{T_2 - T_1} \cdot T_2 = V_Q T =>$$

$$=> N_1 V_{Q1} = V_Q \left(\frac{T}{T_1} - T_2 \frac{T - T_1}{T_1 (T_2 - T_1)} \right) = V_Q \left(\frac{T(T_2 - T_1) - T_2 (T - T_1)}{T_1 (T_2 - T_1)} \right) => N_1 V_{Q1} = V_Q \frac{T_2 - T}{T_2 - T_1}$$

Then

$$N_1 V_{Q1} = V_Q \frac{T_2 - T}{T_2 - T_1} => -N_1 \cdot S \cdot T_1 \cdot e^{-QT_1} = -S \cdot T \cdot e^{-QT} \cdot \frac{T_2 - T}{T_2 - T_1}$$
$$N_2 V_{Q2} = V_Q \frac{T - T_1}{T_2 - T_1} => -N_2 \cdot S \cdot T_2 \cdot e^{-QT_2} = -S \cdot T \cdot e^{-QT} \cdot \frac{T - T_1}{T_2 - T_1}$$

Where it has been assumed that $Q_1 \approx Q$, and $Q_2 \approx Q$. And then,

$$N_1 = \frac{T}{T_1} \cdot \frac{T_2 - T}{T_2 - T_1} \cdot e^{-Q(T - T_1)}$$
$$N_2 = \frac{T}{T_2} \cdot \frac{T - T_1}{T_2 - T_1} \cdot e^{-Q(T - T_2)}$$

Question 5 (4 marks)

Explain principal component analysis and factor models, focusing on the differences between the two approaches to reduce dimensionality.

Often understood as similar approaches to reduce dimensionality, Principal Component Analysis (PCA) and Factor Models are conceptually very different.

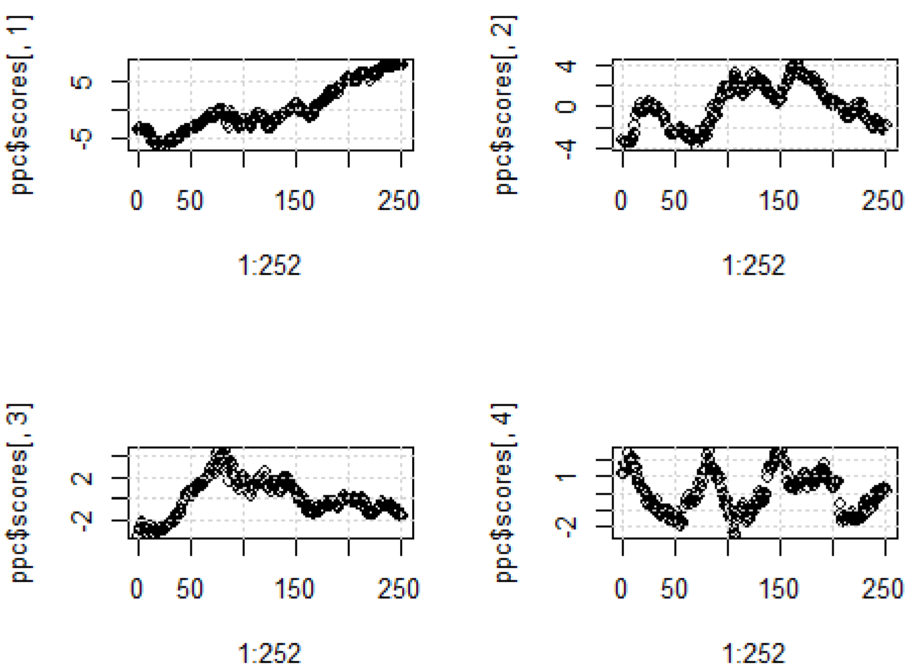
Given a dataset with n parameters and m rows. The idea behind these methods is that it exists causation between the n parameters and a fundamental set of r parameters (r<<n). In a sense, the n parameters of the dataset are the “shadows” of these fundamental r parameters, and these dimensionality reduction approaches try to obtain them.

The idea behind PCA is that the integral of the spectral density of the data yields its total variance. Then, it should be possible to identify the bands of the spectrum that concentrate more variance, or the “components” of the spectrum that concentrate more variance. If we find -let’s say- r components that explain 95% of the variance of the data, we could defend that exists a set of r main components, or r Principal Components, which are the fundamental cause of the n parameters of the data. Thus, the dataset is -almost- fully explained by just these r parameters.

From a methodological point of view, performing spectral analysis has some challenges. Then, since the final goal is try to identify all clusters of variance, a more popular approach is to calculate the covariance matrix of the dataset. This covariance matrix could be seen as the n-dimensional representation of a r dimensional vector sub-space. And a diagonalization transformation would provide the canonical basis of the r vector sub-space, with the eigenvectors being the Principal Components.

The main advantage of PCA is also the main difficulty when dealing with this method. All the Principal Components have “meaning”. The method is brilliant in the sense that provides the main causes that produce the dataset on hand. However, to understand the meaning of each one of the Principal Components is, more often than not, far from obvious.

The plots below are the PC of the 30 DJIA stocks along 2010 (from a Statistical Machine Learning homework).



These 4 PCs explain almost 90% of the variance of the data. It could be said that the 1st one is the drift of the US economy, and the 2nd to 4th might be a sort of seasonal effects; and yet to assign a particular meaning might be difficult, and controverted.

Thus, the main problem of PCA is that the conclusions obtained might be difficult to sell.

Factor Model tries to address the same problem as PCA does, there are a short set of hidden fundamental parameters that explain the dataset. But allows to define the number of these fundamental parameters we want to use (here called factors), and finds the solution that minimize the error. Factor Model is just a -fancy- exercise of regression.

So, let’s say we had 4 points belonging to a function in a cartesian chart. We could find an unique cubic polynomial that perfectly fits the data. But we could also have 2 points of the same function, and the first 2 derivatives on the same location, and we could use Newton polynomials to build a different cubic polynomial. Or, a point and 3 derivatives leading, by means of Taylor series, to a different cubic polynomial.

Regression is about taking measurements of an unknown function, and projecting this measurements into the -often- polynomial vector sub-space, minimizing some error. There are choices to be made, like which norm or method to use to measure the error; or the canonical base of the polynomial vector sub-space; or even if just take the data, or -numerically- calculate some derivatives, or any combination of data and derivatives.

The main advantage of Factor Model in particular and regression methods in general, is that we can project the data into a model that already makes sense to us. We always can find a model that is consistent with what we expect to find. Thus, results are inherently less controverted.

Question 6 (10 marks)

This programming question will try to determine whether using a factor-based approach to reducing dimensionality is better than an ad hoc method.
We start by assuming a toy market: spot = 1, asset currency interest rate curve = Q(T) = flat at 3%, and denominated currency interest rate curve = R(T) = flat at 0%. We assume two benchmark dates, T_1 = 0.25y and T_2 = 1y; we will use forwards to those settlement dates to hedge the forward rate risk (or equivalently, the risk to moves in the asset currency interest rate) of our portfolio.

In the toy market, we assume that we know the dynamics of the asset currency interest rate:

$$dQ = \sigma_1 e^{-\beta_1 T} dz_1 + \sigma_2 e^{-\beta_2 T} dz_2$$
$$E[dz_1 dz_2] = \rho\, dt$$

where σ_1=1%/sqrt(yr), σ_2=0.8%/sqrt(yr), β_1=0.5/yr, β_2=0.1/yr, and ρ=-0.4.

The portfolio to hedge has one position: a unit asset-currency notional of a forward contract settling at time T. You'll try this for values of T in [0.1,0.25,0.5,0.75,1,2] to see how performance changes for portfolios with risk to different tenors.

You will try three different hedging strategies: one where you choose the hedge notionals (of forwards settling at times T_1 and T_2) based on the triangle shock we discussed in class (though as there are only two benchmarks here, the T1 shock will be flat before T1 and the T2 shock will be flat after T2); one where the notionals are set to hedge the actual two shocks from the factors described above; and lastly, one where you don't hedge at all.

The result should show that setting hedge notionals based on the true factor shocks should provide a better hedge performance than based on the ad hoc triangle shocks. You should analyze just how much better that performance is.

Run a Monte Carlo simulation where you do the following on each run, for each value of T, for each of the three hedging strategies described above:

1. Construct a portfolio long 1 unit of the forward settling at time T
2. Add in the hedges: two forwards, settling at times T1 and T2, with notionals set to hedge the portfolio (either against the two triangle shocks or against the two factor shocks). Don't bother adding the hedges in the third hedge scenario where we leave the portfolio unhedged.
3. Simulate the portfolio forward a time dt=0.001y. That will result in the asset-currency rates moving according to the factor model described above, which shocks the benchmark rates for tenors T1 and T2, and for the portfolio's risk tenor T. Determine the PNL realized.

Then construct the PNL distributions for the three hedging approaches. The unhedged version is the benchmark: you should compare how much more effectively the PNL standard deviation is reduced by hedging according to the true factors vs hedging according to the ad hoc triangle shocks.
Do this for all the values of T listed above, and discuss your results.

The results of the simulation are summarized in the table below,

Tenor	0.1	0.25	0.5	0.75	1	2
No Hedging	3.02E-05	7.21E-05	1.31E-04	1.82E-04	2.26E-04	3.63E-04
Triangle hedging	2.05E-06	0.00E+00	1.80E-06	2.57E-06	0.00E+00	1.32E-04
Triangle hedging efficiency	93.21%	100.00%	98.63%	98.59%	100.00%	63.77%
Factor hedging	3.72E-10	8.22E-20	1.34E-09	1.73E-09	9.98E-20	3.32E-08

Factor hedging provides the best strategy; but the triangle hedging strategy is actually very good between the two benchmark tenors.

Below can be found the python code and the output of the program.

```
import scipy

def main():
   vol1=0.01
   vol2=0.008
   b1=0.5
   b2=0.1
   rho=-0.4
   T1=0.25
   T2=1
   Q=0.03
   dt=1e-3
   numruns=100000
   scipy.random.seed(1)

   rr=scipy.zeros((4,6))
   rr[0,]=[0.1, 0.25, 0.5, 0.75, 1, 2]
   for contT in range(0,6):
      T=rr[0,contT]
      for hedge in range(0,3):
         dz1=scipy.random.normal(0,scipy.sqrt(dt), numruns)
         dz2=rho*dz1+scipy.sqrt(1-rho**2)*scipy.random.normal(0,scipy.sqrt(dt), numruns)
         dQT=vol1*scipy.exp(-b1*T)*dz1+vol2*scipy.exp(-b2*T)*dz2
         dQ1=vol1*scipy.exp(-b1*T1)*dz1+vol2*scipy.exp(-b2*T1)*dz2
         dQ2=vol1*scipy.exp(-b1*T2)*dz1+vol2*scipy.exp(-b2*T2)*dz2
         pnl=scipy.exp(-(Q+dQT)*T)-scipy.exp(-Q*T)
         pnl=pnl-N1(vol1,vol2,b1,b2,T,T1,T2,Q,hedge)*(scipy.exp(-(Q+dQ1)*T1)-scipy.exp(-Q*T1))
         pnl=pnl-N2(vol1,vol2,b1,b2,T,T1,T2,Q,hedge)*(scipy.exp(-(Q+dQ2)*T2)-scipy.exp(-Q*T2))
         rr[hedge+1,contT]=pnl.std()
```

```
    scipy.set_printoptions(precision=3)
    print(rr)

def N1(vol1,vol2,b1,b2,T,T1,T2,Q,hedge):
    Not1=0
    dT=T2-T1
    if hedge == 1:
        if T<=T1:
            Not1=T*scipy.exp(-Q*(T-T1))/T1
        elif T>T1 and T<T2:
            Not1=(T2-T)*T*scipy.exp(-Q*(T-T1))/(dT*T1)
    elif hedge==2:
        dz1=-scipy.exp(b1*T2-b2*dT)/(vol1*(1-scipy.exp((b1-b2)*dT)))
        dz2=scipy.exp(b2*T1)/(vol2*(1-scipy.exp((b1-b2)*dT)))
        dQT=vol1*scipy.exp(-b1*T)*dz1+vol2*scipy.exp(-b2*T)*dz2
        Not1=dQT*T/T1*scipy.exp(-Q*(T-T1))
    return Not1

def N2(vol1,vol2,b1,b2,T,T1,T2,Q,hedge):
    Not2=0
    dT=T2-T1
    if hedge == 1:
        if T>=T2:
            Not2=T*scipy.exp(-Q*(T-T2))/T2
        elif T>T1 and T<T2:
            Not2=(T-T1)*T*scipy.exp(Q*(T2-T))/(dT*T2)
    elif hedge==2:
        dz1=-scipy.exp(b1*T1+b2*dT)/(vol1*(1-scipy.exp(-(b1-b2)*dT)))
        dz2=scipy.exp(b2*T2)/(vol2*(1-scipy.exp(-(b1-b2)*dT)))
        dQT=vol1*scipy.exp(-b1*T)*dz1+vol2*scipy.exp(-b2*T)*dz2
        Not2=dQT*T*scipy.exp(Q*(T2-T))/T2
    return Not2

if __name__=="__main__":
    main()
```

```
[[  1.000e-01   2.500e-01   5.000e-01   7.500e-01   1.000e+00   2.000e+00]
 [  3.024e-05   7.212e-05   1.310e-04   1.820e-04   2.263e-04   3.629e-04]
 [  2.054e-06   0.000e+00   1.797e-06   2.567e-06   0.000e+00   1.315e-04]
 [  3.719e-10   8.220e-20   1.342e-09   1.730e-09   9.981e-20   3.322e-08]]
```