Hedge Fund Strategies and Risk

Carry Trade on Currencies

Group 3: Fangyu Qu, Jiawei Wei, Jenny Xu, Jing Zhang, Yunyan Zhang

October 28th, 2016

Introduction

One of the most widely known and profitable strategies in currency markets are carry trades, where one systematically sells low interest rate currencies and buys high interest rate currencies. Such a strategy exploits what academics call "forward-rate bias" or the "forward premium puzzle", that is, the forward rate is not an unbiased estimate of future spot. Academics believe that this is possible because investors who employ the carry trade expose themselves to currency risk. Investors taking this risk might be rewarded by positive returns over time.

FX carry trade is usually implemented by trading FX forwards. As FX forward is one of the most liquid FX market and can simulate the "long high short low" investment strategy, we use daily closing price of FX forwards to calculate returns.

Specification

Universe

We trade 8 pairs of currencies in our carry variant trade. We use 3M forward of the following currency pairs: JPY-USD, GBP-USD, SEK-USD, AUD-USD, CAD-USD, JPY-EUR, GBP-EUR, SEK-EUR.

In J.P. Morgan's Income FX strategy, the portfolio only borrowed USD and EUR. However, due to the liquidity of FX forward market, we didn't limit our short positions on other G10 currencies. We didn't find enough data for AUD-EUR and CAD-EUR on Bloomberg, so we didn't include them in our portfolio.

Data range: From 1/2/2012 to 10/24/2016

During these period, we can find the most complete data. We tried to use the data from 2005 to 2008 before the financial crisis, however, many countries' short-term treasury yield and forward contract are not available on Bloomberg.

Data source: Bloomberg

We found daily closing FX spot rate, daily closing 3-month FX forward outright quote and daily closing 3-month Treasury yield on Bloomberg.

Signal Methodology

Carry-to-Risk Ratio: $\frac{i_1 - i_2}{\sigma}$

The Carry-to-Risk Ratio is a fraction, the numerator of which is the interest rate in Currency One of an eligible currency pair minus the interest rate in Currency Two of an eligible currency pair and the denominator of which is the annualized exchange rate volatility of that currency pair.

In the first step, we compared the carry to risk ratio among our G10 group, where the market has adequate liquidity. We only choose pairs which includes EUR and USD which has relatively low interest rate and easy to short. Thus we generate eight trading pairs(JPY /USD,GBP/USD,SEK/USD,AUD/USD,CAD/USD,JPY/EUR,GBP/EUR,SEK/EUR). Then we calculate the Carry-to-Risk ratio of each pair on the first trading day of a month, and select the top 2 pairs to get into position on their forward contract. And in our code we generate a position matrix to determine which pairs to buy by denoting with 1 and no position with 0.

Portfolio Construction & Optimization Methodology

Re-balance every month: Roll Dates are set as the first trading day of every month. On every rebalance day, we rank the eight pairs by carry-to-risk ratio and trade the top two pairs. At first, We invest equally when there is no optimization.

A portfolio consist of one or more currency pairs, and investors usually use FX forwards as investment vehicles. If there are more than one pair in their portfolio, one may choose different weighting methods. The simplest one is to equally weight every currency pair. For example, J.P. Morgan Income FX Strategy equally weights G10 currency pairs. So in our simple backtest we will only trade the top 2 pairs forward with equal weight and then we will consider about the optimization.

Based on risk-adjusted return, we consider to apply one of two ways of optimization:

Mean-Variance optimization: $\omega^* = argmin_{\omega}(\omega'\Sigma_{t-\psi,t}\omega)$ such that $\overline{\mu}'_{t-\psi,t}\omega \geq \mu_{target}$, where ω^* is updated portfolio weight, μ is the difference in interest rate, $\Sigma_{t-\psi,t}$ is the covariance matrix of exchange rate estimated using data during $(t-\psi, t)$ period;

Carry-to-Risk Ratio Optimization: $\omega^* = argmin_{\omega}\left(\frac{\overline{\mu}'_{t-\psi,t}\omega}{\sqrt{\omega'\Sigma_{t-\psi,t}\omega}}\right)$. Done by solving the efficient frontier of the portfolio and finding its tangent that passes the origin.

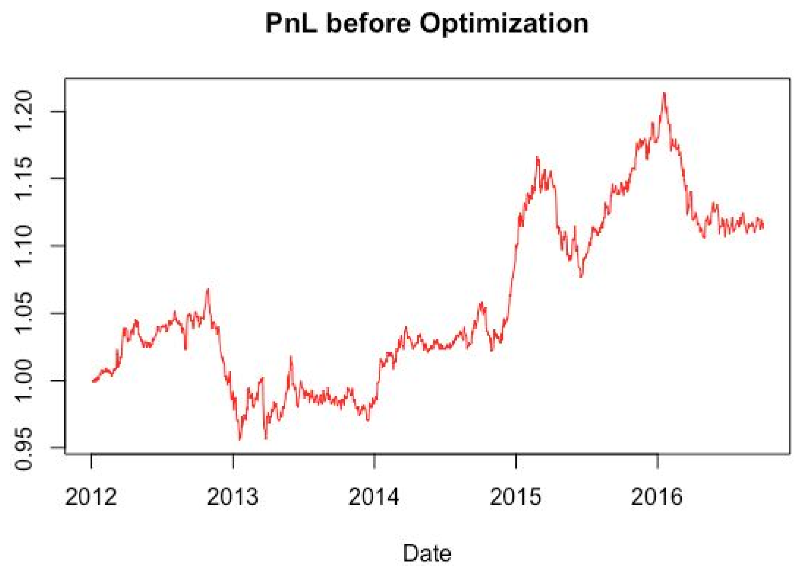
We finally choose the second method, with the intention to make full use of our signals. The way we realize it is through finding the best solution and then multiply the weight to our original position matrix.

Execution

We use 3-month FX forwards to simulate longing high-yield currency and shorting low-yield currency. We induce a cumulative daily PNL equation, which is (f-s0)*(l/(s0*N))+(s-s0)/s0. In this equation we have f: the forward we on the day we trade, s0: the spot rate on the date we trade, l: the holding period, N: the lifetime of a forward contract. In this way we contribute our cumulative return to two parts: the carry interest and the profit of change of spot rate. We refresh the data everyday to generate cumulative simple return. This time we will not set any stop loss constraints in our simple backtest.

Implementation

Cumulative simple return



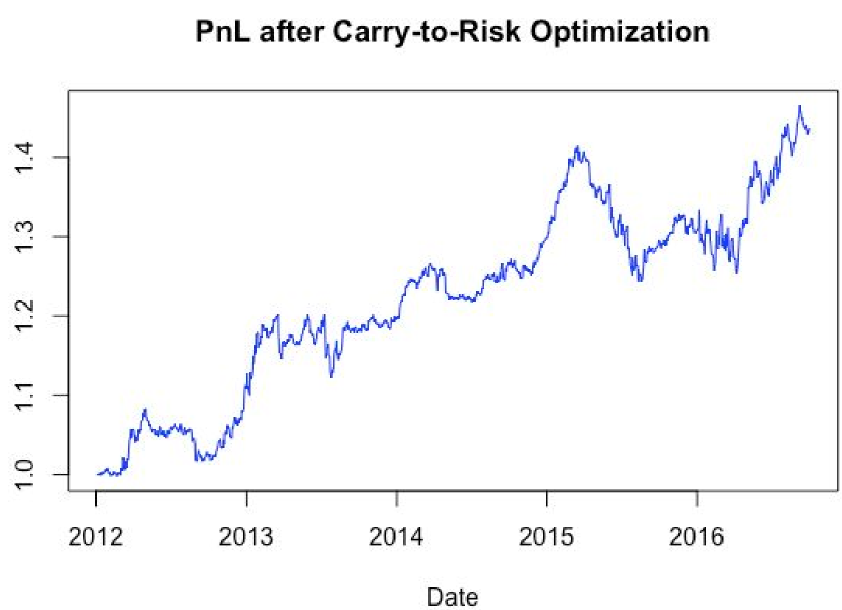
Annualized return: 2.27%

Annualized volatility: 6.16%

Maximum drawdown: 10.55%

Sharpe ratio (with a 0% risk-free interest rate): 0.37

Kurtosis: 2.00



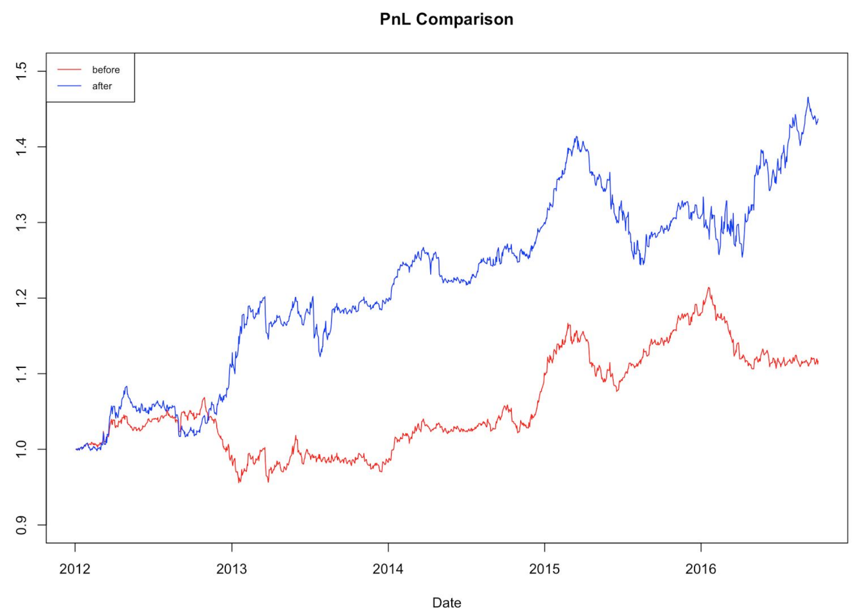
Annualized return: 7.65%

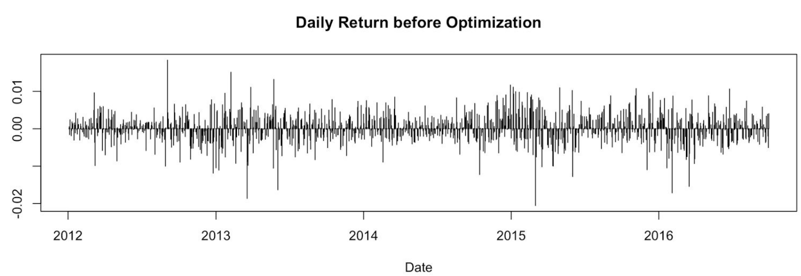
Annualized volatility: 10.26%

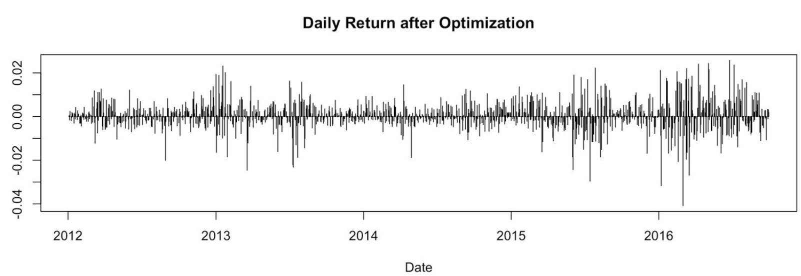
Maximum drawdown: 12.03%

Sharpe ratio (with a 0% risk-free interest rate): 0.75

Kurtosis: 2.23







Comparison with J.P. Morgan's Income FX strategy

From JPM 2005's FX Carry trade research[1]: it has delivered an annualized return of 12.32% with annualized volatility of 8.07% since November 1995, attaining a Sharpe ratio larger than 1. However, after financial crisis, as the central banks keep lower interest rate, it is hard to get a 2-digit annualized return.

Comparison within simulation

Between our pre- and post-optimization results, we can see that introducing Carry-to-Risk optimization significantly enhance the strategy's performance though volatility will also increase. The gap between each other has widened since 2013.

Refinement

Including costs

Because we trade monthly, transaction cost is not a big issue - assume a 5bps transaction cost each time, for a 5-year period, we will lose approximately 0.25%, which is trivial compared with our 40% 5-year return.

Improved Portfolio construction

We limit the largest position on long is 2x notional and the smallest position on short is -1x notional. If use Carry-to-Risk optimization to calculate weights, we can get 11x on one currency trade, while -10x on the other one, which exposes us to substantial risk.

Difficulties

At first, it was difficult for us to figure out the formula for daily Pnl as the time to maturity of forward contract changes over time. Then we approximate the daily Pnl to be ret(fx,t,t+1) = (spot(fx_t)-spot(fx_t+1))/spot(fx_t) + carry(fx,t,t+1)

[1] https://www.sec.gov/Archives/edgar/data/19617/000089109207000565/e26362_424b2.htm

Conclusion

Trading recommendation

Prior to the 2008 financial crisis, the carry trade was popular and profitable. However, we wouldn't recommend the carry variant strategy, especially in today's environment. Many countries, including those that had high interest rates before, are implementing monetary strategies such as quantitative easing. As a result, the carry trade would not be that profitable compared to its previous performance.

Appendix: Source code

We use R to implement the strategy. The following is the code with optimization. Without optimization, simply replace the opt_position as {0.5, 0.5}.

```
library(chron)
library(lubridate)
library(ggplot2)
library(zoo)
library(reshape2)
setwd("~/Downloads/Debugged\ Ver")
forward<-read.csv("3mo_fwd.csv")
spot<-read.csv("spot.csv")
i_differential<-read.csv("Numerator.csv")
i_differential=i_differential[,-1]
i_matrix=matrix(0,nrow=57,ncol=2)
##calculate return
for (i in 1:57){
i_matrix[i,1]=i_differential[i,which(position[i,]==1)[1]]
i_matrix[i,2]=i_differential[i,which(position[i,]==1)[2]]
}
#### calculate covariance of each 2 pair #####
spot[,1]<-as.Date(spot[,1],format='%Y/%m/%d')
exracov<-matrix(0,nrow = 57,ncol = 3)
```

```
for(i in 1:56){

exracov[i,1]<-cov(spot[1+(21*(i-1)):(21*i),c(which(position[i,]==1)[1],which(position[i,]==1)[2])
])[1,1]

exracov[i,2]<-cov(spot[1+(21*(i-1)):(21*i),c(which(position[i,]==1)[1],which(position[i,]==1)[2])
])[2,2]

exracov[i,3]<-cov(spot[1+(21*(i-1)):(21*i),c(which(position[i,]==1)[1],which(position[i,]==1)[2])
])[1,2]
}
 cova<-matrix(0,ncol = 2,nrow = 2)
optSharpe <- function(returns, cov){
 try(if(length(returns) != ncol(cov) | length(returns) != nrow(cov))
  stop("cov must be a square matrix with its # of rows of columns equal to
length(returns)"))
 ones <-         rep(1, length(returns))
 A <- returns %*% solve(cov) %*% returns
 B <- returns %*% solve(cov) %*% ones
 C <- ones %*% solve(cov) %*% ones
 D <- A * C - B * B
 optSha <- A / B
 optPort <- ((optSha * C - B ) / D * returns) %*% solve(cov) +
  ((A - optSha * B ) / D * ones) %*% solve(cov)
if(optPort[1] > 2){
optPort = c(2,-1)
}
if(optPort[2] > 2){
optPort = c(-1,2)
}
 return(optPort)
}
```

```
Initial=100000000
forward[,1]<-as.Date(forward[,1],format='%Y/%m/%d')
#daily pnl
PNL=array(dim = length(forward[,1]))
PNL[1]=0
# We enter the position at the beginning of each month
m=0 #month
y=2012 #year
k=0 #carry-to-risk signal row
N=90
for (n in 2:length(forward[,1]))
{
l=((month(forward[n,1])-1)%%3)*30+day(forward[n,1])
 if (month(forward[n-1,1])==12){ #if the date goes to next year, then the month should be 1
  if (year(forward[n,1])==(y+1)){
   m=1
   y=y+1
   k=k+1
   if (k==1)
   {opt_position=c(0.5,0.5)}
   else
   {cova[1,1]<-exracov[k-1,1]
   cova[1,2]<-exracov[k-1,3]
   cova[2,1]<-exracov[k-1,3]
   cova[2,2]<-exracov[k-1,2]
   opt_position=optSharpe(i_matrix[k,],cova)}
print(opt_position)
   f=forward[n,which(position[k,]!=0)+1]
   s0=spot[n,which(position[k,]!=0)+1]
   print(f)
   print(s0)
   PNL[n]=PNL[n-1]-0.0005
lastPNL = PNL[n]
```

```
  }
  else
   {
   s=spot[n,which(position[k,]==1)+1]
   PNL[n]=sum(as.matrix((f-s0)*(l/(s0*N))+(s-s0)/s0)*opt_position)+lastPNL
  }
 }else{
  if (month(forward[n,1])==(m+1)&&(m+1)<12){ #if the date goes to next month
   m=m+1
   k=k+1
   if (k==1)
   {opt_position=c(0.5,0.5)}
   else
   {cova[1,1]<-exracov[k-1,1]
   cova[1,2]<-exracov[k-1,3]
   cova[2,1]<-exracov[k-1,3]
   cova[2,2]<-exracov[k-1,2]
   opt_position=optSharpe(i_matrix[k,],cova)}
   f=forward[n,which(position[k,]!=0)+1]
   s0=spot[n,which(position[k,]!=0)+1]
print(opt_position)
   print(f)
   print(s0)
   PNL[n]=PNL[n-1]-0.0005
lastPNL = PNL[n]
  }else{
   s=spot[n,which(position[k,]==1)+1]
   PNL[n]=sum(as.matrix((f-s0)*(l/(s0*N))+(s-s0)/s0)*opt_position) + lastPNL
  }
 }
}
PNL = PNL + 1  ### to calculate cumulative simple returns
plot(forward[,1],PNL,type="l",col="blue",xlab = "Date",ylab=" ")
```