COMP226: Slides 15

Slippage revisited

Rahul Savani

rahul.savani@liverpool.ac.uk

Overview

- So far we have **not** included slippage in our backtests
- Results without slippage are **unrealistic**
- **How should we model slippage with daily data?**

Reminder: what is slippage?

- Reminder: Slippage is the difference between the **expected execution price** (e.g. best bid or best ask) and **execution price**

- For us, using **daily data**, the expected execution price may be the Open or Close price for example

- Execution price: (average) price you actually traded at; we won't find this our in backtesting so we will be **conservative** in our assumption

How to be conservative?

If the actual execution price differs from the expected one, either:

- The execution price was **better** (**positive** slippage)
- The execution price was **worse** (**negative** slippage)

We will be **conservative** and assume **negative slippage**:

- go **long** (buy): execution **higher** than expected price
- go **short** (sell) execution is **lower** than expected price

This is the **standard approach** for backtesting

Warning

As noted before, one can quite easily design a strategy that:

- historically perform excellently without slippage
- performs poorly if realistic slippage is included

So, **incorporating reaslitic slippage is crucial**

Market versus limit orders

Recall:

Market orders guarantee execution but not price

- You (almost always) get filled, but may incur large slippage
- **Need to carefully model execution price**

Limit orders guarantee price but not execution

- **Need to carefully model whether we get filled**

How to model slippage?

There are **numerous issues** to consider with modelling slippage:

- characteristics of the particular trading strategy, i.e., in what **market conditions** does one trade: Slippage tends to be worse if others are doing the same as you. For example:

  Negative slippage likely if **buying during a rally**

  Positive slippage likely if **buying during a sell-off**

- characteristics of the particular market and the **type of data** available - **to properly model slippage we really need the data for prices and sizes in the book**

Limit order book data

- **Limit order book data** allows to model market orders and limit orders much more accurately than without it

- With the **best bid** and **best ask** (price and volume)

  - you can use the appropriate one as your expected price (bid for sell market order, ask for buy market order)

  - provided your trade size is significantly smaller than the corresponding volume you might assume **no slippage**

  - otherwise you can model slippage using **other levels in the book** or using the **bid-ask spread**

Our slippage model for daily data

- We describe the slippage model used in **our backtester framework** as used in COMP226 Assignment 2 and COMP396

- Without order book data, **strong assumptions** are required

- We are using OHLC (Open/High/Low/Close) **daily data**

- **For simplicity we model profit and loss (not returns)**

- In the framework, all trades occur **at the open**

- The slippage incurred will be based on the **overnight gap** (difference between the previous close and the open)

- If the open is very different from the close then the market is currently **volatile**. Consequently, slippage (positive or negative) is likely to be greater

Implementation

```r
getOpenClosePrices <- function() {
    prices <- getData() # gets GSPC prices stored offline
    prices <- merge(Op(prices),Cl(prices)) # merge open, close
    colnames(prices) <- c("Open","Close") # rename columns
    return(prices)
}
```

```
> source('utilities.R')
> head(prices <- getOpenClosePrices())
              Open   Close
2007-01-03 1418.03 1416.60
2007-01-04 1416.60 1418.34
2007-01-05 1418.34 1409.71
2007-01-08 1409.26 1412.84
2007-01-09 1412.84 1412.11
2007-01-10 1408.70 1414.85
```

Augment prices

with overnight gap and change from open to open

```r
augmentedPrices <- function(prices) {
    # input xts with columns "Open", "Close"

    overnightGap <- prices[,"Open"] - lag(prices[,"Close"])

    openDiff <- diff(prices[,"Open"])
    # same as prices[,"Open"] - lag(prices[,"Open"])

    prices <- merge.xts(prices,overnightGap,openDiff)
    colnames(prices)[3:4] <- c("Gap","OpenDiff")

    return(prices)
}
```

```
> head(augmentedPrices(prices))
              Open   Close   Gap OpenDiff
2007-01-03 1418.03 1416.60    NA       NA
2007-01-04 1416.60 1418.34  0.00    -1.43
2007-01-05 1418.34 1409.71  0.00     1.74
2007-01-08 1409.26 1412.84 -0.45    -9.08
2007-01-09 1412.84 1412.11  0.00     3.58
2007-01-10 1408.70 1414.85 -3.41    -4.14
```

Simple book-keeping strategy

```r
run <- function(prices,pos,slipMult) {
    # prices and pos should be same length
    prices <- augmentedPrices(prices)
    pnl <- slip <- rep(0,nrow(prices))
    pos[1] <- 0 # start flat
    for (i in 2:(nrow(prices)-1)) { # exit at open on last day
        if ((pos[i] - pos[i-1]) != 0) { # traded
            slipPerUnit <- abs(slipMult * as.numeric(prices[i,"Gap"]))
            slip[i] <- slipPerUnit * abs(pos[i] - pos[i-1])
        }
        pnl[i+1] <- pos[i]* as.numeric(prices[i+1,"OpenDiff"])
        pnl[i+1] <- pnl[i+1] - slip[i]
    }
    return(merge.xts(prices,pos,pnl,slip))
}
```

Example

```
> opens <-  c(100,110,120,100,95,105,105,110)
> closes <- c(112,115,110,102,100,111,101,100)

> prices <- xts(cbind(opens,closes),as.Date(1:length(opens)))
> colnames(prices) <- c("Open","Close")

> head(augmentedPrices(prices))
           Open Close Gap OpenDiff
1970-01-02  100   112  NA       NA
1970-01-03  110   115  -2       10
1970-01-04  120   110   5       10
1970-01-05  100   102 -10      -20
1970-01-06   95   100  -7       -5
1970-01-07  105   111   5       10
```

Example - without slippage

```
> pos <- c(0,-1,-2,1,1,3,0,0)
> pnl <- run(prices,pos,slipMult=0)
> pnl
           Open Close Gap OpenDiff pos pnl slippage
1970-01-02  100   112  NA       NA   0   0        0
1970-01-03  110   115  -2       10  -1   0        0
1970-01-04  120   110   5       10  -2 -10        0
1970-01-05  100   102 -10      -20   1  40        0
1970-01-06   95   100  -7       -5   1  -5        0
1970-01-07  105   111   5       10   3  10        0
1970-01-08  105   101  -6        0   0   0        0
1970-01-09  110   100   9        5   0   0        0
```

Example - with slippage

```
> pos <- c(0,-1,-2,1,1,3,0,0)
> pnl <- run(prices,pos,slipMult=0.5)
> pnl
           Open Close Gap OpenDiff pos   pnl slippage
1970-01-02  100   112  NA       NA   0   0.0      0.0
1970-01-03  110   115  -2       10  -1   0.0      1.0
1970-01-04  120   110   5       10  -2 -11.0      2.5
1970-01-05  100   102 -10      -20   1  37.5     15.0
1970-01-06   95   100  -7       -5   1 -20.0      0.0
1970-01-07  105   111   5       10   3  10.0      5.0
1970-01-08  105   101  -6        0   0  -5.0      9.0
1970-01-09  110   100   9        5   0  -9.0      0.0
```

Commissions/transaction costs

- These are costs of trading paid to a broker or exchange
- A realistic model would include these too, however unlike including slippage, it is more clearcut how to incorporate transaction costs - just follow cost specification