FE8828 Programming Web Applications in Finance

Final Assignment - Supplementary material

Dr. Yang Ye <Email:yy@runchee.com>

Nov 26, 2018

1. Bank management

- Design of the data frames

  ```
  Data frame 1: Account
  | AcountNo | Name | Credit |

  Data frame 2: Transaction
  | TransactionNo | Date | AccountNo | TransactionType | Amount | Currency |

  Data frame 3: Currency to SGD
  | Currency | Conversion | Date |
  ```

- There are three kinds of TransctionType: Deposit/Withdraw/Spend. Amount is of sign +/-/- respectively.
- Deposit / Withdraw is paired up in a way that you can't withdraw more than deposit.
- Credit / Spend is paired up in a way that you can't spend more than credit.
- Assume credit can last from one quarter, 2018Q3: 2018-07-01 to 2018-09-30.
- Example Sections in the solution
  - Generate random test data in three data frames.
  - Pick one random AcountNo to show the monthly statement.
  - Pick a random date between 2017-07-01 and 2017-09-30 to do
    - Risk department:
      - Total balance daily
      - Total receivable from credit daily
      - Top 10 High and low-risk client (i.e. balance - spent).
  - Show what's by the end of the period, on the end of day of 2017-09-30.
    - Customer department:
      - Top 10 customer with large balance (deposit - withdraw)
      - Top 10 spending customer (spend)
      - Top 10 saving customer (deposit)
    - Treasury department:
      - Interest that all customers need to pay for three months
      - Assume annual interest is 0.25%
      - Interest starts when customer spends.

---

2. Delta Hedging

Show case how delta hedging works as a trading strategy.

- Show how one trade works
- Backtest for available history - one-year history from the website.

Assumption:

- You hold 100 ATM call/put option which expires in 30 days (calendar days). You just need to do either Call or Put. After finishing the tasks below, duplicate the R Notebook and carry out analysis using an option of ~40% delta, take a guess for the strike level.
- You start to do delta hedging daily immediately till 2nd last day. You close stock position in the last day. Option expires on the last day as well (either ITM with positive PnL or ATM/OTM with zero PnL)
- Delta hedging: calculate the delta from option, negate it, that's the quantity what you need to hold over 1 day. Repeat for every trading day.
- Daily PnL: (option premium change) + (stock holding quantity * price change).
- You can get your favorite stocks here. There is one year of data.
  - <https://marketchameleon.com/Overview/*Stock Code*/DailyHistory/>
  - e.g.: https://marketchameleon.com/Overview/GS/DailyHistory/
- Daily IV30 is provided, IV 30 is implied volatility for 30 days. We flat-extrapolate it to be for < 30 days.
- As underlying is equity, dividend yield is applicable for B-S valuation
- US risk-free rate for 1M: 0.8% (annualized)

Analysis: For one trade

- Daily PnL v.s. Time to expiry: split into option and stock.

- Final PnL: accumulative of Daily PnL. split into option and stock
- Max Drawdown: accumulative of Daily PnL, max - min.
- Sharpe ratio: Sharpe ratio = (Mean of Daily PnL - Risk-free rate)/Standard deviation of Daily PnL

Analysis: For backtest

- Distribution of Final PnL
- Distribution of Max Drawdown
- Final PnL v.s. Option Expiry Date
- . . .
- More analysis
- in R Markdown

Sample Code:

- Create xts object from the data from website.
- One trade analysis
  - Pick a date range using xts object.
  - Get starting date and end date.

    ```
    dates <- index(xts_obj)
    start_date <- min(dates)
    end_date <- max(dates)
    start_price <- xts_obj[start_date, "Close"]
    start_volatility <- xts_obj[start_date, "IV30"]
    ```

  - create a df with date column

    ```
    df <- tibble(date = dates)
    df$Close <- coredata(xts_obj[, "Close"])
    ```

  - Daily Profit and Loss ("DoD PnL")

    - Option side:

      ```
      X <- start_price
      sigma = start_volatility
      r <- 0.8 / 100
      # Vary S and Time everyday
      S <- Close
      Time <- (end_date - date) / 365
      GBSOption(TypeFlag, S, X, Time, r, b, sigma)@price

      df_opt <- rowwise(df) %>%
                mutate(premium = GBSOption(TypeFlag = "...",
                                           S = Close,
                                           X = start_price,
                                           Time = (end_date - date) / 365,
                                           r = ..., # interest rate
                                           b = ..., # dividend yield
                                           sigma = start_volatility)@price) %>%
                ungroup %>%
                mutate(Option_DoD_PnL = ifelse(date == start_date,
                # On the 1st date, we count the cost of buying the option
                                               premium * (-1),
                                               premium - lag(premium)))
      ```

    - Hedging side:

      ```
      rowwise() %>%
      mutate(delta_hedge = GBSGreeks("delta", TypeFlag, S, X, Time, r, b, sigma) *
                           quantity * (-1)) %>%
      ungroup() %>%
      mutate(Hedging_DoD_Pnl = ifelse(date == start_date,
                                      0,
                                      delta_hedge * (Close - lag(Close))))
      ```

    - Daily PnL (combined):

      ```
      mutate(DoD_PnL = Option_DoD_PnL + Hedging_DoD_Pnl)
      ```

- Max Drawdown: accumulative of Daily PnL, max - min.

  ```
  ungroup() %>%
  mutate(PnL = cumsum(DoD_PnL)) %>%
  {
    xs <- .$PnL
    max(cummax(xs) - cummin(xs))
  }
  ```