

FE8828 Programming Web Applications in Finance - Assignment

Dr. Yang Ye yy@runchee.com

Oct 12, 2020

1. Bank

1. Create 10 client accounts with initial random deposit (random in [1000, 2000]) in SGD and credit (2000 for all) in SGD. Store it to a Rda file (Rda: "R data file", use `save()` and `load()`).

```
Data frame 1: Account
| AccountNo | Name |
```

2. Create 3 currencies: CNY, USD, SGD. Download their conversion rate during 2020-07-01 and 2020-09-30 from MAS website and store in Rda. <https://eservices.mas.gov.sg/Statistics/msb/ExchangeRates.aspx>

```
Data frame 2: Currency to SGD
| Currency | Conversion | Date |
```

3. Generate random transaction data for 10 accounts during 2020-07-01 and 2020-09-30 and store them in Rda.
 - **Deposite:** Make it more realistic, deposit is 1 or 2 times per month, with an amount of random number between 1000-2000, in SGD.
 - **Spend/Withdraw** can be any times [0, 1000] on random days on random amount (could be a random number from a long-tail distribution, e.g. `rlnorm(1,0,3)`), in any of three currencies
 - Deposit is positive, Withdraw/Spend is negative.
 - Spend increases from credit. If it is non-SGD, bank charges additional 2%. If it is SGD, bank charges additional 1%.
 - Withdraw deducts from balance
 - **Interest payment** from bank for end-of-month balance, at 0.5% interest rate (annualized)
 - **Constraints for data generation:** You can't withdraw more than the deposit, can't spend more than credit.

TransactionNo	Date	AccountNo	TransactionType	Amount	Currency
---------------	------	-----------	-----------------	--------	----------

TransactionType is one of Withdraw/Deposit/Spend/Interest

4. Build a Shiny App: reads the autogenerated data files and generate following reports.
5. Client view: Month-end statement generation with following input - Select client account - Select month - Output as below:

```
{ Client Name }
{ Month }
```

Chart of balance: showing credit, deposit, balance along the axis of date

Transaction History

Date	TransactionType	Currency	Amount	Amount (in SGD)	Deposit Balance	Credit Balance
Month-End Balance					Deposit	Credit

Summary

TransactionType		Amount	
...			

2. Bank view, Month report with following input

- Select month
- Output as below:

Chart: showing aggregated total deposit and credit along the axis of date

PnL table: Detail table (for every day)

Date		Total Deposit		Total Credit		PnL from Client Spending	
....							

Risk table as of the end of month (sort descending by largest Credit-Deposit)

Client Name		Deposit		Credit	
...					

2. Option trading - dynamic hedging

Financial derivatives like Options are named as so because their value is derived from its underlyings. Greeks are the sensitivity of the price of the derivatives to its underlying parameters. For example, a vanilla European option depends on S (Underlying price), $striKe$, Vol (atility), r (interest rate), q (cost of carry), T (time).

Greeks:

- $\frac{dV}{dS}$: Delta
- $\frac{d^2V}{dS^2}$: Gamma
- $\frac{dV}{dVol}$: Vega
- $\frac{dV}{dT}$: Theta

To simplify, we set $r = q = 0$ (so Rho is zero).

The daily PnL of option can be decomposed into, according to Taylor expansion

$$\Delta V \approx DeltaPnL + GammaPnL + VegaPnL + ThetaPnL$$

$$\Delta V \approx Delta * \Delta S + 0.5 * Gamma * \Delta S^2 + Vega * \Delta Vol + Theta * (dT)$$

Option trading with dynamic hedging

- When trader sees that the implied volatility from option is lower than the estimated realized volatility, he/she may start to take a long position in option and a short position in the underlying.
- The position on option is established at the beginning of the strategy. Trader pays the option premium as the cost for the trade.
- The position in underlying is adjusted periodically (e.g. daily) to the opposite of the option delta so the strategy-delta is zero at end-of-day.
- Trader's objective is to realize the difference between realized volatility and implied volatility.
- If the realized volatility is higher than implied volatility, trader earns the positive profit. If realized volatility is lower than the implied volatility, trader takes a loss for the market.

For example, trader bought 20-day ATM call option, $K = S = 100$, $r = q = 0$, $vol = 0.3$ constant till option expiry.

On Day 0, end of day.

```

library(conflicted)
library(tidyverse)
library(fOptions)
suppressMessages({
  conflict_prefer("filter", "dplyr")
  conflict_prefer("lag", "dplyr")
})

days <- 20.0
vega0 <- GBSGreeks("Vega", "c", S = 100, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3)
delta0 <- GBSGreeks("Delta", "c", S = 100, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3)
gamma0 <- GBSGreeks("Gamma", "c", S = 100, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3)
theta0 <- GBSGreeks("Theta", "c", S = 100, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3)
price0 <- GBSOption("c", S = 100, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3)@price

```

On day 1, if $S_1 = 98$, volatility is unchanged.

```

S1 <- 98
option_pnl <- GBSOption("c", S = S1, X = 100,
  Time = (days - 1) / 250, r = 0, b = 0, sigma = 0.3)@price - price0
approx_pnl <- delta0 * (S1 - 100) + 0.5 * gamma0 * (S1 - 100) ** 2 + vega0 * 0 + theta0 * 1/250
cat(paste0("option_pnl: ", option_pnl, ", ", approx_pnl, "\n"))

```

```
## option_pnl: -1.02175324971442, -1.02444665039741
```

```
cat(paste0("diff with approx.: ", option_pnl - approx_pnl, "\n"))
```

```
## diff with approx.: 0.00269340068298818
```

S is to be changed daily, below code generates random sequence of S for 20 days, given starting S and *volatility*.

```

S <- 100
K <- 100
sigma <- 0.3 # realized vol can be 0.3 for [2] or 0.5 for [3]
drift <- 0 # drift = r - q
timestep <- 1 / 250
days <- 20 / 250
N <- days / timestep

p1 <- (drift - 0.5 * sigma * sigma) * timestep
p2 <- sigma * sqrt(timestep)

# ss is the simulated price movement for N days
ss <- rep(S, N) * c(cumprod(rlnorm(N, mean = p1, sd = p2)))
# We can calculate the volatility as below from ss
sigma_2 <- sd((log(ss / lag(ss))[-1])) / sqrt(1 / 250)

```

Questions:

Your task is to run 1000 simulations with random-generated price series with a 20-day ATM call option. Collect data to show below three points

[1] Taylor's expansion to actual PnL with indeed small remainder.

[2] If realized volatility and implied volatility are the same, Gamma and Theta can neutralize each other, i.e. $\Sigma(GammaPnL + ThetaPnL) \approx 0$

[3] If realized volatility > implied volatility, there is positive "Overall PnL" (profit and loss).

Note 1: that simulation would show some acceptable level of difference. Note 2: The overall PnL = the cost of buying the option + final payoff + delta PnL.

```
df <- tibble(S = ss, days = 1:days)
opt <- rowwise(df) %>% mutate(
  price = GBSOption("c", S = S, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3)@price,
  delta = GBSGreeks("Delta", "c", S = S, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3),
  gamma = GBSGreeks("Gamma", "c", S = S, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3),
  theta = GBSGreeks("Theta", "c", S = S, X = 100, Time = days / 250, r = 0, b = 0, sigma = 0.3) / 250
)
...
```

3. Portfolio Investment

See “port_invest.Rmd”