DI2C01 Final Assessment

Justin

10 March 2023

Importing the necessary libraries

```
pacman::p_load(knitr, tidyverse, simmer, simmer.plot, triangle, fitdistrplus, vcd)
```

Story: Stock Portfolio Scenario

Suppose we want to start investing to increase our capital. Luckily, we took the DLP2c course and we can simulate our portfolio before really buying into stocks. For instance, this portfolio simulation runs for a period of 2 years, and we have gathered enough information about the monthly stock movement and compute the appropriate probability distribution. Take note that there is no resource involved in our simulation.

Global Variables

We have several global variables that we wish to keep track while running the simulation. The following global variables defined below are in the units of dollar:

- `capital` - the capital we allocated for investment
- `total_stocks` - the total amount of stock that we owned
- `dividend` - the total amount of dividend we gained from holding the stocks
- `inflow` - the amount of money needed (on a monthly basis) to top up our capital

Entity (Months)

The entity in our simulation will be the "monthly review" of the portfolio. Every month, our `capital` will be topped up by `inflow`. If the `total_stocks` value is 0 or negative, we will just use the capital to buy stocks. Otherwise, there is a 10% chance that we will sell our stocks to gain some profit.

(Of course, if you are an avid investor and know trading well, you can always change the proportion of buying and selling by yourself!)

Attribute Type 1 (Buy)

The amount of stocks that we wish to **buy** is estimated to be triangularly distributed from **0** to **100** shares, with a peak at **10** shares, multiply with the stock price, which is estimated to be normally distributed with a mean of **$50** and a sd of **$10**.

Attribute Type 2 (Sell)

The amount of stocks that we wish to **sell** is estimated to be triangularly distributed from **0** to **100** shares, with a peak at **10** shares, multiply with the stock price, which is estimated to be normally distributed with a mean of **$60** and a sd of **$15**.

We will then collect some dividends based on how many stocks we currently owned. The annual yield is estimated to be uniformly distributed from **4%** to **6%**. After that, we also need to keep track of the monthly gain or loss from the stocks we owned. It is estimated to be triangularly distributed from a loss of **10%** to a gain of **10%**, with a peak of a gain at **1%**.

Finally, if we sum up `capital`, `total_stocks` and `dividend` and the value is more than twice the amount of our initial capital, we have 2x our capital and we are happy to sell away all our stocks in that particular month. On the contrast, if our `capital` is negative in value (meaning we bought too many stocks until we have deficit), the broker will force to sell away all our stocks, as we are not allowed to borrow money to buy stocks.

The simulation will run at a "monthly" basis for a period of 2 years (24 months).

Q13: Draw the Activity Diagram [8 marks]

Please draw the diagram using the A4 paper provided.

To assist you, we have prepared you with the complete code of the following simulation. Please run through the whole chunk, and read the (outcome) logs carefully to better understand the model.

```
portfolio_env <- simmer() %>%
    add_global("capital", 10000) %>% # Let us start with capital $10000
    add_global("total_stocks", 0) %>%
    add_global("dividend", 0) %>%
    add_global("inflow", 1000) # We wish to top up our capital $1000 monthly

get_inflow <- function() get_global(portfolio_env, "inflow")
get_stocks <- function() get_global(portfolio_env, "total_stocks")

m4 <- trajectory("portfolio") %>%
  log_("Begin buying/selling my stocks..") %>%
  branch(function() {
    if(get_global(portfolio_env, "total_stocks") <= 0){
      return(1)
      } else {
        sample(2,size=1, prob=c(0.9,0.1))
      }
    }, continue = TRUE,
    trajectory() %>%
      set_attribute("Type", 1) %>%
      log_("Buying into stocks using my capital") %>%
      set_attribute("stocks_bought",
                    function() rtriangle(1,0,100,10)*rnorm(1,50,10)) %>%
      set_global("capital",
                 function() -get_attribute(portfolio_env, "stocks_bought"), mod="+") %>%

      set_global("total_stocks",
                 function() get_attribute(portfolio_env, "stocks_bought"), mod="+"),
    trajectory() %>%
```

```
    set_attribute("Type", 2) %>%
    log_("Selling my stocks..") %>%
    set_attribute("stocks_sold",
                  function() rtriangle(1,0,100,10)*rnorm(1,60,15)) %>%
    branch(function() {
      stocks <- c(get_global(portfolio_env,"total_stocks"),
                  get_attribute(portfolio_env, "stocks_sold"))
      if(stocks[1] >= stocks[2]){
        return (1)
        } else{
          return (2)
        }
      }, continue= TRUE,
      trajectory() %>%
        set_global("total_stocks",
                   function() -get_attribute(portfolio_env, "stocks_sold"), mod="+") %>%
        set_global("capital",
                   function() get_attribute(portfolio_env, "stocks_sold"), mod="+"),
      trajectory() %>%
        log_("Invalid transaction, not enough stocks owned to sell")
      )
  ) %>%
log_("This is the monthly dividend I get..") %>%
set_global("dividend",
           function() get_global(portfolio_env, "total_stocks")*runif(1,0.04,0.06)/12, mod="+") %>%
log_("Lets see what is my gain/loss from the stocks this month") %>%
set_global("total_stocks",
           function() get_global(portfolio_env, "total_stocks")*rtriangle(1,0.90,1.10,1.01)) %>%
set_global("capital", get_inflow, mod="+") %>%
branch(function() {
  attrib <- get_global(portfolio_env,
                       c("capital","total_stocks","dividend"))
  if(sum(attrib) >= 2*initial_cap){
    return(1)
    } else if(attrib[1] < 0){
      return(2)
    } else{
      return(0)
    }
  }, continue = FALSE,
  trajectory() %>%
    log_("Hooray! Ive gained 100% from my initial capital! I am going to sell!") %>%
    set_attribute("selling-type", 1) %>%
    set_global("capital", get_stocks, mod="+") %>%
    set_global("total_stocks", 0) %>%
    set_global("capital", get_inflow, mod="+"),
  trajectory() %>%
    log_("I have negative balance on my capital, hence I am forced to sell all my stocks!") %>%
    set_attribute("selling-type", 2) %>%
    set_global("capital", get_stocks, mod="+") %>%
    set_global("total_stocks", 0) %>%
    set_global("capital", get_inflow, mod="+")
  ) %>%
```

```
  log_("I have done my review for this month!")

set.seed(123)
portfolio_env %>%
  add_generator("month", m4, at(c(1:24)), mon=2) %>%
  run(until=25)
```

Q14: Set Up Monthly Trajectory and Build the Simulation Model [12 marks]

We wrap up our simulation model, which correspond to the earlier scenario with all the numbers/rates fixed, to ease our running of the code.

Suppose we are free to choose the initial capital value, `initial_cap` for investment, as well as the monthly contribution to the capital, `inflow`.

```
set.seed(123)
portfolio <- function(initial_cap, inflow){

  portfolio_env <- simmer() %>%
    add_global("capital", initial_cap) %>%
    add_global("total_stocks", 0) %>%
    add_global("dividend", 0) %>%
    add_global("inflow", inflow)

  get_inflow <- function() get_global(portfolio_env, "inflow")
  get_stocks <- function() get_global(portfolio_env, "total_stocks")

  m4 <- trajectory("portfolio") %>%
    log_("Begin buying/selling my stocks..") %>%
    branch(function() {
      if(get_global(portfolio_env, "total_stocks") <= 0){
        return(1)
        } else {
          sample(2,size=1, prob=c(0.9,0.1))
        }
      }, continue = TRUE,
      trajectory() %>%
        set_attribute("Type", 1) %>%
        log_("Buying into stocks using my capital") %>%
        set_attribute("stocks_bought",
                      function() rtriangle(1,0,100,10)*rnorm(1,50,10)) %>%
        set_global("capital",
                   function() -get_attribute(portfolio_env, "stocks_bought"), mod="+") %>%
        set_global("total_stocks",
                   function() get_attribute(portfolio_env, "stocks_bought"), mod="+"),
      trajectory() %>%
        set_attribute("Type", 2) %>%
        log_("Selling my stocks..") %>%
        set_attribute("stocks_sold",
                      function() rtriangle(1,0,100,10)*rnorm(1,60,15)) %>%
        branch(function() {
          stocks <- c(get_global(portfolio_env,"total_stocks"),
                      get_attribute(portfolio_env, "stocks_sold"))
```

```
          if(stocks[1] >= stocks[2]){
            return (1)
            } else{
              return (2)
            }
          }, continue= TRUE,
          trajectory() %>%
            set_global("total_stocks",
                       function() -get_attribute(portfolio_env, "stocks_sold"), mod="+") %>%
            set_global("capital",
                       function() get_attribute(portfolio_env, "stocks_sold"), mod="+"),
          trajectory() %>%
            log_("Invalid transaction, not enough stocks owned to sell")
          )
      ) %>%
    log_("This is the monthly dividend I get..") %>%
    set_global("dividend",
               function() get_global(portfolio_env, "total_stocks")*runif(1,0.04,0.06)/12, mod="+") %>%
    log_("Lets see what is my gain/loss from the stocks this month") %>%
    set_global("total_stocks",
               function() get_global(portfolio_env, "total_stocks")*rtriangle(1,0.90,1.10,1.01)) %>%
    set_global("capital", get_inflow, mod="+") %>%
    branch(function() {
      attrib <- get_global(portfolio_env,
                           c("capital","total_stocks","dividend"))
      if(sum(attrib) >= 2*initial_cap){
        return(1)
        } else if(attrib[1] < 0){
          return(2)
        } else{
          return(0)
        }
      }, continue = FALSE,
      trajectory() %>%
        log_("Hooray! Ive gained 100% from my initial capital! I am going to sell!") %>%
        set_attribute("selling-type", 1) %>%
        set_global("capital", get_stocks, mod="+") %>%
        set_global("total_stocks", 0) %>%
        set_global("capital", get_inflow, mod="+"),
      trajectory() %>%
        log_("I have negative balance on my capital, hence I am forced to sell all my stocks!") %>%
        set_attribute("selling-type", 2) %>%
        set_global("capital", get_stocks, mod="+") %>%
        set_global("total_stocks", 0) %>%
        set_global("capital", get_inflow, mod="+")
      ) %>%
    log_("I have done my review for this month!")


  portfolio_env %>%
    add_generator("month", m4, at(c(1:24)), mon=2) %>%
    run(until=25) %>% wrap()
}
```

Q14 Open Question

There is no fixed answer. You could mimic Video 7 & 8 materials, as well as WS4 to try some of the following tasks: (optional)

(1) Explain the choice of your input `initial_cap` and `inflow` and try to get the best possible outcome
(2) Extracting one or some performance measures;
(3) Running replications;
(4) Computing confidence interval(s);
(5) Make modifications: the main objective is to modify the model in order to improve the performance;
(6) How can we make this model more realistic, based on some domain knowledge? (qualitatively or quantitatively)

Some Hints: (they are all optional!)

(h1) You may use `get_mon_attributes` to monitor the data. Do you know why `get_mon_arrivals` and `get_mon_resources` are not very meaningful for this model?

```
get_attribute <- get_mon_attributes(portfolio(10000,2000)) # test case of initial_cap = 10000, inflow =

get_attribute %>%
  filter(key %in% c("capital","total_stocks","dividend","inflow")) %>%
  group_by(time, key) %>%
  summarise(sum = sum(last(value, order_by=key))) -> test

# ?last() to find out more
```

What do you observe from the table above?

(h2) What other constraints can you put to the simulation so that neither the broker will force to sell your stocks, nor you will sell all your stocks once you gained 100% capital size?

(h3) If you have trouble in coding for this open question, another strategy is to express your understanding of the given codes and the outputs of the given codes. In other words, you can give comments and suggestions qualitatively. We will reward you accordingly!

Nevertheless, do have fun in this portfolio simulation that we specially designed for all of you. Happy investing (I mean, happy coding)

Thank you all for the consistent support!