FE515 2023 Assignment-3

2023-04-18

```
rm(list=ls())
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## ###################################### WARNING ######################################
## # We noticed you have dplyr installed. The dplyr lag() function breaks how    #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
## #                                                                              #
## # If you call library(dplyr) later in this session, then calls to lag(my_xts) #
## # that you enter or source() into this session won't work correctly.           #
## #                                                                              #
## # All package code is unaffected because it is protected by the R namespace   #
## # mechanism.                                                                   #
## #                                                                              #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop    #
## # dplyr from breaking base R's lag() function.                                 #
## ###################################### WARNING ######################################
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
VIX <- getOptionChain("^VIX")
exp_dates <- unique(VIX$VIX$calls$exDates)
VIX_data <- list()
```

```r
for (exp_date in exp_dates) {
  options <- getOptionChain("^VIX", NULL, exp_date)
  VIX_data[[exp_date]] <- options
}
```

```r
# Save VIX_data as VIX.options
save(VIX_data, file = "VIX.options")

# Download the current price for ^VIX
VIX_price <- as.numeric(getQuote("^VIX")$Last)
```

```r
# Calculate average of Bid and Ask for calls and puts at each expiration
for (exp_date in exp_dates) {
  options <- VIX_data[[exp_date]]
  if (is.null(options)) {
    next
  }
  for (type in c("calls", "puts")) {
    option_type <- options[[type]]
    if (is.null(option_type)) {
      next
    }
    option_type$Price <- rowMeans(option_type[, c("bid", "ask")], na.rm = TRUE)
    options[[type]] <- option_type
  }
  VIX_data[[exp_date]] <- options
}
```

```r
# Add InTheMoney column to calls and puts at each expiration
for (exp_date in exp_dates) {
  options <- VIX_data[[exp_date]]
  for (type in c("calls", "puts")) {
    option_type <- options[[type]]
    option_type$InTheMoney <- ifelse(type == "calls", option_type$strike < VIX_price, option_type$strik
    options[[type]] <- option_type
  }
  VIX_data[[exp_date]] <- options
}
```

```r
# Save calls and puts data to CSV files
for (exp_date in exp_dates) {
  options <- VIX_data[[exp_date]]
  for (type in c("calls", "puts")) {
    option_type <- options[[type]][, c("strike", "bid", "ask", "Price", "InTheMoney")]
    file_name <- paste0("VIXdata", format(Sys.Date(), "%Y-%m-%d"), "Exp", exp_date, type, ".csv")
    write.csv(option_type, file_name, row.names = FALSE)
  }
}
```