Book Option Trade

William HU ZIHAO

2024-10-05

Step 1 Load Option Data

Replace *data_path* with the actual file path where the CSV is stored.

```r
# Load the CSV file into a dataframe
data_path <- "./book_option_trade.csv"
book_option_trade <- read_csv(data_path)
## Rows: 101 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Exp Date, Change, Volume, Change_put, Volume_put, Open Int_put
## dbl (8): Last, Bid, Ask, Open Int, Strike, Last_put, Bid_put, Ask_put
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
View(book_option_trade)
```

```r
# Check the structure of the dataframe
str(book_option_trade)
## spc_tbl_ [101 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Exp Date    : chr [1:101] "Nov-15" "Nov-15" "Nov-15" "Nov-15" ...
##  $ Last        : num [1:101] 61.5 60.5 56.7 58 58.3 ...
##  $ Change      : chr [1:101] "--" "--" "-3.54\xca" "0.99\xca" ...
##  $ Bid         : num [1:101] 62.2 61.3 60.4 59.4 58.5 ...
##  $ Ask         : num [1:101] 62.4 61.5 60.5 59.6 58.6 ...
##  $ Volume      : chr [1:101] "--" "--" "132" "1" ...
##  $ Open Int    : num [1:101] 5 8 54 15 6091 ...
##  $ Strike      : num [1:101] 516 517 518 519 520 521 522 523 524 525 ...
##  $ Last_put    : num [1:101] 2.19 2.27 2.36 2.35 2.44 2.48 2.58 2.58 2.67 2.73 ...
##  $ Change_put  : chr [1:101] "-0.81\xca" "-0.79\xca" "-0.83\xca" "-0.89\xca" ...
##  $ Bid_put     : num [1:101] 2.24 2.3 2.35 2.41 2.47 2.53 2.6 2.66 2.73 2.8 ...
##  $ Ask_put     : num [1:101] 2.26 2.32 2.37 2.43 2.48 2.55 2.62 2.68 2.75 2.82 ...
##  $ Volume_put  : chr [1:101] "287" "84" "236" "277" ...
##  $ Open Int_put: chr [1:101] "1800" "2763" "1291" "1375" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Exp Date` = col_character(),
##   ..   Last = col_double(),
##   ..   Change = col_character(),
##   ..   Bid = col_double(),
##   ..   Ask = col_double(),
##   ..   Volume = col_character(),
##   ..   `Open Int` = col_double(),
##   ..   Strike = col_double(),
##   ..   Last_put = col_double(),
##   ..   Change_put = col_character(),
##   ..   Bid_put = col_double(),
##   ..   Ask_put = col_double(),
##   ..   Volume_put = col_character(),
##   ..   `Open Int_put` = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>

# Preview the first few rows of the dataframe
head(book_option_trade)
## # A tibble: 6 x 14
##   `Exp Date`  Last Change        Bid   Ask Volume `Open Int` Strike Last_put
##   <chr>      <dbl> <chr>       <dbl> <dbl> <chr>       <dbl>  <dbl>    <dbl>
## 1 Nov-15      61.5 "--"         62.2  62.4 --              5    516     2.19
## 2 Nov-15      60.5 "--"         61.3  61.5 --              8    517     2.27
## 3 Nov-15      56.7 "-3.54\xca"  60.4  60.5 132            54    518     2.36
## 4 Nov-15      58.0 "0.99\xca"   59.4  59.6 1              15    519     2.35
## 5 Nov-15      58.3 "5.33\xca"   58.5  58.6 4            6091    520     2.44
## 6 Nov-15      51.9 "--"         57.6  57.7 --             27    521     2.48
## # i 5 more variables: Change_put <chr>, Bid_put <dbl>, Ask_put <dbl>,
## #   Volume_put <chr>, `Open Int_put` <chr>
```

Step 2 Data Cleaning

2.1 Data Type Conversion

```
library(lubridate)

book_option_trade_formatted <- book_option_trade %>%filter(rowSums(is.na(.)) != ncol(.))%>%
  mutate(`Exp Date` = dmy(paste0("15-", substr(`Exp Date`, 1, 3), "-2024"))) # Extract first 3 chars fo

book_option_trade_formatted <- book_option_trade_formatted %>%
  mutate(across(c(Change, Volume, Change_put, Volume_put, `Open Int_put`),
                ~replace(., . == "--", "0")))

# Generated by ChatGPT to remove all chars could not be interpretted by Unicode
book_option_trade_formatted <- book_option_trade_formatted %>%
  mutate(across(c(Change, Change_put),
                ~as.numeric(str_replace_all(., "[^0-9.-]", ""))))


book_option_trade_formatted <- book_option_trade_formatted %>%
  mutate(across(c(Last, Bid, Ask, `Open Int`, Strike, Last_put, Bid_put, Ask_put),
                as.numeric)) %>%
  mutate(across(c(Volume, Volume_put, `Open Int_put`), as.numeric))


str(book_option_trade_formatted)
## tibble [93 x 14] (S3: tbl_df/tbl/data.frame)
##  $ Exp Date    : Date[1:93], format: "2024-11-15" "2024-11-15" ...
##  $ Last        : num [1:93] 61.5 60.5 56.7 58 58.3 ...
##  $ Change      : num [1:93] 0 0 -3.54 0.99 5.33 0 0 0 0 3.36 ...
##  $ Bid         : num [1:93] 62.2 61.3 60.4 59.4 58.5 ...
##  $ Ask         : num [1:93] 62.4 61.5 60.5 59.6 58.6 ...
##  $ Volume      : num [1:93] 0 0 132 1 4 0 0 0 0 5 ...
##  $ Open Int    : num [1:93] 5 8 54 15 6091 ...
##  $ Strike      : num [1:93] 516 517 518 519 520 521 522 523 524 525 ...
##  $ Last_put    : num [1:93] 2.19 2.27 2.36 2.35 2.44 2.48 2.58 2.58 2.67 2.73 ...
##  $ Change_put  : num [1:93] -0.81 -0.79 -0.83 -0.89 -0.78 -0.92 -0.93 -1.06 -1 -0.92 ...
##  $ Bid_put     : num [1:93] 2.24 2.3 2.35 2.41 2.47 2.53 2.6 2.66 2.73 2.8 ...
##  $ Ask_put     : num [1:93] 2.26 2.32 2.37 2.43 2.48 2.55 2.62 2.68 2.75 2.82 ...
##  $ Volume_put  : num [1:93] 287 84 236 277 3198 ...
##  $ Open Int_put: num [1:93] 1800 2763 1291 1375 73091 ...
```

2.2 Columns Transformation

Data frame columns:

`| Exp. Date | Strike | OpenInterest | OptionType | Bid | Ask | Underlying | Today`

- Underlying/Today can be found on the top of the web page.
- OptionType is call for “Calls”, put for “Puts”
- OpenInterest means the total number of outstanding contracts that are held by the two sides of market participants.

```
# Create the call options data frame
call_options <- book_option_trade_formatted %>%
  select(
    `Exp Date`,
    Strike,
    OpenInterest = `Open Int`,
    Bid,
    Ask
  ) %>%
  mutate(OptionType = "call")

# Create the put options data frame
put_options <- book_option_trade_formatted %>%
  select(
    `Exp Date`,
    Strike,
    OpenInterest = `Open Int_put`,
    Bid = Bid_put,
    Ask = Ask_put
  ) %>%
  mutate(OptionType = "put")

call_options
## # A tibble: 93 x 6
##    `Exp Date` Strike OpenInterest   Bid   Ask OptionType
##    <date>      <dbl>        <dbl> <dbl> <dbl> <chr>
##  1 2024-11-15    516            5  62.2  62.4 call
##  2 2024-11-15    517            8  61.3  61.5 call
##  3 2024-11-15    518           54  60.4  60.5 call
##  4 2024-11-15    519           15  59.4  59.6 call
##  5 2024-11-15    520         6091  58.5  58.6 call
##  6 2024-11-15    521           27  57.6  57.7 call
##  7 2024-11-15    522           30  56.6  56.8 call
##  8 2024-11-15    523           30  55.7  55.8 call
##  9 2024-11-15    524           89  54.8  54.9 call
## 10 2024-11-15    525         2851  53.8  54   call
## # i 83 more rows
put_options
## # A tibble: 93 x 6
##    `Exp Date` Strike OpenInterest   Bid   Ask OptionType
##    <date>      <dbl>        <dbl> <dbl> <dbl> <chr>
##  1 2024-11-15    516         1800  2.24  2.26 put
##  2 2024-11-15    517         2763  2.3   2.32 put
##  3 2024-11-15    518         1291  2.35  2.37 put
##  4 2024-11-15    519         1375  2.41  2.43 put
##  5 2024-11-15    520        73091  2.47  2.48 put
##  6 2024-11-15    521         2002  2.53  2.55 put
##  7 2024-11-15    522         1719  2.6   2.62 put
##  8 2024-11-15    523         2898  2.66  2.68 put
##  9 2024-11-15    524         1212  2.73  2.75 put
## 10 2024-11-15    525        55430  2.8   2.82 put
## # i 83 more rows
```

```
bot_final <- bind_rows(call_options, put_options)%>%mutate(Underlying = 572.86, Today = dmy("05-10-2024
  mutate_if(is.numeric, as.double)

bot_final
## # A tibble: 186 x 8
##    `Exp Date` Strike OpenInterest   Bid   Ask OptionType Underlying Today
##    <date>      <dbl>        <dbl> <dbl> <dbl> <chr>           <dbl> <date>
##  1 2024-11-15    516            5  62.2  62.4 call             573. 2024-10-05
##  2 2024-11-15    517            8  61.3  61.5 call             573. 2024-10-05
##  3 2024-11-15    518           54  60.4  60.5 call             573. 2024-10-05
##  4 2024-11-15    519           15  59.4  59.6 call             573. 2024-10-05
##  5 2024-11-15    520         6091  58.5  58.6 call             573. 2024-10-05
##  6 2024-11-15    521           27  57.6  57.7 call             573. 2024-10-05
##  7 2024-11-15    522           30  56.6  56.8 call             573. 2024-10-05
##  8 2024-11-15    523           30  55.7  55.8 call             573. 2024-10-05
##  9 2024-11-15    524           89  54.8  54.9 call             573. 2024-10-05
## 10 2024-11-15    525         2851  53.8  54   call             573. 2024-10-05
## # i 176 more rows
```

Step 2 Option Value Calculation

2.1 Call Options Alone

```
call_value <- bot_final %>%
  filter(OptionType == "call") %>%          # Filter call options
  mutate(Value = OpenInterest * (Bid + Ask) / 2) %>%  # Calculate value
  summarise(TotalCallValue = sum(Value, na.rm = TRUE)) # Sum up the value
call_value
## # A tibble: 1 x 1
##   TotalCallValue
##            <dbl>
## 1       4894049.
```

2.2 Put Options Alone

```
put_value <- bot_final %>%
  filter(OptionType == "put") %>%          # Filter call options
  mutate(Value = OpenInterest * (Bid + Ask) / 2) %>%  # Calculate value
  summarise(TotalPutValue = sum(Value, na.rm = TRUE)) # Sum up the value
put_value
## # A tibble: 1 x 1
##   TotalPutValue
##           <dbl>
## 1      4385312.
```

2.3 Call & Put Options

```
option_value <- bot_final %>%
  mutate(Value = OpenInterest * (Bid + Ask) / 2) %>%  # Calculate value
  summarise(TotalOptionValue = sum(Value, na.rm = TRUE)) # Sum up the value
option_value
## # A tibble: 1 x 1
##   TotalOptionValue
##              <dbl>
## 1         9279361.
```

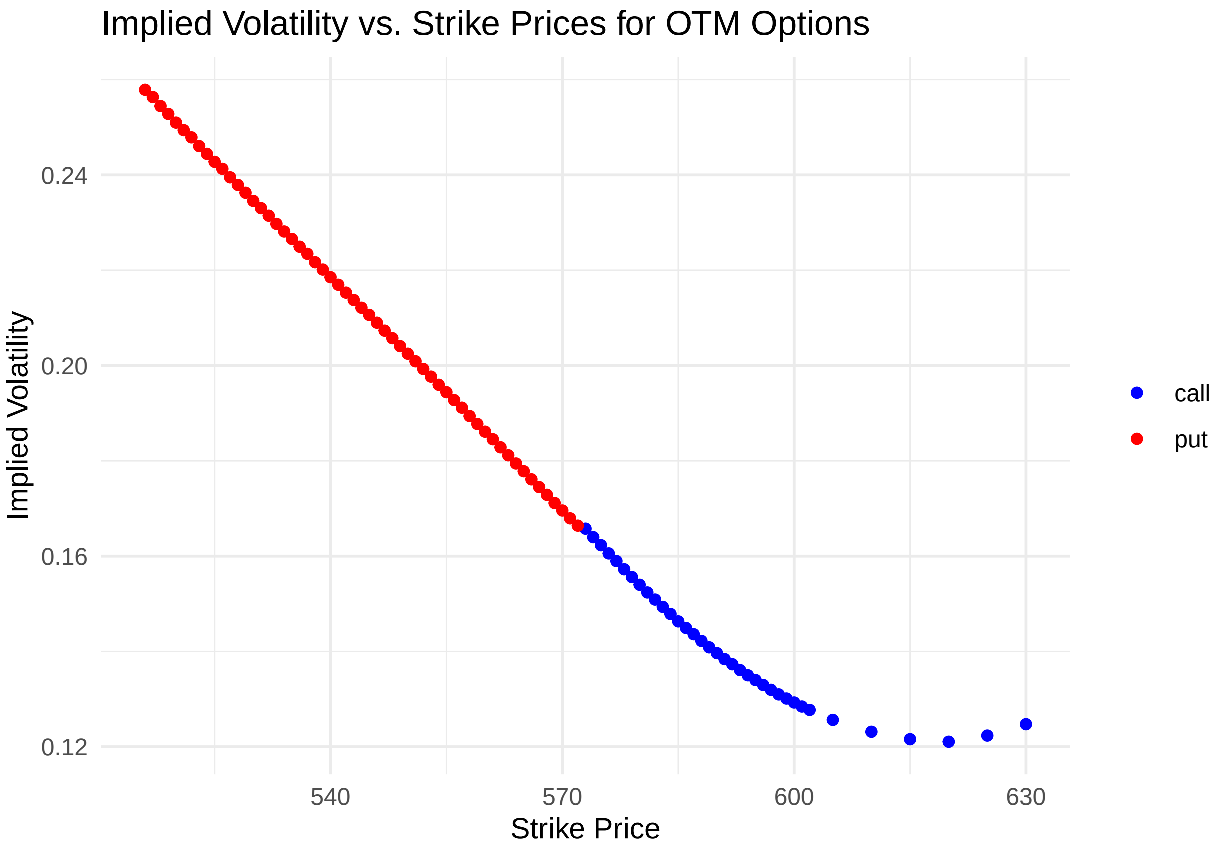
Step 3 Open Interest of in Money and Out Money

```
# Calculate total Open Interest for ITM and OTM for both Call and Put options in the same table
itm_otm_summary <- bot_final %>%
  summarise(
    TotalOpenInterest_ITM = sum(OpenInterest[(OptionType == "call" & Strike < Underlying) |
                                             (OptionType == "put" & Strike > Underlying)], na.rm = TRUE
    TotalOpenInterest_OTM = sum(OpenInterest[(OptionType == "call" & Strike > Underlying) |
                                             (OptionType == "put" & Strike < Underlying)], na.rm = TRUE
  )

# View result
itm_otm_summary
## # A tibble: 1 x 2
##   TotalOpenInterest_ITM TotalOpenInterest_OTM
##                   <dbl>                 <dbl>
## 1                150463                909584
```

Step 4 Plotting of Implied Volatility vs. Strike for OTM options

```
library(NMOF)

# Define rate and today
rate <- 0.05
today <- as.Date("2024-10-05")  # Data retrieved on 05/10/2024
tau <- as.numeric((as.Date("2024-11-15") - today)) / 365
```

```r
otm_options <- bot_final %>%
  filter((OptionType == "call" & Strike > Underlying) |
         (OptionType == "put" & Strike < Underlying))

# Calculate implied volatility
otm_options <- otm_options %>%
  rowwise() %>%
  mutate(ImpliedVolatility = vanillaOptionImpliedVol(
    exercise = "european",
    price = (Bid + Ask) / 2,
    S = Underlying,
    X = Strike,
    tau = tau,
    r = rate,
    q = 0,
    type = OptionType
  )) %>%
  ungroup()

ggplot(otm_options, aes(x = Strike, y = ImpliedVolatility, color = OptionType)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Implied Volatility vs. Strike Prices for OTM Options",
       x = "Strike Price",
       y = "Implied Volatility") +
  scale_color_manual(values = c("call" = "blue", "put" = "red")) +
  theme(legend.title = element_blank())
```



Step 5 Plotting of Implied Volatility vs. Strike for OTM options from Yahoo Finance

5.1 Load Option Data from Yahoo

```
library(readr)
book_option_trade_yahoo <- read_csv("./book_option_trade_yahoo.csv")
## Rows: 167 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (13): Exp Date, Last, Change, Bid, Ask, Volume, Open Int, Last_put, Chan...
## dbl  (1): Strike
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
View(book_option_trade_yahoo)
```

```
# Check the structure of the dataframe
str(book_option_trade_yahoo)
## spc_tbl_ [167 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Exp Date    : chr [1:167] "Nov-15" "Nov-15" "Nov-15" "Nov-15" ...
##  $ Last        : chr [1:167] "314.69" "308.75" "303.99" "299.21" ...
##  $ Change      : chr [1:167] "5.88" "0.00" "0.00" "0.00" ...
##  $ Bid         : chr [1:167] "314.08" "309.11" "304.15" "299.19" ...
##  $ Ask         : chr [1:167] "315.24" "310.27" "305.30" "300.33" ...
##  $ Volume      : chr [1:167] "93.00" "2.00" "4.00" "3.00" ...
##  $ Open Int    : chr [1:167] "104.00" "17.00" "4.00" "29.00" ...
##  $ Strike      : num [1:167] 260 265 270 275 280 285 290 295 300 305 ...
##  $ Last_put    : chr [1:167] "0.05" "0.05" "0.06" "0.08" ...
##  $ Change_put  : chr [1:167] "-0.01" "-0.02" "-0.02" "0.00" ...
##  $ Bid_put     : chr [1:167] "0.04" "0.05" "0.05" "0.06" ...
##  $ Ask_put     : chr [1:167] "0.06" "0.06" "0.06" "0.07" ...
##  $ Volume_put  : chr [1:167] "360.00" "92.00" "3.00" "5.00" ...
##  $ Open Int_put: chr [1:167] "2570.00" "11742.00" "14978.00" "4862.00" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Exp Date` = col_character(),
##   ..   Last = col_character(),
##   ..   Change = col_character(),
##   ..   Bid = col_character(),
##   ..   Ask = col_character(),
##   ..   Volume = col_character(),
##   ..   `Open Int` = col_character(),
##   ..   Strike = col_double(),
##   ..   Last_put = col_character(),
##   ..   Change_put = col_character(),
##   ..   Bid_put = col_character(),
##   ..   Ask_put = col_character(),
##   ..   Volume_put = col_character(),
##   ..   `Open Int_put` = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>

# Preview the first few rows of the dataframe
head(book_option_trade_yahoo)
## # A tibble: 6 x 14
##   `Exp Date` Last   Change Bid    Ask    Volume `Open Int` Strike Last_put
##   <chr>      <chr>  <chr>  <chr>  <chr>  <chr>  <chr>       <dbl> <chr>
## 1 Nov-15     314.69 5.88   314.08 315.24 93.00  104.00        260 0.05
## 2 Nov-15     308.75 0.00   309.11 310.27 2.00   17.00         265 0.05
## 3 Nov-15     303.99 0.00   304.15 305.30 4.00   4.00          270 0.06
## 4 Nov-15     299.21 0.00   299.19 300.33 3.00   29.00         275 0.08
## 5 Nov-15     293.88 0.00   294.23 295.36 13.00  21.00         280 0.07
## 6 Nov-15     284.36 0.00   289.26 290.40 4.00   294.00        285 0.08
## # i 5 more variables: Change_put <chr>, Bid_put <chr>, Ask_put <chr>,
## #   Volume_put <chr>, `Open Int_put` <chr>
```

5.2 Data Cleaning

```
library(lubridate)
book_option_trade_yahoo_formatted <- book_option_trade_yahoo %>%filter(rowSums(is.na(.)) != ncol(.))%>%
  mutate(`Exp Date` = dmy(paste0("15-", substr(`Exp Date`, 1, 3), "-2024"))) # Extract first 3 chars fo

book_option_trade_yahoo_formatted <- book_option_trade_yahoo_formatted %>%
  mutate(across(c(Last, Change, Bid, Ask, Volume, `Open Int`, Strike,
                 Last_put, Change_put, Bid_put, Ask_put, Volume_put, `Open Int_put`),
                ~replace(., . == "--", "0")))
```

```
# Generated by ChatGPT to remove all chars could not be interpretted by Unicode
book_option_trade_yahoo_formatted <- book_option_trade_yahoo_formatted %>%
  mutate(across(c(Change, Change_put),
                ~as.numeric(str_replace_all(., "[^0-9.-]", ""))))

book_option_trade_yahoo_formatted <- book_option_trade_yahoo_formatted %>%
  mutate(across(c(Last, Bid, Ask, `Open Int`, Strike, Last_put, Bid_put, Ask_put),
                as.numeric)) %>%
  mutate(across(c(Volume, Volume_put, `Open Int_put`), as.numeric))


str(book_option_trade_yahoo_formatted)
## tibble [167 x 14] (S3: tbl_df/tbl/data.frame)
##  $ Exp Date    : Date[1:167], format: "2024-11-15" "2024-11-15" ...
##  $ Last        : num [1:167] 315 309 304 299 294 ...
##  $ Change      : num [1:167] 5.88 0 0 0 0 0 0 0 0 0 ...
##  $ Bid         : num [1:167] 314 309 304 299 294 ...
##  $ Ask         : num [1:167] 315 310 305 300 295 ...
##  $ Volume      : num [1:167] 93 2 4 3 13 4 2 1 1 14 ...
##  $ Open Int    : num [1:167] 104 17 4 29 21 294 341 530 574 554 ...
##  $ Strike      : num [1:167] 260 265 270 275 280 285 290 295 300 305 ...
##  $ Last_put    : num [1:167] 0.05 0.05 0.06 0.08 0.07 0.08 0.08 0.11 0.1 0.1 ...
##  $ Change_put  : num [1:167] -0.01 -0.02 -0.02 0 -0.02 -0.02 -0.02 0 -0.02 -0.03 ...
##  $ Bid_put     : num [1:167] 0.04 0.05 0.05 0.06 0.07 0.07 0.08 0.08 0.09 0.1 ...
##  $ Ask_put     : num [1:167] 0.06 0.06 0.06 0.07 0.08 0.08 0.09 0.09 0.1 0.11 ...
##  $ Volume_put  : num [1:167] 360 92 3 5 2 16 45 16 130 55 ...
##  $ Open Int_put: num [1:167] 2570 11742 14978 4862 3026 ...
View(book_option_trade_yahoo_formatted)
```

5.2.1 Data Type Conversion

```
# Create the call options data frame
call_options_yahoo <- book_option_trade_yahoo_formatted %>%
  select(
    `Exp Date`,
    Strike,
    OpenInterest = `Open Int`,
    Bid,
    Ask
  ) %>%
  mutate(OptionType = "call")

# Create the put options data frame
put_options_yahoo <- book_option_trade_yahoo_formatted %>%
  select(
    `Exp Date`,
    Strike,
    OpenInterest = `Open Int_put`,
    Bid = Bid_put,
    Ask = Ask_put
  ) %>%
  mutate(OptionType = "put")
```

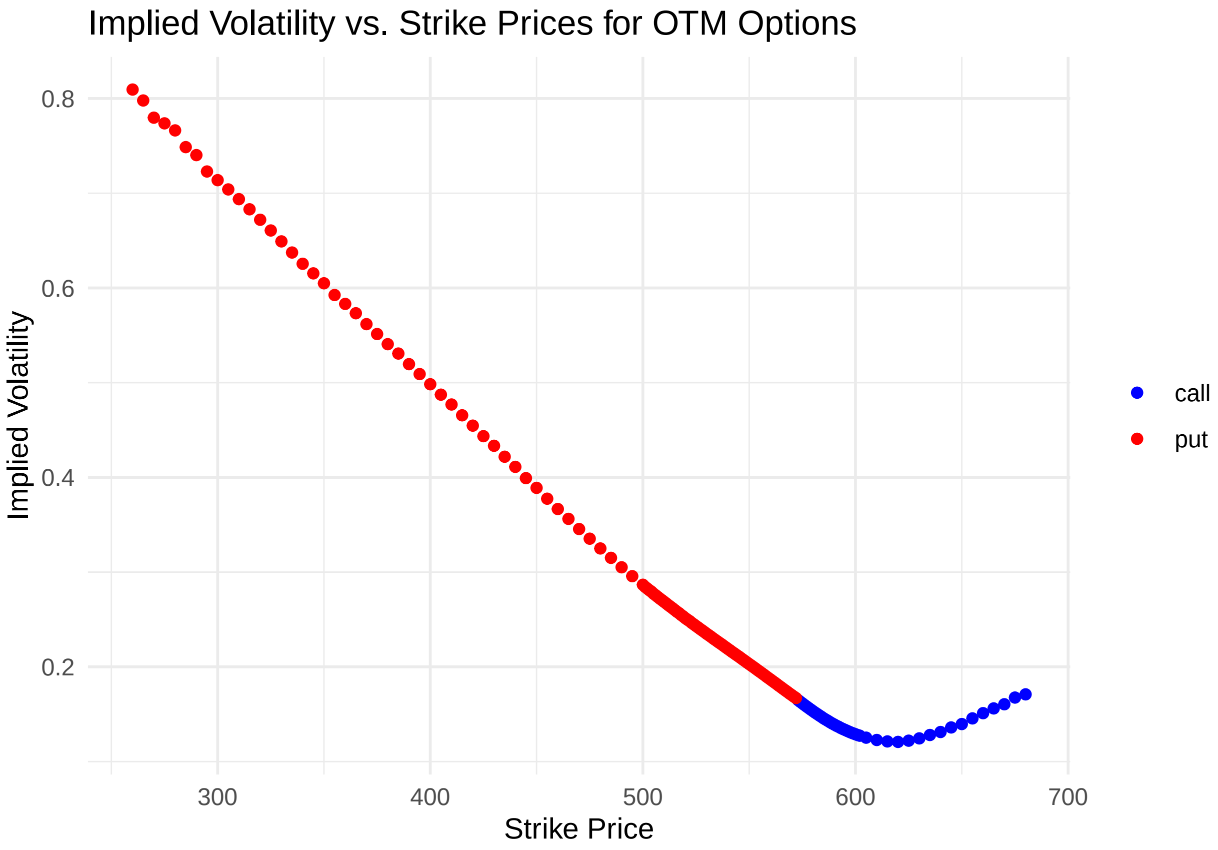
```
call_options_yahoo
## # A tibble: 167 x 6
##    `Exp Date` Strike OpenInterest   Bid   Ask OptionType
##    <date>      <dbl>        <dbl> <dbl> <dbl> <chr>
##  1 2024-11-15    260          104  314.  315. call
##  2 2024-11-15    265           17  309.  310. call
##  3 2024-11-15    270            4  304.  305. call
##  4 2024-11-15    275           29  299.  300. call
##  5 2024-11-15    280           21  294.  295. call
##  6 2024-11-15    285          294  289.  290. call
##  7 2024-11-15    290          341  284.  285. call
##  8 2024-11-15    295          530  279.  280. call
##  9 2024-11-15    300          574  274.  275. call
## 10 2024-11-15    305          554  269.  271. call
## # i 157 more rows
put_options_yahoo
## # A tibble: 167 x 6
##    `Exp Date` Strike OpenInterest   Bid   Ask OptionType
##    <date>      <dbl>        <dbl> <dbl> <dbl> <chr>
##  1 2024-11-15    260         2570  0.04  0.06 put
##  2 2024-11-15    265        11742  0.05  0.06 put
##  3 2024-11-15    270        14978  0.05  0.06 put
##  4 2024-11-15    275         4862  0.06  0.07 put
##  5 2024-11-15    280         3026  0.07  0.08 put
##  6 2024-11-15    285         8577  0.07  0.08 put
##  7 2024-11-15    290         1029  0.08  0.09 put
##  8 2024-11-15    295        14053  0.08  0.09 put
##  9 2024-11-15    300         9103  0.09  0.1  put
## 10 2024-11-15    305         2690  0.1   0.11 put
## # i 157 more rows
```

```
bot_yahoo_final <- bind_rows(call_options_yahoo, put_options_yahoo) %>%
  mutate(Underlying = 572.98, Today = dmy("05-10-2024")) %>%
  mutate_if(is.numeric, as.double) %>%
  filter(!(Bid == 0 & Ask == 0))  # Remove rows where both Bid and Ask are 0

bot_yahoo_final
## # A tibble: 327 x 8
##    `Exp Date` Strike OpenInterest   Bid   Ask OptionType Underlying Today
##    <date>      <dbl>        <dbl> <dbl> <dbl> <chr>           <dbl> <date>
##  1 2024-11-15    260          104  314.  315. call             573. 2024-10-05
##  2 2024-11-15    265           17  309.  310. call             573. 2024-10-05
##  3 2024-11-15    270            4  304.  305. call             573. 2024-10-05
##  4 2024-11-15    275           29  299.  300. call             573. 2024-10-05
##  5 2024-11-15    280           21  294.  295. call             573. 2024-10-05
##  6 2024-11-15    285          294  289.  290. call             573. 2024-10-05
##  7 2024-11-15    290          341  284.  285. call             573. 2024-10-05
##  8 2024-11-15    295          530  279.  280. call             573. 2024-10-05
##  9 2024-11-15    300          574  274.  275. call             573. 2024-10-05
## 10 2024-11-15    305          554  269.  271. call             573. 2024-10-05
## # i 317 more rows
```

5.2.2 Column Transformation

```r
library(NMOF)

# Define rate and today
rate <- 0.05
today <- as.Date("2024-10-05")  # Data retrieved on 05/10/2024
tau <- as.numeric((as.Date("2024-11-15") - today)) / 365

otm_options_yahoo <- bot_yahoo_final %>%
  filter((OptionType == "call" & Strike > Underlying) |
         (OptionType == "put" & Strike < Underlying))

# Calculate implied volatility
otm_options_yahoo <- otm_options_yahoo %>%
  rowwise() %>%
  mutate(ImpliedVolatility = vanillaOptionImpliedVol(
    exercise = "european",
    price = (Bid + Ask) / 2,
    S = Underlying,
    X = Strike,
    tau = tau,
    r = rate,
    q = 0,
    type = OptionType
  )) %>%
  ungroup()

ggplot(otm_options_yahoo, aes(x = Strike, y = ImpliedVolatility, color = OptionType)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Implied Volatility vs. Strike Prices for OTM Options",
       x = "Strike Price",
       y = "Implied Volatility") +
  scale_color_manual(values = c("call" = "blue", "put" = "red")) +
  theme(legend.title = element_blank())
```

5.2.3 Plot of Implied Volatility


Implied Volatility vs. Strike Prices for OTM Options
Implied Volatility
0.8
0.6
0.4
0.2
300
400
500
600
700
Strike Price
call
put