Project #2: More on portfolios. Futures prices. Call prices.

Samuel Kalisch

2024-03-04

```r
library(ggplot2)
library(tidyverse)
library(lubridate)
```

Problem #1 (25 points)

Your initial wealth is exactly \$100. You are allowed to invest in shares of a particular stock. You are also allowed to both lend and borrow at the continuously compounded risk-free interest rate of 0.05. Keeping your money uninvested is **not allowed**.

You can rebalance your portfolio every morning, once you have observed the opening stock price. This means that you can change the number of shares you own (if you decide to do so) and accordingly adapt your risk-free investment.

You proceed to create a "rule' ' according to which you will be rebalancing your portfolio. Here is a possible rule you can use:

If the stock price drops overnight (regardless of the extent of the drop), sell half of the stock investment. If the stock price rises overnight (regardless of the extent of the increase), double the amount of the stock investment. If the stock price does not change, do nothing. In all three cases, the rest of my wealth is to be invested at the risk-free rate (if needed, I will borrow at the continuously compounded risk-free interest rate).

Over the following 10 days, you observe the following stock prices for a non-dividend-paying stock:

Day	0	1	2	3	4	5	6	7	8	9	10
Stock price	100	80	64	80	64	80	100	80	64	80	100

As the time passes you follow investment rules above to rebalance your portfolio. Complete the following table describing your portfolio **just before and just after** the rebalancing is done. Even more precisely, for the 10 days, both for "morning" and "evening" print out:

- the number of shares of stock in the portfolio,
- the balance of the risk-free investment,
- the wealth in the stock,
- the total wealth.

First take note of the continuously compounded, risk-free interest rate:

```
r=0.05
```

Then, I create a vectore containing the evolution of the stock price:

```
s= c(100, 80, 64, 80, 64, 80, 100, 80, 64, 80, 100)
```

Rule (assumed start with 50% in shares):

```
#set up vectors
pi.v=numeric(11)
cash=numeric(11)
wealth=numeric(11)

#initialize
wealth[1]=100
pi.v[1]= 0.5
cash[1]=wealth[1]-pi.v[1]*s[1]

print(sprintf("Day=%d, %s", 1, "morning"))
## [1] "Day=1, morning"
  print(sprintf("shares=%6.4f, cash=%6.4f, wealth in stock=%6.4f, total wealth=%6.4f",
              pi.v[1], cash[1], pi.v[1]*s[1], wealth[1]))
## [1] "shares=0.5000, cash=50.0000, wealth in stock=50.0000, total wealth=100.0000"
#dynamics of the rule
rebalance<-function(shares,s.prev,s.curr){
  if(s.prev>s.curr){return(shares/2)}
  if(s.prev<s.curr){return(shares*2)}
  if(s.prev==s.curr){return(shares)}
  }

#we move forward thorugh time with this rule
for(i in 1:10){
  cash[i+1]=cash[i]*exp(r/365)
  wealth[i+1]=cash[i+1]+pi.v[i]*s[i+1]

  #before re-balancing
  print(sprintf("Day=%d, %s", i, "evening"))
  print(sprintf("shares=%6.4f, cash=%6.4f, wealth in stock=%6.4f, total wealth=%6.4f",
              pi.v[i], cash[i+1], pi.v[i]*s[i+1], wealth[i+1]))

  #now we re-balance
  pi.v[i+1]=rebalance(pi.v[i],s[i],s[i+1])
  cash[i+1]=wealth[i+1]-pi.v[i+1]*s[i+1]

  #after re-balancing
  print(sprintf("Day=%d, %s", i+1, "morning"))
  print(sprintf("shares=%6.4f, cash=%6.4f, wealth in stock=%6.4f, total wealth=%6.4f",
              pi.v[i+1], cash[i+1], pi.v[i+1]*s[i+1], wealth[i+1]))
}
## [1] "Day=1, evening"
## [1] "shares=0.5000, cash=50.0068, wealth in stock=40.0000, total wealth=90.0068"
## [1] "Day=2, morning"
## [1] "shares=0.2500, cash=70.0068, wealth in stock=20.0000, total wealth=90.0068"
```

```
## [1] "Day=2, evening"
## [1] "shares=0.2500, cash=70.0164, wealth in stock=16.0000, total wealth=86.0164"
## [1] "Day=3, morning"
## [1] "shares=0.1250, cash=78.0164, wealth in stock=8.0000, total wealth=86.0164"
## [1] "Day=3, evening"
## [1] "shares=0.1250, cash=78.0271, wealth in stock=10.0000, total wealth=88.0271"
## [1] "Day=4, morning"
## [1] "shares=0.2500, cash=68.0271, wealth in stock=20.0000, total wealth=88.0271"
## [1] "Day=4, evening"
## [1] "shares=0.2500, cash=68.0364, wealth in stock=16.0000, total wealth=84.0364"
## [1] "Day=5, morning"
## [1] "shares=0.1250, cash=76.0364, wealth in stock=8.0000, total wealth=84.0364"
## [1] "Day=5, evening"
## [1] "shares=0.1250, cash=76.0469, wealth in stock=10.0000, total wealth=86.0469"
## [1] "Day=6, morning"
## [1] "shares=0.2500, cash=66.0469, wealth in stock=20.0000, total wealth=86.0469"
## [1] "Day=6, evening"
## [1] "shares=0.2500, cash=66.0559, wealth in stock=25.0000, total wealth=91.0559"
## [1] "Day=7, morning"
## [1] "shares=0.5000, cash=41.0559, wealth in stock=50.0000, total wealth=91.0559"
## [1] "Day=7, evening"
## [1] "shares=0.5000, cash=41.0615, wealth in stock=40.0000, total wealth=81.0615"
## [1] "Day=8, morning"
## [1] "shares=0.2500, cash=61.0615, wealth in stock=20.0000, total wealth=81.0615"
## [1] "Day=8, evening"
## [1] "shares=0.2500, cash=61.0699, wealth in stock=16.0000, total wealth=77.0699"
## [1] "Day=9, morning"
## [1] "shares=0.1250, cash=69.0699, wealth in stock=8.0000, total wealth=77.0699"
## [1] "Day=9, evening"
## [1] "shares=0.1250, cash=69.0794, wealth in stock=10.0000, total wealth=79.0794"
## [1] "Day=10, morning"
## [1] "shares=0.2500, cash=59.0794, wealth in stock=20.0000, total wealth=79.0794"
## [1] "Day=10, evening"
## [1] "shares=0.2500, cash=59.0875, wealth in stock=25.0000, total wealth=84.0875"
## [1] "Day=11, morning"
## [1] "shares=0.5000, cash=34.0875, wealth in stock=50.0000, total wealth=84.0875"
```

Problem #2 (25 points)

Repeat the above problem with **your own rule** based on *technical analysis*, i.e., the study of past prices. Explain why you designed the rule the way you did and why you believe it would prove to be a successful investment strategy. Provide references if you have any.

Rule:

Mean reversing strategy:

Short Signal: If the short-term moving average (2-day moving average) crosses above the long-term moving average (4-day moving average) then short one unit of the stock.

Long Signal: If the short-term moving average crosses below the long-term moving average then invest all available wealth in the stock.

Hold: If there is no crossover, do nothing and hold the current position.

Explanation:

I formulated my rule around a 2-day and 4-day mean reversing strategy. This decision was influenced by a desire to capture short-term trends efficiently. In an ideal scenario, with more extensive historical data, I would opt for longer moving averages to provide a broader perspective on trends. Basic mean reversing strategies are widely discussed in introductory technical analysis literature. Note: the effectiveness of any strategy depends on multiple factors, and rigorous backtesting is necessary to evaluate its performance across various market conditions/cycles.

Investopedia: Mean Reversion

```
#set up vectors
pi.v=numeric(11)
cash=numeric(11)
wealth=numeric(11)

#initialize
wealth[1]=100
pi.v[1]= 0
cash[1]=wealth[1]-pi.v[1]*s[1]

print(sprintf("Day=%d, %s", 1, "morning"))
## [1] "Day=1, morning"
  print(sprintf("shares=%6.4f, cash=%6.4f, wealth in stock=%6.4f, total wealth=%6.4f",
              pi.v[1], cash[1], pi.v[1]*s[1], wealth[1]))
## [1] "shares=0.0000, cash=100.0000, wealth in stock=0.0000, total wealth=100.0000"
#dynamics of the rule
rebalance<-function(st_avg,lt_avg,s.curr){
  if(st_avg>lt_avg){return(-1)}
  if(st_avg<lt_avg){return(wealth[i+1]/s[i+1])}
  if(st_avg==lt_avg){return(s.curr)}
  }

#we move forward thorugh time with this rule
for(i in 1:10){
  cash[i+1]=cash[i]*exp(r/365)
  wealth[i+1]=cash[i+1]+pi.v[i]*s[i+1]

  #before re-balancing
  print(sprintf("Day=%d, %s", i, "evening"))
  print(sprintf("shares=%6.4f, cash=%6.4f, wealth in stock=%6.4f, total wealth=%6.4f",
              pi.v[i], cash[i+1], pi.v[i]*s[i+1], wealth[i+1]))

  #rebalance, we don't trade in the first 4 days to gather info
  if(i+1==4){print("----------------------Mean reversing strategy begins.----------------------")}
  if(i+1>=4){
    short_term_avg = (s[i]+s[i+1])/2
    long_term_avg = (s[i-2]+s[i-1]+s[i]+s[i+1])/4

    print(sprintf("Longterm Average=%6.4f, Shorterm Average=%6.4f",long_term_avg,short_term_avg))

    pi.v[i+1]=rebalance(short_term_avg, long_term_avg,pi.v[i])
    cash[i+1]=wealth[i+1]-pi.v[i+1]*s[i+1]
  }


  #after re-balancing
```

```r
  print(sprintf("Day=%d, %s", i+1, "morning"))
  print(sprintf("shares=%6.4f, cash=%6.4f, wealth in stock=%6.4f, total wealth=%6.4f",
                pi.v[i+1], cash[i+1], pi.v[i+1]*s[i+1], wealth[i+1]))
}
## [1] "Day=1, evening"
## [1] "shares=0.0000, cash=100.0137, wealth in stock=0.0000, total wealth=100.0137"
## [1] "Day=2, morning"
## [1] "shares=0.0000, cash=100.0137, wealth in stock=0.0000, total wealth=100.0137"
## [1] "Day=2, evening"
## [1] "shares=0.0000, cash=100.0274, wealth in stock=0.0000, total wealth=100.0274"
## [1] "Day=3, morning"
## [1] "shares=0.0000, cash=100.0274, wealth in stock=0.0000, total wealth=100.0274"
## [1] "Day=3, evening"
## [1] "shares=0.0000, cash=100.0411, wealth in stock=0.0000, total wealth=100.0411"
## [1] "----------------------Mean reversing strategy begins.----------------------"
## [1] "Longterm Average=81.0000, Shorterm Average=72.0000"
## [1] "Day=4, morning"
## [1] "shares=1.2505, cash=0.0000, wealth in stock=100.0411, total wealth=100.0411"
## [1] "Day=4, evening"
## [1] "shares=1.2505, cash=0.0000, wealth in stock=80.0329, total wealth=80.0329"
## [1] "Longterm Average=72.0000, Shorterm Average=72.0000"
## [1] "Day=5, morning"
## [1] "shares=1.2505, cash=0.0000, wealth in stock=80.0329, total wealth=80.0329"
## [1] "Day=5, evening"
## [1] "shares=1.2505, cash=0.0000, wealth in stock=100.0411, total wealth=100.0411"
## [1] "Longterm Average=72.0000, Shorterm Average=72.0000"
## [1] "Day=6, morning"
## [1] "shares=1.2505, cash=0.0000, wealth in stock=100.0411, total wealth=100.0411"
## [1] "Day=6, evening"
## [1] "shares=1.2505, cash=0.0000, wealth in stock=125.0514, total wealth=125.0514"
## [1] "Longterm Average=81.0000, Shorterm Average=90.0000"
## [1] "Day=7, morning"
## [1] "shares=-1.0000, cash=225.0514, wealth in stock=-100.0000, total wealth=125.0514"
## [1] "Day=7, evening"
## [1] "shares=-1.0000, cash=225.0822, wealth in stock=-80.0000, total wealth=145.0822"
## [1] "Longterm Average=81.0000, Shorterm Average=90.0000"
## [1] "Day=8, morning"
## [1] "shares=-1.0000, cash=225.0822, wealth in stock=-80.0000, total wealth=145.0822"
## [1] "Day=8, evening"
## [1] "shares=-1.0000, cash=225.1130, wealth in stock=-64.0000, total wealth=161.1130"
## [1] "Longterm Average=81.0000, Shorterm Average=72.0000"
## [1] "Day=9, morning"
## [1] "shares=2.5174, cash=0.0000, wealth in stock=161.1130, total wealth=161.1130"
## [1] "Day=9, evening"
## [1] "shares=2.5174, cash=0.0000, wealth in stock=201.3913, total wealth=201.3913"
## [1] "Longterm Average=81.0000, Shorterm Average=72.0000"
## [1] "Day=10, morning"
## [1] "shares=2.5174, cash=0.0000, wealth in stock=201.3913, total wealth=201.3913"
## [1] "Day=10, evening"
## [1] "shares=2.5174, cash=0.0000, wealth in stock=251.7391, total wealth=251.7391"
## [1] "Longterm Average=81.0000, Shorterm Average=90.0000"
## [1] "Day=11, morning"
## [1] "shares=-1.0000, cash=351.7391, wealth in stock=-100.0000, total wealth=251.7391"
```

Problem #3 (25 points)

The Dow index and Dow futures

Please, in no particular order read the Wikipedia entry about *futures contracts* and listen to the fun and educational episode of *Planet money* about credit risk in cafeteria futures. Both are linked below:

Wikipedia: Futures

Planet money: Delicious cake futures

(10 points) Provide the definitions of the forward contract and the futures contract. Then, highlight the similarities and the differences between the two.

The files "dow.csv" and "dow-futures.csv" contain index prices and the futures prices for delivery on a particular date in the future.

Forward Contract: A forward contract is a financial agreement between two parties to buy or sell an asset at a future date for a price agreed upon today. This contract is customizable in terms of the asset, quantity, delivery date, and price. It is traded over-the-counter (OTC), meaning it is not standardized and is typically tailored to the specific needs of the parties involved. Since it is not traded on an exchange, the counterparty risk is a significant consideration.

Futures Contract: A futures contract is a standardized financial agreement to buy or sell a specified quantity of an asset at a predetermined price on a specified future date. Unlike forward contracts, futures are traded on organized exchanges, which act as intermediaries and enforce standardization. The standardized nature of futures contracts reduces counterparty risk and enhances liquidity. The exchange also requires participants to deposit margin to mitigate the risk of default.

Similarities:

1. *Purpose:* Both forward and futures contracts are financial instruments designed for hedging or speculating on future price movements of underlying assets.

2. *Agreement:* In both contracts, parties agree to buy or sell an asset at a future date for a predetermined price.

3. *Risk Management:* Both contracts can be used for risk management, allowing parties to hedge against price fluctuations in the underlying asset.

Differences:

1. *Standardization:* Futures contracts are standardized and traded on organized exchanges, whereas forward contracts are customized agreements negotiated directly between the two parties involved.

2. *Exchange vs. OTC:* Futures contracts are traded on exchanges, providing a centralized marketplace, transparency, and reduced counterparty risk. Forward contracts, on the other hand, are traded over-the-counter (OTC), leading to higher counterparty risk and less transparency.

3. *Flexibility:* Forward contracts offer more flexibility in terms of customization, allowing parties to tailor the terms to their specific needs. Futures contracts have less flexibility due to their standardized nature.

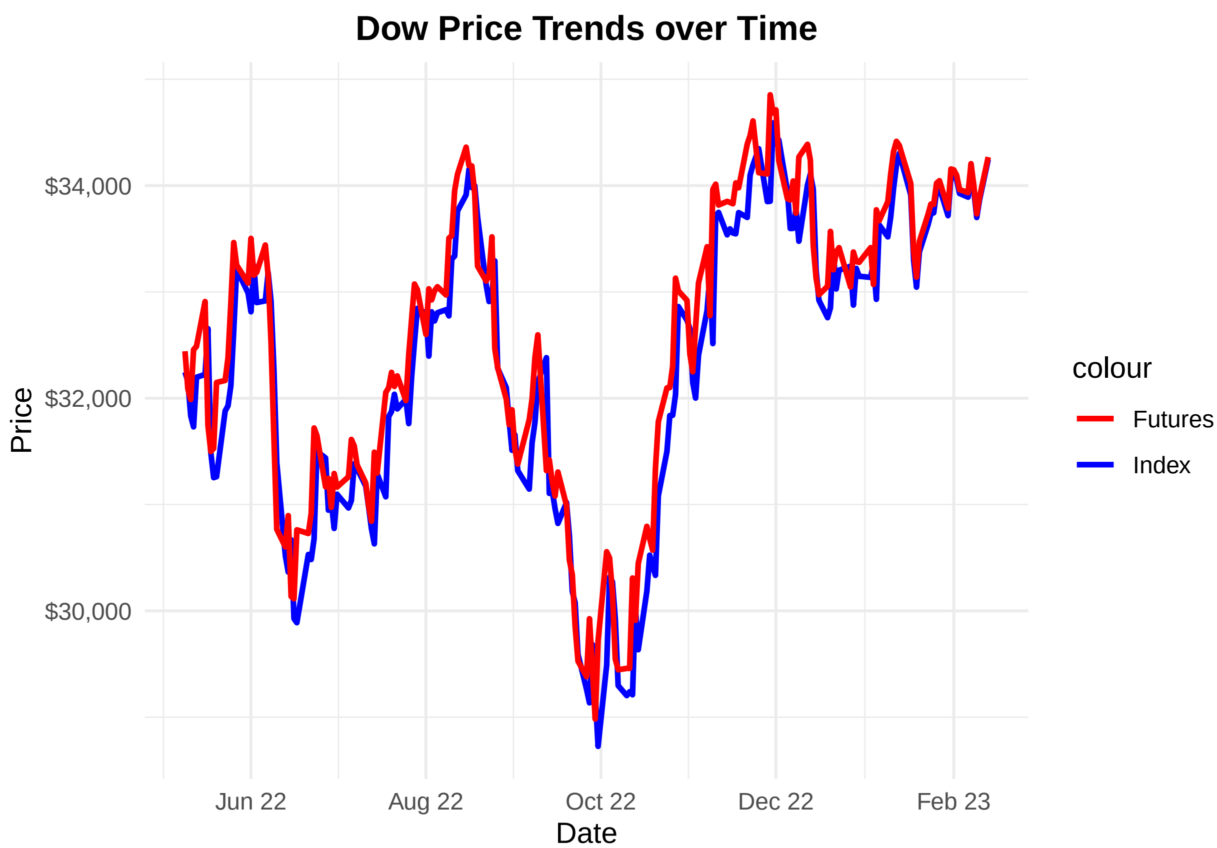
4. *Counterparty Risk*: Counterparty risk is higher in forward contracts because they are private agreements. In futures contracts, the exchange acts as an intermediary, enforcing contracts and reducing the risk of default through margin requirements.

(5 points) Import the data into a data frame (or two?). Draw the time-plot of the evolution of the closing index prices and the futures prices on the same coordinate system. You do **not** need to put the calendar days on the horizontal axis, but you **do** need to label your axes and give your time-plot a title indicating the dates. Make sure that you plot the two trajectories in **different** colors indicating in the text which color corresponds to which company.

Here is a link to a straightforward tutorial which will help you accomplish the above neatly:

Timeline plotting

```
dow<-read.csv("dow.csv")
dow_futures<-read.csv("dow-futures.csv")

# Adjust to make them have same number of days
dow_futures <- dow_futures[-1, ]

# Wrangle the Data Frame
combined = dow%>%
  select(Date,Close)%>%
  mutate(F =dow_futures%>%select(Close))

# customize the date with Lubridate
combined$Date<- as.Date(combined$Date, origin = "1899-12-30")

# Plotting and Formatting
ggplot(combined, aes(x = Date)) +
  geom_line(aes(y = Close, color = "Index"), linewidth = 1) +
  geom_line(aes(y = F$Close, color = "Futures"), linewidth = 1) +
  labs(title = "Dow Price Trends over Time",
       x = "Date",
       y = "Price") +
  scale_color_manual(values = c("Index" = "blue", "Futures" = "red")) +
  scale_x_date(date_breaks = "2 month", labels = scales::date_format("%b %y")) +
  scale_y_continuous(labels = scales::dollar_format(big.mark = ",")) +
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```



(10 points) What do you notice when you compare the timeplots? Which of the formulae for calculating futures prices you encountered on the Wikipedia entry come in handy?

```
combined%>%count(F$Close>Close)%>%
  mutate(probability = n / sum(n))%>%
  round(3)
##   F$Close > Close   n probability
## 1               0  39       0.202
## 2               1 154       0.798
combined = combined%>%mutate(Difference = (F$Close-Close)/Close)

summary(combined$Difference)%>%round(4)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.0328  0.0011  0.0054  0.0071  0.0138  0.0446
```

Differences & SimilariltIes: When I compare timeplots I can see that the futures from the Dow is trading at a higher price, in fact from the table above, the Dow Future's price is higher 79.8% of the time. Additionally, it is higher by an average of 0.7% with the highest being 4.4%. However, the future' price closely follows the actual prices and shape of the index.

Formula for Calculating Future's prices:

Arbitrage arguments apply when an asset is abundant, and the forward price represents the future value discounted at the risk-free rate. Any deviation from this theoretical price creates a riskless profit opportunity, which is arbitraged away.

Assuming constant rates and employing continuous compounding, the value of the futures/forward price, for a simple, non-dividend paying asset (such as the Dow) can be determined by continuously compounding the present value using the continuous rate of risk-free return. Additonaly,in an efficient market, the futures price should balance supply and demand, representing the continuous compounded present value of an unbiased expectation for the asset's price at the delivery date T. This yields the following formula.

$$F(t,T) = E_t[s(t)] \cdot e^{r(T-t)}$$

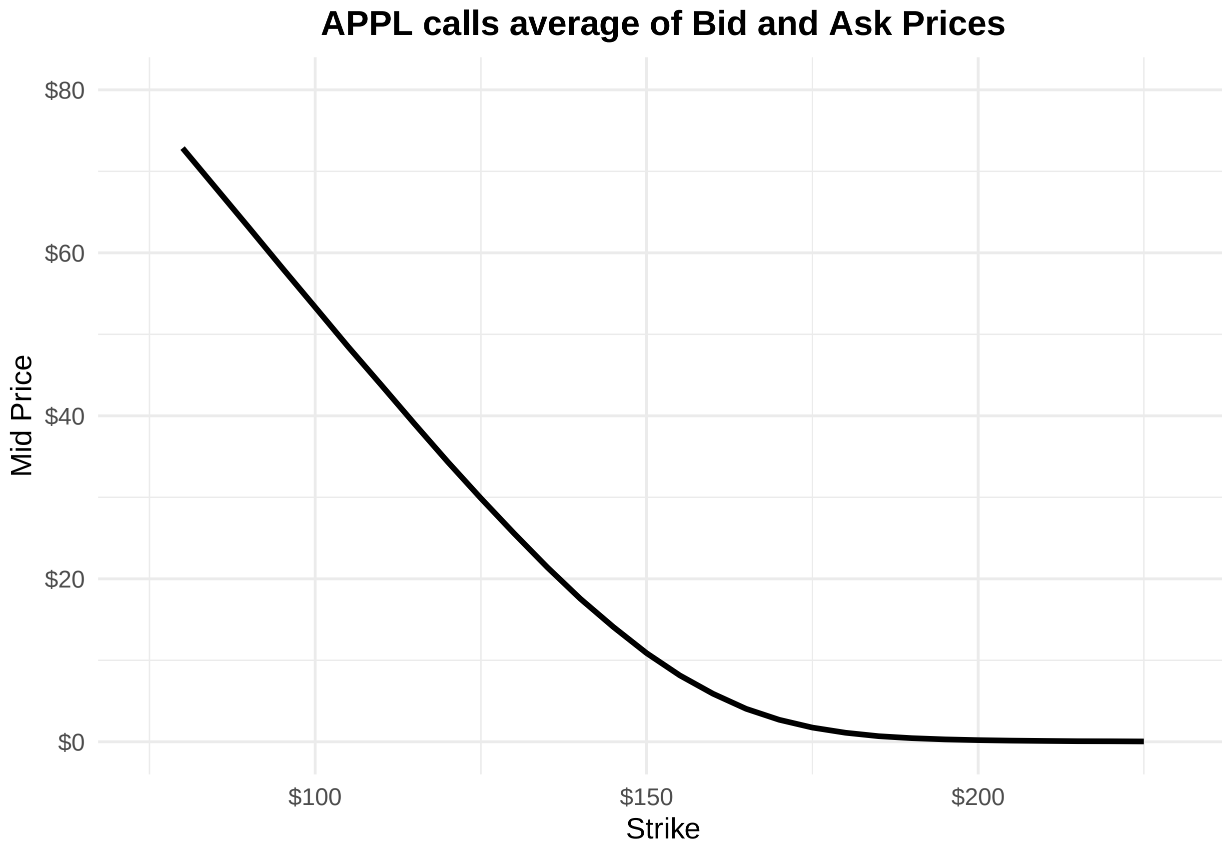
Problem #4 (25 points)

Apple calls

The file "apple-calls.csv" contains call prices observed on February 14th for European call options on the stocks of Apple Inc with expiry on May 19th and with varying strike prices.

(5 points) Import the data into a data frame. Create a line plot of the mid price (defined as the midpoint between the bid and the ask price) as it depends on the strike.

```r
apple_calls<-read.csv("apple-calls.csv")

apple_calls = apple_calls%>%
  mutate(Mid = (Bid+Ask)/2)%>%
  select(-X,-X.1,-X.2,-X.3)

# Plotting and Formatting
ggplot(apple_calls, aes(x = Strike)) +
  geom_line(aes(y = Mid), linewidth = 1) +
  labs(title = "APPL calls average of Bid and Ask Prices",
       x = "Strike",
       y = "Mid Price") +
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5, face = "bold"))+
  scale_x_continuous(labels = scales::dollar_format(big.mark = ","),limits = c(75, 230)) +
  scale_y_continuous(labels = scales::dollar_format(big.mark = ","),limits = c(0, 80))
```



(5 points) What do you notice about the monotonicity and convexity of the above curve?

The curve is monotonically decreasing and it is convex. The declining trend of the curve suggests a consistent decrease in values. Additionally, the convex shape indicates that the rate of decrease is slowing down over time.

(15 points) Show that the same conclusions regarding the monotonicity and convexity of the call prices with respect to the strike must hold in general **under the no arbitrage assumption**.

Emprically:

First, I will show that it is monotonically decreasing:

```r
# sort in decreasing order
apple_calls = apple_calls%>%
  arrange(Strike)

# calculate the deltas
apple_calls = apple_calls %>% mutate(price_diff = Mid - lag(Mid, default = first(Mid)))

# check if condition holds
print(all(apple_calls$price_diff <= 0))
## [1] TRUE
```

Second, I will show that it is convex:

```r
# sort in decreasing order
apple_calls = apple_calls %>%
  arrange(Strike)

# calculate  lambda, xr, xl
convexity_data <- apple_calls %>%
  mutate(
    xl = lag(Mid, default = first(Strike)),
    xr = lead(Mid, default = last(Strike)),
    lambda = (xr - Strike) / (xr - xl),
    #f(x*) - e  <= lambda*f(xl)-(1- lambda)*f(xr), e = .0000001,fix rounding errors
    check = Strike -.0000001<= lambda * xl + (1 - lambda) * xr
  )

# check if condition holds
print(all(convexity_data$check))
## [1] TRUE
```

Mathematically:

We assume no arbitrage for all arguments.

First, I will show that it is monotonically decreasing:

Claim. Assume that one Call has a higher strike price. If K2 >= K1 then V1(0)>=V2(0),

Proof. By definition, the value of a call at time T is Vc(T) = (S(T) - K)+.

We have that V1(T) = (S(T) - K1)+ and V2(T) = (S(T) - K2)+. Since K2 is greater than or equal to K1, there is no scenario where (S(T) - K1)+ is less than (S(T) - K2)+, implying V1(T) is greater than or equal to V2(T).

Since the future value of the option is higher or equal at any point in time, the present value of the option is also higher, i.e., V1(0) is greater than or equal to V2(0). Therefore, the function is monotonically decreasing as a function of the strike, as desired.

Now, I will show that it is convex:

Claim. A monotonically decreasing function of the price of a Call as a function of the strike is convex.

Proof. Let K1 < K2 < K3, assume that Vc(K2) > lambda*Vc(K1) + (1-lambda)*Vc(K3). We will show that there exist an arbitrage portfolio. We defie lambda as the following:

$$\lambda = \frac{K_3 - K_2}{K_3 - K_1}$$

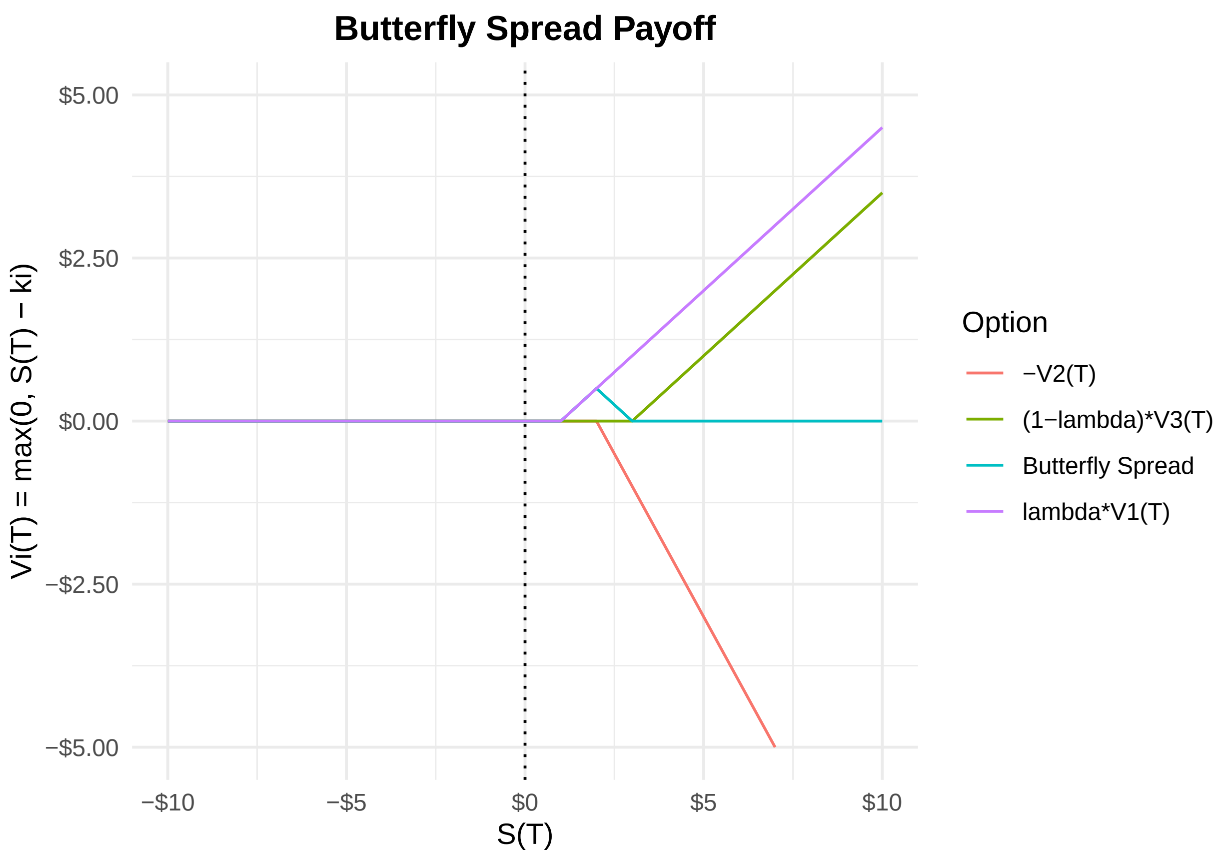
Firstly, we will show that the payoff is strictly positive. Let us construct a Butterfly spread. Visual Representation of Butterfly Spread. We define the butterfly as longing lambda and (1-lambda) options of K1 and K3 respectively and shorting one of K2. Thus the initial cost and profit are:

$$payoff = \lambda \cdot V_1(T) - V_2(T) + (1 - \lambda) \cdot V_3(T)$$

$$InitialCost = \lambda \cdot V_1(0) - V_2(0) + (1 - \lambda) \cdot V_3(0)$$

By our definition of lambda and the payyoff funtion of the butterfly, it's payoff function is strictly positive.

```r
# Function to calculate max(0, x - ki)
max_function <- function(x, k) {pmax(0, x - k)}

# Values for x-axis
x_values <- seq(-10, 10, by = 0.1)

# Values for k1, k2, k3
k1 <- 1
k2 <- 2
k3 <- 3

lambda = (k3 - k2) / (k3 - k1)

# Create a data frame with x values and corresponding y values for each function
data <- data.frame(
  x = rep(x_values, 4),
  y = c(lambda*max_function(x_values, k1), -max_function(x_values, k2), (1-lambda)*max_function(x_values
        lambda*max_function(x_values, k1) -max_function(x_values, k2) + (1-lambda)*max_function(x_values
  Option = rep(c("lambda*V1(T)", "-V2(T)", "(1-lambda)*V3(T)","Butterfly Spread"), each = length(x_valu
)

# Create the ggplot
ggplot(data, aes(x = x, y = y, color = Option)) +
  geom_line() +
  labs(title = "Butterfly Spread Payoff", x = "S(T)", y = "Vi(T) = max(0, S(T) - ki)") +
  geom_vline(xintercept = 0, linetype = "dotted", color = "black") +
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5, face = "bold"))+
   scale_x_continuous(labels = scales::dollar_format(big.mark = ",")) +
  scale_y_continuous(labels = scales::dollar_format(big.mark = ","),limits = c(-5, 5))
## Warning: Removed 30 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



Now, we will show that the cost is negative. Implying that it has a strictly positive profit. From our assumption:

$$Vc(k2) > \lambda \cdot Vc(k1) + (1 - \lambda) \cdot Vc(k3)$$

Implying that:

$$0 > \lambda \cdot Vc(k1) - Vc(k2) + (1 - \lambda) \cdot Vc(k3)$$

By our Initial cost function we have that:

$$0 > InitialCost$$

Since the payoff function is strictly positive and the Initial Cost is strictly negative by assumption, we have that the butterfly spread has a strictly positive profit. Therfore, it contradicts our no arbitrage assumption, as desired.