



MTHM005: MATHEMATICAL SCIENCES PROJECT

# Pricing Asian Options

Candidate Number: 145670

Supervised by Prof. BYOTT

November 15, 2022

## Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Insert some cool abstract that makes our examiner go, “hmm that’s a pretty good abstract”.

All files available online:

<https://github.com/leele2/Mathematics-in-Business-Project>

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	A brief overview of call and put options . . . . .	1
1.1.1	A short history of option trading . . . . .	2
1.1.2	Standard options . . . . .	2
1.1.3	Asian options . . . . .	2
1.2	Market definition and assumptions . . . . .	2
1.2.1	Asset definitions . . . . .	2
1.2.2	Market Assumptions . . . . .	3
1.3	Asset pricing models . . . . .	3
1.4	Pricing options . . . . .	3
1.4.1	Pricing standard options . . . . .	4
1.4.2	Pricing Asian options . . . . .	4
1.4.3	Monte-Carlo method . . . . .	4
	<b>Appendix A Matlab Files</b>	<b>6</b>
	<b>Appendix B Python Files</b>	<b>8</b>

## List of Figures and Tables

1.1	Time series plot of the average daily option trading volume per annum. Data provided by the Options Clearing Corporation (OCC) [Opt22]. Source code: Listing B.1 . . . . .	1
-----	--	---

# Chapter 1: Introduction

This report studies a vital financial derivative in today's markets, namely options. The importance of the option market has been shown by empirical studies which suggest that option trading improves information efficiency in the broader stock market [PP06, Li21], and also that firms with listed options experience lower implied cost of equity capital [NNT13]; indicating that options trading reduces the cost of capital [Li21]. The popularity of the options market can easily be seen in Figure 1.1, which shows the exponential growth in trading volume since standardized, exchange-traded stock options were first listed in The Chicago Board Options Exchange in 1973 [Mar02]. In 2020 single stock option trading volume became higher than the underlying stock volume for the first time ever [Fin20].

It is this explosive popularity and significance which have motivated this report. We will begin by describing standard options and explore popular methods that are used to price them. We will then move onto Asian options and look at the literature surrounding how to price them before implementing several pricing methods with the use of MATLAB. Furthermore, we will then take an analytical approach to determine how Asian options can be priced accurately and efficiently.

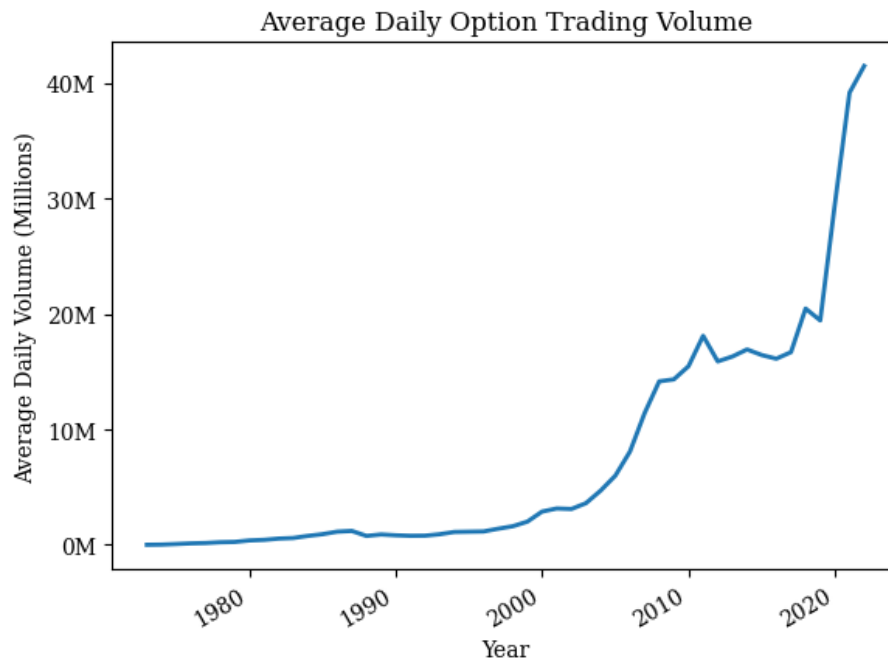


Figure 1.1: Time series plot of the average daily option trading volume per annum. Data provided by the Options Clearing Corporation (OCC) [Opt22]. Source code: Listing B.1

## 1.1 A brief overview of call and put options

Options are a particular type of financial derivative, a contract that details the conditions under which payments are made between two counterparties. They are purchased for a set fee, and in return the buyer is granted the right, but not the obligation to buy or sell an underlying asset — such as commodities, stocks or bonds — for a predetermined price (the strike price) on or before a determined date (the expiry date).

Call options allow the buyer to purchase an asset for the strike price at a future date. The buyer can make a return if the value of the asset is worth more than the strike price when exercised. Alternatively, put options allow the buyer to sell an asset for the strike price at a future date and the buyer can make a return if the value of the asset is less than the strike price when exercised.

The option market is widely considered a venue for informed trading [Li21, Hu14, CGM04], that is, investors trading with superior knowledge of the probability distribution of share prices, through either access to private information or skillful processing of public information [Gro75].

### 1.1.1 A short history of option trading

The history of financial options can be traced as far back as 6th century BCE when ancient Greek mathematician and philosopher Thales of Miletus predicted through his astrological knowledge there was going to be a great olive harvest. As he did not have much money, he used what he had as a deposit on the rights to the local olive presses; due to no competition he secured this at a relatively low price. When the harvest proved to be bountiful leading to high demand, Thales charged a high price for use of the presses and reaped a considerable profit. His deposit gave him the right but not the obligation to hire the presses, thus his losses were limited to his initial deposit [Ins12, Ari77].

Whilst this is quite a positive look on option trading, throughout history this has not always been the case. During the Dutch tulip bubble of the seventeenth century, tulips were seen as a status symbols which caused high demand and consequently drove their prices up, creating a bubble [Das11]. Tulip growers would buy puts to protect their profits in case the price of tulip bulbs went down and wholesalers would buy calls to protect against the risk of tulip bulbs going up. When the bubble eventually burst, there was no way to force investors to fulfil their obligations of the options contracts, due to the unregulated nature of the option market. This ultimately led to options gaining a dubious reputation and bans were later placed on them within Britain between 1733-1860 [Poi08].

During the late nineteenth century, brokers started to arrange deals between buyers and sellers of options for particular stocks at prices that were arranged between the two parties. Trades were arranged similarly until the 1960s when the options market started to become regulated by the Chicago Board of Trade. In 1973, the Chicago Board of Options Exchange (CBOE) began trading and for the first time options contracts were properly standardized. At the same time, the Options Clearing Corporation was established for centralized clearing and enforcing the proper fulfillment of contracts, ensuring that they were honored [Mar02].

### 1.1.2 Standard options

A standard option comes in two styles; European: which restricts the holder of the option to only exercise the option on the expiry date, and American: which allows the holder to exercise the option at anytime up till or on the expiry date. They will take the current value of the underlying asset as the spot price — that is the price that the asset can be purchased for on the open market. The payoff in this case then becomes the difference between the spot price and strike price.

### 1.1.3 Asian options

Whilst standard options involve using the spot price as the underlying value of the asset; this is not always the case with so-called exotic options. Exotic options differ in their payment structures, expiration dates, and/or strike prices. In the case of exotic fixed-strike price Asian options, the average price of the asset is used in place of the underlying asset value. This differs from fixed-price Asian options which instead use the average price of the asset to take place of the strike price. These are the two main variations of Asian style options but both of these can be varied further in how the averaging is calculated, for example: geometrically, arithmetically, average taken every day or average taken at the start of each month and so on. They can be varied further by having an expiry structure matching a European or American style option.

## 1.2 Market definition and assumptions

To begin pricing options we must first establish assumptions on the value of assets and the market in which they are traded. We will consider a market in which there are two classes of assets; risk-free assets and risky assets. We allow derivatives to exist on risky assets.

### 1.2.1 Asset definitions

We define our two classes of assets with the following assumptions:

#### Risk-free assets

Risk-free assets are assumed to have no uncertainty in regard to their future value and defined as returning interest continuously compounded at a fixed rate  $r$ . Hence, an investment of  $A$  made at time  $t$  will return  $Ae^{r(T-t)}$  at time  $T > t$  with no uncertainty. These can be positive e.g., making an investment, or negative e.g., taking out a loan. Risk-free assets are akin to fixed rate investments at a well-know bank or government bonds. However, even sovereign nations have been known to default on their bond repayments so in reality are not truly risk-free, but instead considered low-risk [K<sup>+</sup>15].

## Risky assets

Risky assets are assets in which the future value can not be determined without at least some uncertainty. These are typically shares in a company or commodities such as gold or oil. These assets are traded on the open market and their value is heavily dependent on the balance of supply and demand of the asset; this in turn is dependent on so many variables that we assume their future value to be a random process. Depending on our model we can make different assumptions on the distribution of the process and thus how the value evolves over a time period. In practice there are people who argue that future value of risky assets are truly random [Fam65], whilst others believe they can be at least partly deterministic where patterns and trends exists [SP85].

### 1.2.2 Market Assumptions

We shall assume that an investor may hold any quantity (including negative and/or fractional) of an asset, known as *divisibility* and *short-selling* assumption. We will also assume that our market is *frictionless* — that is that any quantity of assets can be bought and sold at the same price with no transaction cost and are entirely *liquid* meaning there is always a buyer for every seller and vice versa.

In practice these assumptions do not accurately represent the real markets, for example it is common place for markets to have bid-ask spreads which represents the different in price at which you can buy and sell assets for. Furthermore, transaction costs are impossible to fully eliminate - even in scenarios with no tax or commission fees, buying or selling assets (especially in large quantities) will in theory have an effect on the supply and demand of that asset; this introduces market impact costs [MVM<sup>+</sup>09]. There are other transaction fees to also consider such as: time spent deciding what assets to buy, computing power costs and their associated opportunity costs.

### No-Arbitrage assumption

Perhaps our most important market assumption is the No-Arbitrage principle; sometimes referred to as the no free-lunch principle. This assumption states that there can be no risk-free way to get a better rate of return than the market risk-free interest rate,  $r$ . The logic behind this assumption is that if there exists a way to earn a higher rate of return than  $r$  with no initial outlay; than other traders would employ the same method and market forces would eliminate the arbitrage opportunity. This assumption assumes that markets are perfectly efficient meaning all traders have access to the same information and that market adjustments to supply and demand happen instantaneously, which we know not to be true in practice.

## 1.3 Asset pricing models

Before we can begin pricing options, we must define our assumptions on the random process which describes how the asset value may evolve over a specified time period. We will first introduce the mathematical concepts in which these models are built upon. Then we will introduce a simple asset model and one which has more assumptions in hopes of better describing the future asset value in practice.

## 1.4 Pricing options

Since the holder of the contract is not obliged to exercise the contract at the expiry time, they do not hold any liability in the absence of a price to purchase the option. The problem then becomes, what is the correct price to charge the holder of the option to balance this inequality of liability.

### 1.4.1 Pricing standard options

Binomial method

Black-Scholes Formula

### 1.4.2 Pricing Asian options

Hull-White model

Costabile adjusted binomial method

Analytical solution for geometric average Asian options

### 1.4.3 Monte-Carlo method

# Bibliography

- [Ari77] Aristotle. *Aristotle's Politics I*. Number Book 1 in Aristotle's Politics. Longmans, Green, and Company, 1877.
- [CGM04] Sugato Chakravarty, Huseyin Gulen, and Stewart Mayhew. Informed trading in stock and option markets. *The Journal of Finance*, 59(3):1235–1257, 2004.
- [Das11] M. Dash. *Tulipomania: The Story of the World's Most Coveted Flower and the Extraordinary Passions it Aroused*. Orion, 2011.
- [Fam65] Eugene F. Fama. Random walks in stock market prices. *Financial Analysts Journal*, 21(5):55–59, 1965.
- [Fin20] Yahoo! Finance. Option trading volume higher than underlying stock volume for first time ever. <https://finance.yahoo.com/news/option-trading-volume-higher-underlying-211006236.html>, 06 2020.
- [Gro75] Steven D Grossman. An application of communication theory to financial statement analysis. *Financial Review*, 10(1):12–20, 1975.
- [Hu14] Jianfeng Hu. Does option trading convey stock price information? *Journal of Financial Economics*, 111(3):625–645, 2014.
- [Ins12] Business Insider. The story of the first-ever options trade in recorded history. <https://www.businessinsider.com/the-story-of-the-first-ever-options-trade-in-recorded-history-2012-3?r=US&IR=T>, 03 2012. Accessed: 18-07-2022 [Online].
- [K<sup>+</sup>15] Yakim Kitanov et al. Are risk free government bonds risk free indeed. *Economy & Business*, 9:523–530, 2015.
- [Li21] Keming Li. The effect of option trading. *Financial Innovation*, 7(1):1–32, 2021.
- [Mar02] J.W. Markham. *A Financial History of the United States Volume III (1970-2001) From the Age of Derivatives to the New Millennium*. A Financial History of the United States. M.E. Sharpe, 2002.
- [MVM<sup>+</sup>09] Esteban Moro, Javier Vicente, Luis G Moyano, Austin Gerig, J Doyne Farmer, Gabriella Vaglica, Fabrizio Lillo, and Rosario N Mantegna. Market impact and trading profile of hidden orders in stock markets. *Physical Review E*, 80(6):066102, 2009.
- [NNT13] Vic Naiker, Farshid Navissi, and Cameron Truong. Options trading and the cost of equity capital. *The Accounting Review*, 88(1):261–295, 2013.
- [Opt22] Options Clearing Corporation. Annual Volume and Open Interest Statistics. <https://www.theocc.com/Market-Data/Market-Data-Reports/Volume-and-Open-Interest/Historical-Volume-Statistics>, 06 2022. Accessed: 08-06-2022 [Online].
- [Poi08] Geoffrey Poitras. The early history of option contracts. *Vinzenz Bronzin's Option Pricing Models: Exposition and Appraisal*, 01 2008.
- [PP06] Jun Pan and Allen M. Poteshman. The Information in Option Volume for Future Stock Prices. *The Review of Financial Studies*, 19(3):871–908, 02 2006.
- [SP85] Robert J Shiller and Pierre Perron. Testing the random walk hypothesis: Power versus frequency of observation, 1985.

# Appendix A: Matlab Files

All files can be found: <https://github.com/leele2/Mathematics-in-Business-Project/tree/master/MATLAB%20Files>

Listing A.1: .././MATLAB Files/BinoAsian.m

```
1 %function BinoAsian(S0,E,T,r,sigma,N,F)
2 addpath('Functions')
3 %% Test
4 clear; tic; S0=100; E=100; T=0.25; r=0.1; sigma=0.1; N=10; F=@(S,A)max(A-E,0);
5 %% Function to evaluate European Call option by Binomial Method
6 % Parameters:
7 % S0 = initial share price
8 % E = exercise price`
9 % T = time to expiry
10 % r = riskfree interest rate
11 % sigma = volatility
12 % N = Number of steps
13 % F = Option Payoff (European Call in given)
14 %% Calculated parameters
15 dt = T/N; %Timestep
16 u = exp(sigma*sqrt(dt)); %Up price movement
17 d = 1/u; %Down price movement
18 disF = exp(-r*dt); %Discount factor over each timestep
19 p = (1/disF - d)/(u-d); %Risk-neutral probability
20 %% Initalizing Arrays and Functions
21 S = zeros(N+1,N+1); %Underlying Asset Price
22 S_k = cell(N+1,N+1); %S_max %% Cells are used so different sized vectors can be stored at each element in
    cell array
23 A = zeros(N+1,N+1); %Average of Underlying Asset Price
24 A_k = cell(N+1,N+1); %Representitive averages
25 C = zeros(N+1,N+1); %Price of Option
26 C_k = cell(N+1,N+1); %Option price for given representative avg
27 %F = @(S,A)max(A-E,0); %Option Payoff (European Call)
28 %% Functions to Calculate Maximum and Minimum Representative Averages
29 A_min = @(i,j) (1/(i+1)) * (sum(S0*d.^(0:(i-j))) + sum(S0*d.^(0:(j-1)+i-(2*j))));
30 A_max = @(i,j) (1/(i+1)) * (sum(S0*u.^(0:j)) + sum(S0*u.^(0:(i-j-1)+(2*j)-i)));
31 %% Calculate Underlying asset price
32 for i = 0:N
33     for j = 0:i
34         S(i+1,j+1) = S0*u^(j)*d^(i-j);
35     end
36 end
37 %% Calculating All S_max and Representative Averages
38 for i = 0:N
39     for j = 0:i %j indexes at j+1 due to matlab not allowing C_k{:,0)
40         A_k{i+1,j+1} = zeros(j*(i-j)+1,1); %Create Vector to hold rep avgs
41         S_k{i+1,j+1} = NaN(j*(i-j)+1,1); %Create Vector to hold S_max
42         A_k{i+1,j+1}(1) = A_max(i,j); %Assign A_max to first element in vector
43         % Paths with only up (i = j) or down movements (j = 0) or i = 1 will only have one representative
            average
44         if i < 1 || i == j || j == 0
45             S_k{i+1,j+1}(1) = S0*(u^j)*d^(i-j);
46             continue
47         end
48         %% Filling S_max Vector
49         % Setting Paramaters
50         n = j*(i-j); %Size of vector
51         rep = min([j,i-j]); %Maximum element repetition in vector
52         % Calculating unique elements in vector
53         unique = 2*numel(1:rep-1) + (n - (2*sum(1:rep-1)))/rep;
54         % First and last values
55         S_k{i+1,j+1}(1) = S0 * (u ^ j);
56         S_k{i+1,j+1}(end) = S0 * (u ^ (j - unique + 1));
57         %%The following if statement changes the final output%%
58         if n ~= 2
59             % Ascending/Descending repeated values
60             k = 2; %iterator
61             count = 2; %Number of elements filled
62             while k < rep
63                 % Filling from top
64                 S_k{i+1,j+1}(sum(1:k-1)+1:sum(1:k-1)+k) = repmat(S0*(u^(j-k+1)), k, 1);
65                 % Filling from bottom
66                 S_k{i+1,j+1}(n-sum(1:k-1)+1:n-sum(1:k-1)) = repmat(S0*(u^(j-unique+k)), k, 1);
67                 count = count + 2*k;
68                 k = k + 1;
69             end
70             % "Middle" repeated values
71             for l = 0 : (n-count)/rep - 1
72                 S_k{i+1,j+1}(count/2+1+l*rep:end) = [repmat(S0*u^(j-k+1), rep, 1);
73                 S_k{i+1,j+1}(count/2+1+l*rep+rep:end)];
74                 k = k + 1;
75             end
76         end
77     end
78 end
```



```

77     %% Calculating Representative Averages Using S_max
78     A_k{i+1,j+1}(j*(i-j) + 1) = A_min(i,j); %Assign A_min to last element in vector
79     for k = 2:j*(i-j)
80         A_k{i+1,j+1}(k) = A_k{i+1,j+1}(k-1) - (1/(i+1))* ...
81             (S_k{i+1,j+1}(k-1) - S_k{i+1,j+1}(k-1)*(d^2));
82     end
83 end
84 end
85 %% Pricing Option Value at Final Time (N)
86 for j = 0:N
87     C_k{N+1,j+1} = F(S(N+1,j+1),A_k{N+1,j+1});
88 end
89 %% Pricing Option
90 err = 1e-3;
91 for i = N-1:-1:0
92     for j = 0:i
93         C_k{i+1,j+1} = zeros(j*(i-j)+1,1);
94         for k = 1:j*(i-j)+1
95             %% Find K_u
96             Ku = ( (i+1)*A_k{i+1,j+1}(k) + u*S(i+1,j+1) )/(i+2);
97             [loc, ubound, lbound] = findInSorted(Ku,A_k{i+2,j+2},err);
98             % If found set accordingly
99             if loc > 0
100                 Cu = C_k{i+2,j+2}(loc);
101             else
102                 % If not found, interpolate between closest values
103                 Cu = C_k{i+2,j+2}(lbound)+(Ku-A_k{i+2,j+2}(lbound))*(...
104                     (C_k{i+2,j+2}(ubound)-C_k{i+2,j+2}(lbound))/...
105                     (A_k{i+2,j+2}(ubound)-A_k{i+2,j+2}(lbound)));
106             end
107             %% Find K_d
108             Kd = ( (i+1)*A_k{i+1,j+1}(k) + d*S(i+1,j+1) )/(i+2);
109             [loc, ubound, lbound] = findInSorted(Kd,A_k{i+2,j+1},err);
110             %If found set accordingly
111             if loc > 0
112                 Cd = C_k{i+2,j+1}(loc);
113             else
114                 %If not, interpolate between closest values
115                 Cd = C_k{i+2,j+1}(lbound)+(Kd-A_k{i+2,j+1}(lbound))*(...
116                     (C_k{i+2,j+1}(ubound)-C_k{i+2,j+1}(lbound))/...
117                     (A_k{i+2,j+1}(ubound)-A_k{i+2,j+1}(lbound)));
118             end
119             %% Calculate option value at previous node (i,j decreasing)
120             C_k{i+1,j+1}(k) = disf*(p*C_u + (1-p)*C_d);
121         end
122     end
123 end
124 disp(C_k{1,1})
125 toc;

```

Listing A.2: ../MATLAB Files/Functions/findInSorted.m

```

1 function [loc, A, B] = findInSorted(x,range,err)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% inputs %%%%%%%%%%%%%
3 % x = value to find %
4 % range = (sorted) vector to search %
5 % err = tolerance on search %
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %% Outputs
8 A = 1; %left boundary
9 B = numel(range); %right boundary
10 loc = 0; %initially set to 0 implying not found
11 %% Binary Search Algorithm
12 while B-A > 1
13     mid = floor((A+B)/2);
14     if x > range(mid)
15         B = mid;
16     else
17         A = mid;
18     end
19 end
20 %% Returning returned value if found (within tolerance)
21 if abs(range(A) - x) < err
22     loc = A;
23     return
24 elseif abs(range(B) - x) < err
25     loc = B;
26     return
27 end

```

# Appendix B: Python Files

All files can be found: <https://github.com/leele2/Mathematics-in-Business-Project/tree/master/Python%20Files>

Listing B.1: ../../Python Files/OptionVolume.py

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Jun  8 18:05:55 2022
4  Scraping and plotting data from theocc.com
5  @author: leelee2
6  """
7  import pandas as pd
8  import matplotlib as mpl
9  import matplotlib.pyplot as plt
10 import matplotlib.dates as mdates
11 from datetime import datetime
12 from bs4 import BeautifulSoup
13 from os import listdir
14 from pathlib import Path
15
16 def string_to_int(string):
17     if not string:
18         return 0
19     if string[0] == "$":
20         string = string[1:]
21     return int(string.replace(',', ''))
22
23 # Ensuring html file is on path
24 """
25 html file taken from:
26 https://www.theocc.com/Market-Data/Market-Data-Reports/Volume-and-Open-Interest/Historical-Volume-Statistics
27 """
28 html_file = [f for f in listdir('.') if f.endswith('.html')]
29 if len(html_file) != 1:
30     raise ValueError('should be exactly one html file in the current directory')
31 # Importing html file
32 with open(html_file[0], 'r', encoding='utf-8') as file:
33     soup = BeautifulSoup(file, 'xml')
34 # Finding Table
35 table = soup.find_all("table")
36 # Selecting Data
37 table_data = table[0].find_all("tr")
38 # Finding Table Names
39 table_names = []
40 for names in table_data[0]:
41     table_names.append(str(names.find("span").string))
42 del table_names[0]
43 del table_names[-1]
44 # Finding Headers
45 headers = []
46 for header in table_data[1]:
47     headers.append(str(header.find("span").string))
48 # Populating Data
49 Data = {}
50 for rows in table_data[2:]:
51     temp = {}
52     for i, data in enumerate(rows):
53         if i > 6 and i < 10 or i == 0:
54             temp[headers[i]] = data.get_text()
55             Date = datetime.strptime(temp["Date"], "%Y")
56             del temp["Date"]
57             Data[Date] = {k: string_to_int(v) for k, v in temp.items()}
58 # Converting data into pandas dataframes
59 Data = pd.DataFrame.from_dict(Data, orient="index")
60 # Plotting time-series
61 plt.rc('font', family='serif')
62 fig, ax1 = plt.subplots()
63 plt.gcf().set_size_inches(6, 4.5)
64 ax1.plot(Data['Options']/1e6, linewidth=2, zorder=1, label="Options")
65 # ax1.plot(Data[Data['Futures'] != 0]['Futures']/1e6, linewidth=2, zorder=1, label="Futures")
66 # ax1.legend()
67 ax1.set_title('Average Daily OptionS Trading Volume')
68 ax1.set_xlabel('Year')
69 ax1.xaxis.set_minor_formatter(mdates.DateFormatter("%Y"))
70 ax1.set_ylabel('Average Daily Volume (Millions)')
71 ax1.yaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.0f}M'))
72 fig.autofmt_xdate()
73 plt.tight_layout()
74 plt.show()
75 # Saving plot
76 cwd = str(Path(__file__).parent.parent.absolute())
77 fig.savefig(cwd + "\Latex_Files\Main\Chapters\C1\plots\OptionVolume.png", bbox_inches='tight')
```