Term Paper

Topic: The Effects of Policy Variables on Stock Market Return

1. Introduction:

To alleviate the credit crunch following the financial crisis of 2007, central banks in various countries, notably the United States, the United Kingdom, and the Eurozone, used a monetary strategy known as quantitative easing (QE). Under normal circumstances, a central bank's monetary policy entails setting a target interest rate and enforcing it by open market operations, which include buying and selling short-term government bonds (Curdia & Woodford, 2011). The central bank cannot, of course, lower interest rates further if the target interest rate is already exceedingly low, such as around 0 percent. Instead, it can credit its own bank account and spend the funds to purchase financial assets (typically government bonds) in order to boost the market's money supply. Quantitative easing is the name given to this strategy.

When a central bank decides to utilize quantitative easing, it purchases huge amounts of financial assets such as government and corporate bonds, as well as stocks. This seemingly insignificant decision has far-reaching consequences: When the amount of money in circulation in an economy rises, longer-term interest rates fall. This lowers borrowing costs, resulting in increased economic growth (Gagnon et al., 2010).

The goal of quantitative easing is to accomplish the following: By purchasing longer-term securities, a central bank hopes to lower longer-term market interest rates. When compared to the major tool used by central banks, The focus of standard rate policy is short-term market interest rates. When the Federal Reserve employs standard rate policy, the federal funds rate target is adjusted. The purpose is to sway the short-term interest rates that banks charge each other for overnight lending. For decades, the Federal Reserve has utilized interest rate policy to keep credit flowing and the US economy on track (Joyce et al., 2012).

When the federal funds rate fell to zero during the Great Recession, the Fed used quantitative easing (QE) to keep the economy from freezing up by buying mortgage-backed securities (MBS) and Treasuries (Mendez-Carbajo, 2020).

QE Timeline:

The Great Recession of 2007-09 and its aftermath marked the start of a new era in American monetary policy. The Federal Reserve's policy rate was over 500 basis points prior to 2008. The policy rate dropped fast to 25 basis points in 2008, and has remained there ever since (Huston &

Spencer, 2016). Prior to 2008, the Federal Reserve's balance sheet was less than 1 trillion dollar which is now approximately 7 trillion dollar and FED has done it with four timeline:

Quantitative Easing 1 (QE1) – In November 2008, the Federal Reserve ("the Fed") began purchasing mortgage-backed securities (MBS) from commercial banks in the amount of 600 billion USD. Excess liquidity was pumped into banks as a result of this operation (cash in their reserve accounts). The surplus liquidity pushed down short-term interest rates, also known as the federal funds rate, which are the borrowing rates between banks. The Federal Reserve had USD 1.75 trillion in bank debt, MBS, and Treasury notes on its balance sheet by March 2009. The Federal Reserve's assets peaked at USD 2.1 trillion in June 2010. The Fed bought more Treasuries at a USD 30 billion monthly rate as debt aged to keep its balance sheet above USD 2.0 trillion.

Quantitative Easing 2 (QE2) - The Fed launched a second phase of QE in November 2010. By the second quarter of 2011, the Fed has stated that it will purchase an extra USD 600 billion in Treasury securities. Operation Twist — The Federal Reserve declared in September 2011 that it would buy and sell short- and long-term bonds to affect policy goals. Long-term interest rates were lowered by Operation Twist. Short-term Treasury bonds were sold while long-term Treasury bonds were purchased by the Fed.

Quantitative Easing 3 (QE3) — In September 2012, the Federal Reserve announced QE3, an additional open-ended bond purchase program of USD 40 billion per month in agency MBS. The Fed also stated that short-term interest rates (the fed funds rate) will remain near zero until at least 2015. "QE-Infinity" is a nickname for QE3. The FOMC declared in December 2012 that open-ended purchases would be increased from USD 40 billion to USD 85 billion each month. The Fed would purchase USD 45 billion in Treasury bonds and $40 billion in mortgage-backed securities.

Quantitative Easing 4 (QE4) - The Federal Reserve stated on March 15, 2020 that it will purchase USD 500 billion in US Treasury bonds. It also plans to purchase USD 200 billion in MBS during the following few months

How QE Works:

One of the Fed's more powerful arrows is quantitative easing. Quantitative easing is a monetary policy instrument in which a central bank, such as the Federal Reserve, floods the market with cash to try to boost a slumping economy and avoid deflation (Williamson, 2017). The theory is that if the central bank injects enough currency into the market, it will start off a cascade of events that will include:

1. Banks and other financial institutions will amass ever-increasing cash reserves.
2. Banks will eventually opt to relax lending requirements in order to make use of their excess capital.
3. Individuals and businesses will begin to receive the loans they require.
4. As people and businesses begin to spend again, the economy will begin to revive.
5. If the economy expands, the unemployment rate will decrease.

In a speech in 2002, "Helicopter Ben" (Federal Reserve Chairman Ben Bernanke) stated that he would be willing to inject as much cash into the economy as was necessary to encourage growth. Quantitative easing entails dumping cash into the market. How does a central bank, such as the Federal Reserve, flood the market with currency, is the question (Selgin, 2018).

The central bank must take the following three measures to implement quantitative easing:

1. Reduce the short-term interest rate to 0%.
2. State how long the short-term interest rate will remain at zero percent.
3. Start investing in long-term securities such as Treasury bonds, corporate bonds, and asset-backed securities.

Moreover, the majority portion of the liabilities of Fed were: Currency in Circulation and vault cash. In addition, currency in circulation on an average used to consist 93% of the Fed’s Balance Sheet during the time. Bernanke with the unconventional monetary policy expand the policy in such a way that after 2008, the percentage of this initially fall below 40/5 and recently it has fall under 26%. That simply indicates that, Fed has a greater influence on the resource allocation of the financial markets. Thus, I would say that QE not only affects the stock market but also affects policy variables like: the unemployment rate, interest rate.

Thus, my objective in this term paper is to see the effects of policy variables on stock market return after introducing Quantitative Easing.

2. Literature Review:

The stock market is obviously affected by the Federal Reserve's quantitative easing (QE) program, albeit it is difficult to determine exactly how and to what extent. There appears to be a link between a QE policy and a growing stock market, according to the evidence. In reality, while a QE strategy was in place, some of the biggest stock market gains in US history occurred. After all, the goal of a quantitative easing strategy is to support or even accelerate a country's economic growth. In practice, QE includes purchasing large quantities of government bonds or other investments from banks in order to inject additional cash into the economy. Banks then lend that money to businesses, who use it to expand their operations and grow their sales. Stock investors buy stocks in anticipation of greater company revenue. That's the large picture; yet, there are additional, more subtle consequences of QE on stock market return (Al-Jassar & Moosa, 2019).

The unemployment rate and the stock market have a strong and direct relationship. It works like this: For the vast majority of Americans, the money they earn through their occupations is their principal source of income. This means that the amount of money people have is directly proportional to their employment – or lack thereof. The quantity of money people have based on their earnings is the most important factor in determining how much and where they spend their money. As you may expect, this has a significant economic impact. As a result, if the unemployment rate rises, general income (and thus spending power) will be limited. People spend less money when they have less money, and there is less demand for goods. As a result, stock prices may fall in a variety of areas due to a lack of demand for certain commodities (Krishnamurthy & Vissing-Jorgensen, 2011). Luxury objects, for example, will almost always depreciate in value during a downturn in the economy. People don't have money to buy jewels or expensive cars when they're just trying to get by. Simply put, a low unemployment rate might signal economic expansion, whilst a high unemployment rate can signal an underperforming economy. But it's more than that. The unemployment rate is seen as a key measure of the overall health of the economy. This isn't just because of the number of people out of work; it's also because of the ramifications for the economy, the dollar's worth, and the stock market. Many people are unaware that the size of unemployment rates may have an impact on the Federal Reserve. Depending on whether the economy is too hot or cold, they may raise or cut interest rates (Ross, 2015).

For good reason, the investment community and financial media obsess over interest rates. The cost of borrowing someone else's money is referred to as interest rates.When the Federal Open Market Committee (FOMC), which consists of seven Federal Reserve Board governors and five Federal Reserve Bank presidents, sets the target for the federal funds rate—the rate at which banks borrow from and lend to one another overnight—it has repercussions throughout the United States, including the stock market. While a change in the interest rate normally takes at least a year to have a wider economic impact, the stock market's reaction to a change is sometimes much faster.The Federal Reserve also sets a discount rate in addition to the federal funds rate. The discount rate is the rate at which the Federal Reserve charges banks who borrow money from it (Huston & Spencer, 2016; Krishnamurthy & Vissing-Jorgensen, 2011).The expected quantity of future cash flows will fall if a company is perceived as cutting back on its expansion or is less profitable—either through increased debt expenses or lower sales. If all other factors remain constant, this will result in a decrease in the company's stock price. If enough firms' stock prices fall, the entire market, or the main indices that many people associate with the market—the Dow Jones Industrial Average, S&P 500, and so on—will fall. Investors will not obtain as much growth from stock price rise if they have lower expectations for a company's growth and future cash flows. Stock ownership may become less appealing as a result of this. Furthermore, when compared to other assets, investing in shares can be perceived as unduly risky.Interest rate hikes, on the other hand, may boost some industries. The banking business is one of the sectors that benefits the most. Banks, brokerages, mortgage businesses, and insurance companies are all examples of financial institutions(Rudebusch, 2018).

From the literature review, my hypothesis are:

~ A positive relationship between total asset of Fed and stock market return

~ A negative relationship between unemployment rate and stock market return

~ A negative relationship between effective federal funds rate and stock market return

~ A negative relationship between currency in circulation/total asset and stock market return

3.Methodology

3.1 Regression Model (Ordinary Least Square Model)

The effect of policy variable on stock market in US was estimated using an Ordinary Least Square (OLS) estimation model. When the OLS assumptions hold, the estimates provided are the best estimates among the class of linear model estimation methods, according to the Gauss-Markov theorem. The model are specified below:

ln (Y) = α + β1 ln (total asset) + β2 Currency in Circulation/Total Asset + β3 Effective Federal Funds Rate+ β4 Unemployment Rate + u;

where Y is the dependent variable and that is ln (Stock Price) which I have considered S&P 500 here.

3.2 Variables

Dependent Variables: In this term paper, I have analyzed S&P 500 index and considered it as a dependent variable and Effective Federal Funds Rate, Unemployment Rate, Currency in Circulation/Total Asset of FED and total asset of Fed as independent variables.

3.3 Sample Size:

The term paper uses data from 2009 to 2021 which results in 262 observations.

3.4 Source of Data:

All the data I have used in this research were collected from the secondary sources. The relevant data as well as information were collected from Yahoo Finance and from Federal Reserve Economic Data. Different articles and literature relevant to this arena have also been reviewed.

4. Analysis & Result

In [ ]:
```
##import library

%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import io, base64, os, json, re
import pandas as pd
import pandas_datareader.data as web
import datetime
import numpy as np

import numpy as np
import pandas as pd
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt

import statsmodels.api as sm
from statsmodels.stats import diagnostic as diag
from statsmodels.stats.outliers_influence import variance_inflation_factor

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

import matplotlib.transforms as mtransforms
import datetime
import math
```

In [2]:
```
#Fred data

def gather_data(data_codes, start,
                end = datetime.datetime.today(), freq = "M"):
    i = 0
    # dct.items() calls key and value that key points to
    for key, val in data_codes.items():
        if i == 0:
            # Create dataframe for first variable, then rename column
            df = web.DataReader(val, "fred", start, end).resample(freq).mean()
            df.rename(columns = {val:key}, inplace = True)
            # setting i to None will cause the next block of code to execute,
            # placing data within df instead of creating a new dataframe for
            # each variable
            i = None
        else:
            # If dataframe already exists, add new column
            df[key] = web.DataReader(val, "fred", start, end).resample(freq).mear

    return df
```

In [3]:
```
data_codes  = {"Base: Total ($ Mil)": "BOGMBASE",
               "Base: Currency in Circulation ($ Mil)": "WCURCIR",
               # Assets
               "Balance Sheet: Total Assets ($ Mil)": "WALCL",
               "Balance Sheet Securities, Prem-Disc, Repos, and Loans ($ Mil)": '
               "Balance Sheet: Securities Held Outright ($ Mil)": "WSHOSHO",
               ### breakdown of securities holdings ###
               "Balance Sheet: U.S. Treasuries Held Outright ($ Mil)":"WSHOTSL",
               "Balance Sheet: Federal Agency Debt Securities ($ Mil)" : "WSHOFAD
               "Balance Sheet: Mortgage-Backed Securities ($ Mil)": "WSHOMCB",
               # other forms of lending
               "Balance Sheet: Repos ($ Mil)": "WORAL",
               "Balance Sheet: Central Bank Liquidity Swaps ($ Mil)" : "SWPT",
               "Balance Sheet: Direct Lending ($ Mil)" : "WLCFLL",
               # unamortized value of securities held (due to changes in interest
               "Balance Sheet: Unamortized Security Premiums ($ Mil)": "WUPSHO",
               # Liabilities
               "Balance Sheet: Total Liabilities ($ Mil)" : "WLTLECL",
               "Balance Sheet: Federal Reserve Notes Outstanding ($ Mil)" : "WLFN
               "Balance Sheet: Reverse Repos ($ Mil)": "WLRRAL",
               ### Major share of deposits
               "Balance Sheet: Excess Reserves ($ Mil)": "EXCSRESNW",
               "Balance Sheet: Required Reserves ($ Mil)": "RESBALREQW",
               "Balance Sheet: Total Reserves ($ Mil)": "WRESBAL",
               "Balance Sheet: Deposits from Dep. Institutions ($ Mil)":"WLODLL",
               "Balance Sheet: U.S. Treasury General Account ($ Mil)": "WDTGAL",
               "Balance Sheet: Other Deposits ($ Mil)": "WOTHLB",
               "Balance Sheet: All Deposits ($ Mil)": "WLDLCL",
               # Interest Rates
               "Federal Funds Target (Pre-Crisis)": "DFEDTAR",
               "Federal Funds (Upper) Target":"DFEDTARU",
               "Effective Federal Funds Rate":"DFF",
               "Interest on Excess Reserves":"IOER",
               # Req Reserves and Vault Cash
               "Vault Cash ($ Mil)": "TLVAULTW",
               "Vault Cash Used as Req. ($ Mil)": "VAULT",
               }

# Select start and end dates
start = datetime.datetime(2000, 1, 1)
end = datetime.datetime.today()

# freq refers to data frequency. Choose "D", "W", "M", "Q", "A"
# a number may also be place in front of a letter. "2D" indicates
#       alternating days
fed_data = gather_data(data_codes = data_codes, start = start,
                   end = end, freq = "M")
```

Load Data

In [4]:
```
fed_data
```

Out[4]:

DATE	Base: Total ($ Mil)	Base: Currency in Circulation ($ Mil)	Balance Sheet: Total Assets ($ Mil)	Balance Sheet Securities, Prem-Disc, Repos, and Loans ($ Mil)	Balance Sheet: Securities Held Outright ($ Mil)	Balance Sheet: U.S. Treasuries Held Outright ($ Mil)	Balance Sheet: Federal Agency Debt Securities ($ Mil)	Balan She Mortgag Back Securiti ($ M
2000-01-31	601900.0	594.67875	NaN	NaN	NaN	NaN	NaN	Na
2000-02-29	578000.0	566.14375	NaN	NaN	NaN	NaN	NaN	Na
2000-03-31	577100.0	563.70500	NaN	NaN	NaN	NaN	NaN	Na
2000-04-30	578600.0	564.73350	NaN	NaN	NaN	NaN	NaN	Na
2000-05-31	580600.0	565.83560	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	
2021-06-30	6027000.0	2178.84320	8026555.20	7874851.20	7450341.40	5149903.00	2347.0	2298091.
2021-07-31	6130200.0	2186.11750	8190356.75	8042917.00	7617340.75	5232755.50	2347.0	2382238.
2021-08-31	6328700.0	2187.61075	8291893.25	8153497.75	7733453.75	5312788.25	2347.0	2418318.
2021-09-30	6388900.0	2194.88100	8418612.40	8289740.40	7881596.20	5394942.60	2347.0	2484306.
2021-10-31	6331000.0	2202.48475	8516524.50	8393472.00	8000350.50	5481949.25	2347.0	2516054.

262 rows × 28 columns

In [55]:
```
# Load data

data=pd.read_csv("data 11.csv")
```

In [56]:
```
data
```

Out[56]:

	Date	Log Stock Price	Base: Currency in Circulation ($ Mil)	Balance Sheet: Total Assets ($ Mil)	Effective Federal Funds Rate	Adj Close	Log Total Asset	Currency in Circulation/Total Asset
0	1/31/2000	7.240263	594.68	0.00	5.45	1394.46	0.000000	0.000000
1	2/29/2000	7.219949	566.14	0.00	5.73	1366.42	0.000000	0.000000
2	3/31/2000	7.312273	563.71	0.00	5.85	1498.58	0.000000	0.000000
3	4/30/2000	7.280993	564.73	0.00	6.02	1452.43	0.000000	1.000000
4	5/31/2000	7.258835	565.84	0.00	6.27	1420.60	0.000000	0.000000
...	...	...	...	...	...	...	...	...
257	6/30/2021	8.365789	2178.84	8026555.20	0.08	4297.50	15.898266	0.000271
258	7/31/2021	8.388282	2186.13	8190356.75	0.10	4395.26	15.918468	0.000267
259	8/31/2021	8.416860	2187.63	8291893.25	0.09	4522.68	15.930789	0.000264
260	9/30/2021	8.368122	2194.90	8418612.40	0.08	4307.54	15.945956	0.000261
261	10/31/2021	8.434980	2202.51	8516524.50	0.08	4605.38	15.957519	0.000259

262 rows × 9 columns

In [7]:
```
data.dtypes
```

Out[7]:
```
Date                                      object
Log Stock Price                          float64
Base: Currency in Circulation ($ Mil)    float64
Balance Sheet: Total Assets ($ Mil)      float64
Effective Federal Funds Rate             float64
Adj Close                                float64
Log Total Asset                          float64
Currency in Circulation/Total Asset      float64
Unemployment Rate                        float64
dtype: object
```

In [57]:
```
data = data.set_index('Date')
```

In [58]:
```
data
```

Out[58]:

Date	Log Stock Price	Base: Currency in Circulation ($ Mil)	Balance Sheet: Total Assets ($ Mil)	Effective Federal Funds Rate	Adj Close	Log Total Asset	Currency in Circulation/Total Asset	Unem
1/31/2000	7.240263	594.68	0.00	5.45	1394.46	0.000000	0.000000	
2/29/2000	7.219949	566.14	0.00	5.73	1366.42	0.000000	0.000000	
3/31/2000	7.312273	563.71	0.00	5.85	1498.58	0.000000	0.000000	
4/30/2000	7.280993	564.73	0.00	6.02	1452.43	0.000000	1.000000	
5/31/2000	7.258835	565.84	0.00	6.27	1420.60	0.000000	0.000000	
...	...	...	...	...	...	...	...	
6/30/2021	8.365789	2178.84	8026555.20	0.08	4297.50	15.898266	0.000271	
7/31/2021	8.388282	2186.13	8190356.75	0.10	4395.26	15.918468	0.000267	
8/31/2021	8.416860	2187.63	8291893.25	0.09	4522.68	15.930789	0.000264	
9/30/2021	8.368122	2194.90	8418612.40	0.08	4307.54	15.945956	0.000261	
10/31/2021	8.434980	2202.51	8516524.50	0.08	4605.38	15.957519	0.000259	

262 rows × 8 columns

In [59]:
```
data_new = data.drop(['Base: Currency in Circulation ($ Mil)', 'Balance Sheet: To
```

In [60]:
```
data_new
```

Out[60]:

Date	Log Stock Price	Effective Federal Funds Rate	Log Total Asset	Currency in Circulation/Total Asset	Unemployment Rate
1/31/2000	7.240263	5.45	0.000000	0.000000	4.1
2/29/2000	7.219949	5.73	0.000000	0.000000	4.0
3/31/2000	7.312273	5.85	0.000000	0.000000	3.8
4/30/2000	7.280993	6.02	0.000000	1.000000	4.0
5/31/2000	7.258835	6.27	0.000000	0.000000	4.0
...	...	...	...	...	...
6/30/2021	8.365789	0.08	15.898266	0.000271	5.4
7/31/2021	8.388282	0.10	15.918468	0.000267	5.2
8/31/2021	8.416860	0.09	15.930789	0.000264	4.8
9/30/2021	8.368122	0.08	15.945956	0.000261	4.6
10/31/2021	8.434980	0.08	15.957519	0.000259	4.2

262 rows × 5 columns

In [61]:
```
data_new = data_new.loc['1/31/2009':'10/31/2021']
```

In [62]:
```
data_new
```

Out[62]:

Date	Log Stock Price	Effective Federal Funds Rate	Log Total Asset	Currency in Circulation/Total Asset	Unemployment Rate
1/31/2009	6.716449	0.15	14.525435	0.000436	8.3
2/28/2009	6.599993	0.22	14.447648	0.000475	8.7
3/31/2009	6.681946	0.18	14.501045	0.000454	9.0
4/30/2009	6.771718	0.15	14.568579	0.000425	9.4
5/31/2009	6.823438	0.18	14.573726	0.000424	9.5
...	...	...	...	...	...
6/30/2021	8.365789	0.08	15.898266	0.000271	5.4
7/31/2021	8.388282	0.10	15.918468	0.000267	5.2
8/31/2021	8.416860	0.09	15.930789	0.000264	4.8
9/30/2021	8.368122	0.08	15.945956	0.000261	4.6
10/31/2021	8.434980	0.08	15.957519	0.000259	4.2

154 rows × 5 columns

In [39]:
```
from statsmodels.tsa.stattools import adfuller
```

ADF Test

The row named "data_new" is my desired dataset with which I am going to work for the analysis and to check that whether my dataset is stationary or nonstationary, I will do the ADF test.

The hypotheses for the test:

The null hypothesis for this test is that there is a unit root.

The alternate hypothesis states that the time series is stationary (or trend-stationary)

In [46]:
```
## Check "Log Stock Price"

X = data_new["Log Stock Price"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -0.828252
p-value: 0.810682
Critical Values:
        1%: -3.474
        5%: -2.881
        10%: -2.577
Failed to Reject Ho - Time Series is Non-Stationary
```

In [47]:
```
## Check "Effective Federal Funds Rate"

X = data_new["Effective Federal Funds Rate"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -2.462716
p-value: 0.124811
Critical Values:
        1%: -3.476
        5%: -2.882
        10%: -2.577
Failed to Reject Ho - Time Series is Non-Stationary
```

In [48]:
```
## Check "Currency in Circulation/Total Asset"

X = data_new["Currency in Circulation/Total Asset"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -1.328545
p-value: 0.616014
Critical Values:
	1%: -3.474
	5%: -2.881
	10%: -2.577
Failed to Reject Ho - Time Series is Non-Stationary
```

In [49]:
```
## Check "Unemployment Rate"

X = data_new["Unemployment Rate"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -2.432458
p-value: 0.132784
Critical Values:
	1%: -3.474
	5%: -2.881
	10%: -2.577
Failed to Reject Ho - Time Series is Non-Stationary
```

From the above ADF test of each variable, I have observed that my dataset is non-stationary. If my dataset is not stationary, then the OLS result will not be consistent,t test and F test will not be valid and I might get spurious regression result and thus I need to check and correct my dataset.

In [63]:
```
###Modifying Dataset with "diff"

data_new.diff()
```

Out[63]:

Date	Log Stock Price	Effective Federal Funds Rate	Log Total Asset	Currency in Circulation/Total Asset	Unemployment Rate
1/31/2009	NaN	NaN	NaN	NaN	NaN
2/28/2009	-0.116457	0.07	-0.077786	0.000039	0.4
3/31/2009	0.081953	-0.04	0.053397	-0.000021	0.3
4/30/2009	0.089772	-0.03	0.067534	-0.000028	0.4
5/31/2009	0.051721	0.03	0.005147	-0.000001	0.1
...	...	...	...	...	...
6/30/2021	0.021971	0.02	0.020092	-0.000004	-0.5
7/31/2021	0.022493	0.02	0.020202	-0.000005	-0.2
8/31/2021	0.028578	-0.01	0.012321	-0.000003	-0.4
9/30/2021	-0.048738	-0.01	0.015167	-0.000003	-0.2
10/31/2021	0.066858	0.00	0.011563	-0.000002	-0.4

154 rows × 5 columns

In [64]:
```
data_stock =  data_new.diff()
```

In [65]:
```
data_stock
```

Out[65]:

Date	Log Stock Price	Effective Federal Funds Rate	Log Total Asset	Currency in Circulation/Total Asset	Unemployment Rate
1/31/2009	NaN	NaN	NaN	NaN	NaN
2/28/2009	-0.116457	0.07	-0.077786	0.000039	0.4
3/31/2009	0.081953	-0.04	0.053397	-0.000021	0.3
4/30/2009	0.089772	-0.03	0.067534	-0.000028	0.4
5/31/2009	0.051721	0.03	0.005147	-0.000001	0.1
...	...	...	...	...	...
6/30/2021	0.021971	0.02	0.020092	-0.000004	-0.5
7/31/2021	0.022493	0.02	0.020202	-0.000005	-0.2
8/31/2021	0.028578	-0.01	0.012321	-0.000003	-0.4
9/30/2021	-0.048738	-0.01	0.015167	-0.000003	-0.2
10/31/2021	0.066858	0.00	0.011563	-0.000002	-0.4

154 rows × 5 columns

In [66]:
```
data_updated = data_stock.dropna()
```

In [67]:
```
data_updated
```

Out[67]:

Date	Log Stock Price	Effective Federal Funds Rate	Log Total Asset	Currency in Circulation/Total Asset	Unemployment Rate
2/28/2009	-0.116457	0.07	-0.077786	0.000039	0.4
3/31/2009	0.081953	-0.04	0.053397	-0.000021	0.3
4/30/2009	0.089772	-0.03	0.067534	-0.000028	0.4
5/31/2009	0.051721	0.03	0.005147	-0.000001	0.1
6/30/2009	0.000196	0.03	-0.036956	0.000017	0.0
...	...	...	...	...	...
6/30/2021	0.021971	0.02	0.020092	-0.000004	-0.5
7/31/2021	0.022493	0.02	0.020202	-0.000005	-0.2
8/31/2021	0.028578	-0.01	0.012321	-0.000003	-0.4
9/30/2021	-0.048738	-0.01	0.015167	-0.000003	-0.2
10/31/2021	0.066858	0.00	0.011563	-0.000002	-0.4

153 rows × 5 columns

In [19]:
```
data_updated.isnull().sum()
```

Out[19]:
```
Log Stock Price                        0
Effective Federal Funds Rate           0
Log Total Asset                        0
Currency in Circulation/Total Asset    0
Unemployment Rate                      0
dtype: int64
```

|| As I have set my dataset with diff, now I will again run the ADF test in order to check the stationarity in the dataset ||

In [40]:
```
## Check "Log Stock Price"

X = data_updated["Log Stock Price"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -10.397420
p-value: 0.000000
Critical Values:
	1%: -3.474
	5%: -2.881
	10%: -2.577
Reject Ho - Time Series is Stationary
```

In [41]:
```
## Check "Effective Federal Funds Rate"

X = data_updated["Effective Federal Funds Rate"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -3.452057
p-value: 0.009307
Critical Values:
	1%: -3.475
	5%: -2.881
	10%: -2.577
Reject Ho - Time Series is Stationary
```

In [42]:
```
## Check "Log Total Asset"

X = data_updated["Log Total Asset"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -7.482122
p-value: 0.000000
Critical Values:
	1%: -3.474
	5%: -2.881
	10%: -2.577
Reject Ho - Time Series is Stationary
```

In [43]:
```
## Check "Currency in Circulation/Total Asset"

X = data_updated["Currency in Circulation/Total Asset"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -7.705484
p-value: 0.000000
Critical Values:
        1%: -3.474
        5%: -2.881
        10%: -2.577
Reject Ho - Time Series is Stationary
```

In [44]:
```
## Check "Unemployment Rate"

X = data_updated["Unemployment Rate"].values
result = adfuller(X)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

if result[0] < result[4]["5%"]:
    print ("Reject Ho - Time Series is Stationary")
else:
    print ("Failed to Reject Ho - Time Series is Non-Stationary")
```

```
ADF Statistic: -9.669131
p-value: 0.000000
Critical Values:
        1%: -3.474
        5%: -2.881
        10%: -2.577
Reject Ho - Time Series is Stationary
```

Checking for Heteroscedasticity

One of the assumptions of the model is that there is no heteroscedasticity and thus to give a simple definition it merely means the standard errors of a variable, monitored over a specific amount of time, are non-constant.

According to the Gauss–Markov theorem, if the errors in the linear regression model are uncorrelated, have equal variances, and an expectation value of zero, the ordinary least squares (OLS) estimator has the minimum sampling variance of all linear unbiased estimators (Wooldridge 2015). To check if the independent variables were highly correlated with each other, we can do several tests and one of them is known as Breuschpagan test. The OLS assumes that the conditional variance is homoscedastic. Hence, the Breuschpagan test was used to check for heteroskedasticity.

To check for heteroscedasticity, the null hypothesis for both the White's test and the Breusch-Pagan test is that the variances for the errors are equal:

• H0 = σ2i = σ2

The alternate hypothesis is that the variances are not equal:

• H1 = σ2i ≠ σ2

The goal is to fail to reject the null hypothesis. Results from the tests as presented below indicate that there is absence of heteroskedasticity.

In [68]:
```
X = data_updated.drop('Log Stock Price', axis = 1)
Y = data_updated[['Log Stock Price']]

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.20, random_


regression_model = LinearRegression()


regression_model.fit(X_train, y_train)
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

Out[68]:
```
LinearRegression(normalize=False)
```

In [69]:
```
# let's grab the coefficient of our model and the intercept
intercept = regression_model.intercept_[0]
coefficent = regression_model.coef_[0][0]

print("The intercept for our model is {:.4}".format(intercept))
print('-'*100)

# loop through the dictionary and print the data
for coef in zip(X.columns, regression_model.coef_[0]):
    print("The Coefficient for {} is {:.2}".format(coef[0],coef[1]))
```

```
The intercept for our model is 0.01259
--------------------------------------------------------------------------------
--------------------
The Coefficient for Effective Federal Funds Rate is -0.016
The Coefficient for Log Total Asset is -0.62
The Coefficient for Currency in Circulation/Total Asset is -2.9e+03
The Coefficient for Unemployment Rate is -0.019
```

In [28]:
```
## Breuschpagan Test

_, pval, __, f_pval = diag.het_breuschpagan(est.resid, est.model.exog)
print(pval, f_pval)
print('-'*100)

# print the results of the test
if pval > 0.05:
    print("For the Breusch-Pagan's Test")
    print("The p-value was {:.4}".format(pval))
    print("We fail to reject the null hypthoesis, so there is no heterosecdastici

else:
    print("For the Breusch-Pagan's Test")
    print("The p-value was {:.4}".format(pval))
    print("We reject the null hypthoesis, so there is heterosecdasticity.")
```

```
0.3512441342490251 0.3576171816128195
--------------------------------------------------------------------------------
--------------------
For the Breusch-Pagan's Test
The p-value was 0.3512
We fail to reject the null hypthoesis, so there is no heterosecdasticity.
```

Checking for Autocorrelation

Autocorrelation is a characteristic of data in which the correlation between the values of the same variables is based on related objects. It violates the assumption of instance independence, which underlies most of conventional models.

When you have a series of numbers, and there is a pattern such that values in the series can be predicted based on preceding values in the series, the set of numbers is said to exhibit autocorrelation. This is also known as serial correlation and serial dependence. It generally exists in those types of data-sets in which the data, instead of being randomly selected, are from the same source.

To test the autocorrelation:

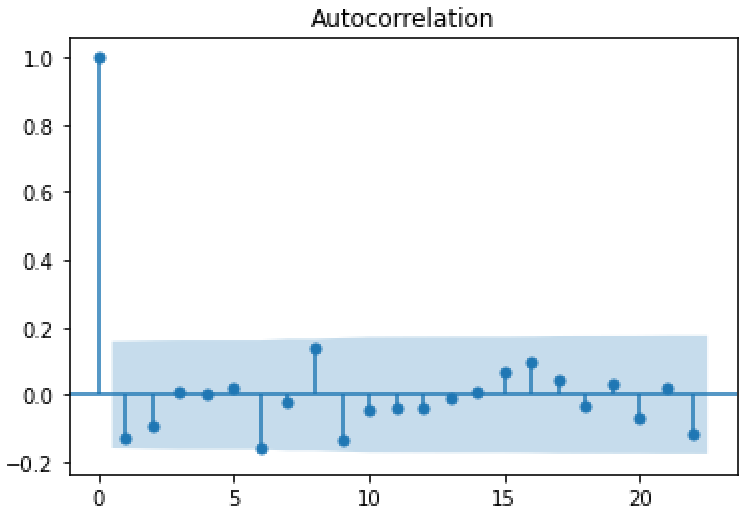
• H0: The data are random.

• Ha: The data are not random.

That means I want to fail to reject the null hypothesis, have a large p-value because then it means I will have no autocorrelation.

In [29]:
```
# test for autocorrelation
from statsmodels.stats.stattools import durbin_watson

# calculate the lag, optional
lag = min(10, (len(X)//5))
print('The number of lags will be {}'.format(lag))
print('-'*100)

# run the Ljung-Box test for no autocorrelation of residuals
# test_results = diag.acorr_breusch_godfrey(est, nlags = lag, store = True)
test_results = diag.acorr_ljungbox(est.resid, lags = lag)

# grab the p-values and the test statistics
ibvalue, p_val = test_results

# print the results of the test
if min(p_val) > 0.05:
    print("The lowest p-value found was {:.4}".format(min(p_val)))
    print("We fail to reject the null hypthoesis, so there is no autocorrelation.
    print('-'*100)
else:
    print("The lowest p-value found was {:.4}".format(min(p_val)))
    print("We reject the null hypthoesis, so there is autocorrelation.")
    print('-'*100)

# plot autocorrelation
sm.graphics.tsa.plot_acf(est.resid)
plt.show()
```

```
The number of lags will be 10
--------------------------------------------------------------------------------
--------------------
The lowest p-value found was 0.1065
We fail to reject the null hypthoesis, so there is no autocorrelation.
--------------------------------------------------------------------------------
--------------------

C:\Users\proma.gupta\Anaconda3\lib\site-packages\statsmodels\stats\diagnostic.p
y:559: FutureWarning: The value returned will change to a single DataFrame afte
r 0.12 is released.  Set return_df to True to use to return a DataFrame now.  S
et return_df to False to silence this warning.
  warnings.warn(msg, FutureWarning)
```



Summary Statistics

In [21]:
```
# get the summary
desc_df = data_updated.describe()

# add the standard deviation metric
desc_df.loc['+3_std'] = desc_df.loc['mean'] + (desc_df.loc['std'] * 3)
desc_df.loc['-3_std'] = desc_df.loc['mean'] - (desc_df.loc['std'] * 3)

# display it
desc_df
```

Out[21]:

	Log Stock Price	Effective Federal Funds Rate	Log Total Asset	Currency in Circulation/Total Asset	Unemployment Rate
count	153.000000	153.000000	153.000000	1.530000e+02	153.000000
mean	0.011232	-0.000458	0.009360	-1.157529e-06	-0.026797
std	0.041633	0.107375	0.031180	1.068846e-05	0.908440
min	-0.133668	-0.930000	-0.077786	-9.568000e-05	-2.200000
25%	-0.007526	-0.010000	-0.001927	-3.443000e-06	-0.200000
50%	0.017726	0.000000	0.002383	8.780000e-07	-0.100000
75%	0.035100	0.010000	0.016727	3.022000e-06	0.100000
max	0.119421	0.240000	0.310113	3.948700e-05	10.400000
+3_std	0.136132	0.321668	0.102900	3.090786e-05	2.698524
-3_std	-0.113667	-0.322583	-0.084180	-3.322292e-05	-2.752119

Summary statistics of variables used in the model for the analysis are presented above. The statistics computed are mean, standard deviation (Std. Dev), minimum (Min) and maximum (Max), outliers with number of observations being 262. Here, the Table shows that the mean effective federal funds rate was -0.000458 with the maximum of 0.24 and the minimum of -0.93 . That means that usually Therefore, a negative interest rate environment occurs when the nominal interest rate drops below 0% for a specific economic zone. This effectively means that banks and other financial firms have to pay to keep their excess reserves stored at the central bank, rather than receiving positive interest income. From the analysis, average total asset occupied by FED was approximately $2,644,019(million). The average unemployment rate of US is -0.027% with the maximum of 10.40% and the minimum of -2.20%. The stock price (ln) also showed an average of 1.1% return with the maximum of 12%.

In [22]:
```
# filter the data frame to remove the values exceeding 3 standard deviations [out
data_remove_df = data_updated[(np.abs(stats.zscore(data_updated)) < 3).all(axis=1

# what rows were removed
data_updated.index.difference(data_remove_df.index)
```

Out[22]:
```
Index(['2/28/2009', '3/31/2020', '4/30/2020'], dtype='object', name='Date')
```

The above data represents the outliers. However, the outliers value came as the consequences of intoducing and executing Quantitative Easing.

Ordinary Least Squares Regression

Build Model

In [32]:
```
y_var = ["Log Stock Price"]
x_vars = ["Log Total Asset",
          "Currency in Circulation/Total Asset",
          "Effective Federal Funds Rate",
          "Unemployment Rate"]
reg_vars = y_var + x_vars
reg_data = data[reg_vars].dropna()
reg_data
```

Out[32]:

Date	Log Stock Price	Log Total Asset	Currency in Circulation/Total Asset	Effective Federal Funds Rate	Unemployment Rate
1/31/2000	7.240263	0.000000	0.000000	5.45	4.1
2/29/2000	7.219949	0.000000	0.000000	5.73	4.0
3/31/2000	7.312273	0.000000	0.000000	5.85	3.8
4/30/2000	7.280993	0.000000	1.000000	6.02	4.0
5/31/2000	7.258835	0.000000	0.000000	6.27	4.0
...	...	...	...	...	...
6/30/2021	8.365789	15.898266	0.000271	0.08	5.4
7/31/2021	8.388282	15.918468	0.000267	0.10	5.2
8/31/2021	8.416860	15.930789	0.000264	0.09	4.8
9/30/2021	8.368122	15.945956	0.000261	0.08	4.6
10/31/2021	8.434980	15.957519	0.000259	0.08	4.2

262 rows × 5 columns

In [35]:
```
## Summary statistics of regression model

reg_data.describe().round(2)
```

Out[35]:

	Log Stock Price	Log Total Asset	Currency in Circulation/Total Asset	Effective Federal Funds Rate	Unemployment Rate
count	262.00	262.00	262.00	262.00	262.00
mean	7.38	12.71	0.52	1.65	5.97
std	0.42	5.05	2.34	1.89	1.95
min	6.60	0.00	0.00	0.05	3.50
25%	7.07	13.61	0.00	0.13	4.60
50%	7.25	14.68	0.00	1.01	5.45
75%	7.65	15.29	0.00	2.39	6.90
max	8.43	15.96	16.00	6.54	14.80

Here, the Table shows that The stock price (ln) had an average of 7.38% return with the maximum of 8.43%. The mean effective federal funds rate was 1.65% with the maximum of 6.54% and the minimum of 0.05. From the analysis, average ln (total asset) occupied by FED was approximately

12.71. The average unemployment rate of US is 5.97% with the maximum of 14.80% and the minimum of 3.50%.

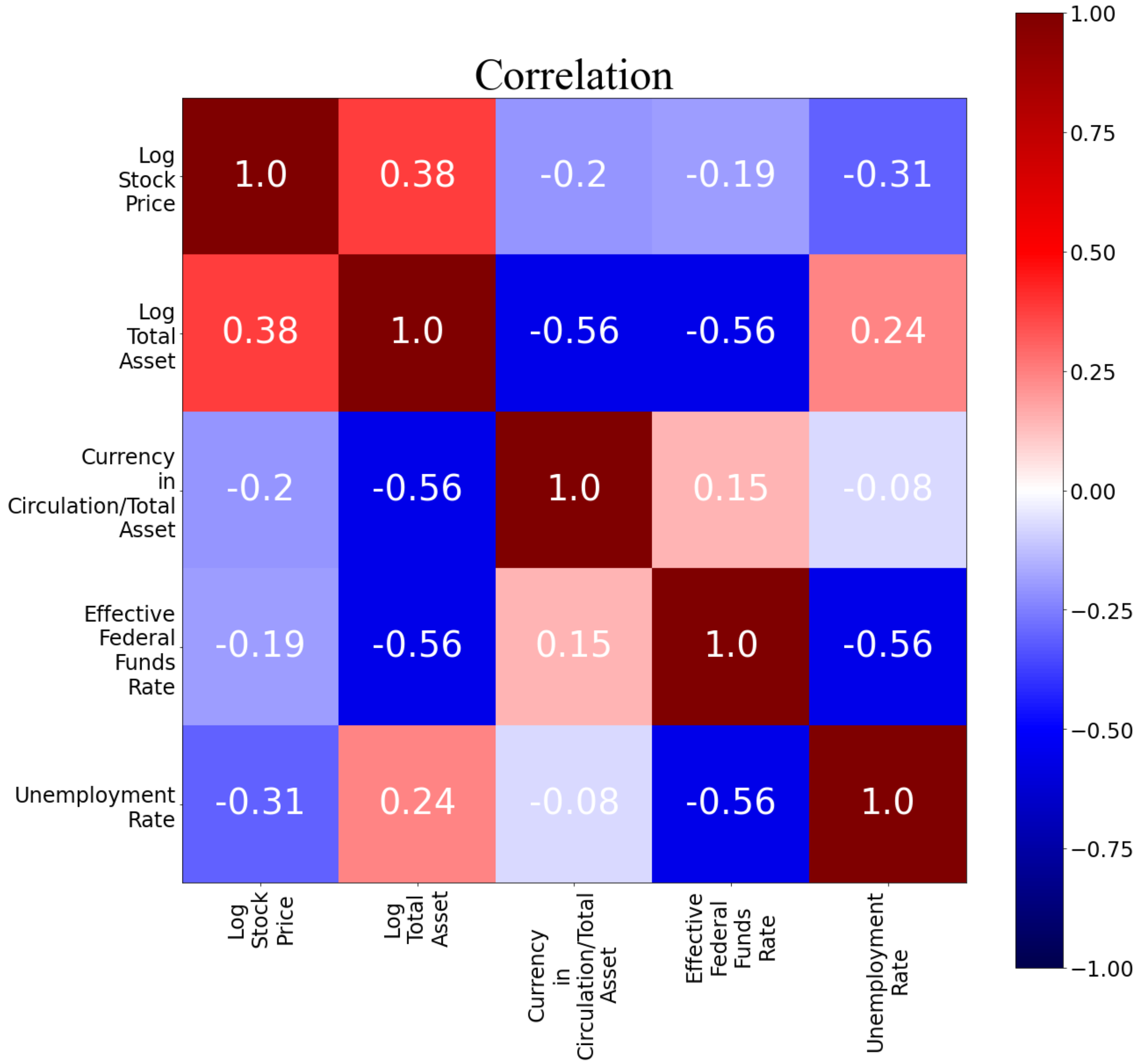
Correlation Analysis

In [33]:
```
reg_data.corr()
```

Out[33]:

	Log Stock Price	Log Total Asset	Currency in Circulation/Total Asset	Effective Federal Funds Rate	Unemployment Rate
Log Stock Price	1.000000	0.379568	-0.204783	-0.193854	-0.311260
Log Total Asset	0.379568	1.000000	-0.560952	-0.555311	0.235350
Currency in Circulation/Total Asset	-0.204783	-0.560952	1.000000	0.147557	-0.077404
Effective Federal Funds Rate	-0.193854	-0.555311	0.147557	1.000000	-0.561607
Unemployment Rate	-0.311260	0.235350	-0.077404	-0.561607	1.000000

In [34]:
```
from datlib.plots import *
corr_matrix_heatmap(reg_data.corr())
```



Correlation analysis is performed to find out the relationship between the variables that is how the independent variables affect the dependent variable. From the table and also from the figure, it can be stated that federal funds rate, unemployment rate and proportion of currency in circular to total asset has a negative relationship with the return of S&P 500. In comparison with these three variables, unemployment rate has a strong negative affect of 31% followed by currency in circular to total asset of 20% and federal funds rate of 19% which means that if these variables go up, then the return on stock market will go down. On the other hand, total asset of balance sheet has a positive relationship with stock price of 38%.

In [36]:
```
y = reg_data[y_var]
X = reg_data[x_vars]
X["Constant"]=1
results = sm.OLS(y,X).fit()
results.summary()
```

Out[36]:

OLS Regression Results

Dep. Variable:	Log Stock Price	**R-squared:**	0.370
Model:	OLS	**Adj. R-squared:**	0.360
Method:	Least Squares	**F-statistic:**	37.71
Date:	Fri, 17 Dec 2021	**Prob (F-statistic):**	8.20e-25
Time:	10:12:30	**Log-Likelihood:**	-81.411
No. Observations:	262	**AIC:**	172.8
Df Residuals:	257	**BIC:**	190.7
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Log Total Asset	0.0258	0.006	4.232	0.000	0.014	0.038
Currency in Circulation/Total Asset	-0.0042	0.011	-0.385	0.701	-0.026	0.017
Effective Federal Funds Rate	-0.0754	0.016	-4.713	0.000	-0.107	-0.044
Unemployment Rate	-0.1238	0.013	-9.582	0.000	-0.149	-0.098
Constant	7.9149	0.141	56.248	0.000	7.638	8.192

Omnibus:	47.332	**Durbin-Watson:**	0.073
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	94.752
Skew:	0.917	**Prob(JB):**	2.66e-21
Kurtosis:	5.305	**Cond. No.**	103.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

From the results of the OLS regression, R-squared was 0.370 which indicates that 37% of the variation in the stock return of S&P 500 can be explained by the model. The F-statistic of the model was significant at 1% indicating joint significance of the independent variables . The results indicate that the independent variables were all significant at 1% significance level. The model estimated homoskedasticity and no auto correlation which will eliminate the issue of heteroskedasticity and auto correlation in order to make sound inference. From the results of the OLS, the effect of total asset of FED represents that 1% increase in total asset is estimated to increase the return of stock price by 0.02% with the 1% significance level. The currency in circulation/total asset from the calculation of model represents that 1$ increase due to the change of the proportion of currency in circulation on total asset is estimated to decrease the retrn of the stock market by 0.4% with a very low significance level. Acccordingly, federal funds rate and unemployment rate also showed a negative relationship with stock market of 7% and 12% respectively with the 1% significance level. how long a firm has been operating has not a significant effect on accessing return on equity at 1% significance level.

5. Conclusion

The unconventional monetary policy was embraced as a solution to the financial crisis in 2008. Some critics question the effectiveness of Qe and as per them, there is a risk of Inflation due to it and also Quantitative easing can cause the stock market to boom, and stock ownership is concentrated among Americans who are already well-off, crisis or not. However, keeping these criticism aside, my findings are consistent with the intuition that the change in monetary policy has had a significant impact on economic activity.

6. References

Al-Jassar, S. A., & Moosa, I. A. (2019). The effect of quantitative easing on stock prices: a structural time series approach. Applied Economics, 51(17), 1817-1827.

Curdia, V., & Woodford, M. (2011). The central-bank balance sheet as an instrument of monetarypolicy. Journal of Monetary Economics, 58(1), 54-79.

Gagnon, J., Raskin, M., Remache, J., & Sack, B. (2011). The financial market effects of the Federal Reserve’s large-scale asset purchases. International Journal of Central Banking, 7(1), 45-52.

Gagnon, J., Raskin, M., Remache, J., & Sack, B. P. (2010). Large-scale asset purchases by the Federal Reserve: did they work? FRB of New York Staff Report(441).

Huston, J. H., & Spencer, R. W. (2016). The wealth effects of quantitative easing. Atlantic Economic Journal, 44(4), 471-486.

Joyce, M., Miles, D., Scott, A., & Vayanos, D. (2012). Quantitative easing and unconventional monetary policy–an introduction. The Economic Journal, 122(564), F271-F288.

Krishnamurthy, A., & Vissing-Jorgensen, A. (2011). The effects of quantitative easing on interest rates: channels and implications for policy.

Mendez-Carbajo, D. (2020). Temporary Open Market Operations and Large-Scale Asset Purchases. Page One Economics®.

Newman, R. (2012). Why Wall Street Loves Quantitative Easing. US News & World Report September, 12.

Ross, S. (2015). How Does Quantitative Easing in the US Affect the Stock Market? In.

Rudebusch, G. (2018). A review of the fed’s unconventional monetary policy. FRBSF Economic Letter, 27, 1-5.

Selgin, G. A. (2018). Floored!: How a Misguided Fed Experiment Deepened and Prolonged the E Great Recession. Cato Institute.

Williamson, S. D. (2017). Quantitative easing: How well does this tool work? The Regional Economist, 25(3).