1. Introduction

1) Assumption

This study assumes that fluctuations in AAPL's stock price are a strong predictor of movements in the GSPC.

2) Objective

The primary objective of this study is to explore the influence of Apple Inc.'s stock price on the S&P 500 Index from 2000-01-01 to 2020-12-31 (Monthly basis) using regression analysis.

3) Background

The S&P 500 Index (GSPC): is a critical barometer of the U.S. stock market, encompassing 500 of the largest companies by market capitalization and representing about 80% of the market's value. It is widely regarded as the best single gauge of large-cap U.S. equities.

Apple Inc. (AAPL): a technology behemoth known for its innovation in consumer electronics and software, is a key component of this index. AAPL's market capitalization and influence make it a significant player whose performance can impact the GSPC's overall movement.

Understanding the relationship between AAPL's stock price and the GSPC is essential. This study aims to quantify the impact of AAPL on the GSPC, providing insights into how individual company dynamics can influence major stock market indices.

2. Methodology

1) Data Collection Method:

The data collection for this project involved several key steps, ensuring secure and efficient retrieval of relevant financial data.

Data Sources: WRDS (Wharton Research Data Services) database was used as the primary data source. Google Finance and Yahoo Finance were utilized for supplementary data.

2) Project Initial Setup:

1. **Programming Language:** Installed Python for scripting and data analysis.

2. **IDE Setup:** Set up Visual Studio Code (VS Code) for code development.

3. **Libraries Installed: Command**: 'pip install wrds pandas numpy statsmodels matplotlib'
    a. **wrds**: for accessing WRDS data.
    b. **pandas**: for data manipulation.
    c. **statsmodels**: for statistical analysis.
    d. **matplotlib**: for data visualization.

4. **Configuring .pgpass File:** Configured .pgpass file. This file, stored securely, allows the Python script to access the database without manual authentication each time.

    Ran this python script in VS Code:

```
1   import wrds
2   db = wrds.Connection(wrds_username='nikhil_01a')
3   db.create_pgpass_file()
```

We are prompted once for the WRDS username and password on the first login, at the Connection() step, but after that we don't have to enter the password again:

```
(base) Nikhils-MBP:Python-Stock-Data-Analysis nikhilparmar$ python -u "/Users/nikhilpa
Enter your WRDS username [nikhil_01a]:nikhil_01a
Enter your password:
WRDS recommends setting up a .pgpass file.
Create .pgpass file now [y/n]?: y
Created .pgpass file successfully.
You can create this file yourself at any time with the create_pgpass_file() function.
Loading library list...
Done
```

3. Data Collection and Analysis: Python Code

1) Code Description:

The Python script connects to the WRDS database, retrieves AAPL and GSPC data, processes it, and performs a linear regression analysis to explore the influence of AAPL's stock price on the GSPC.

2) Code Interpretation:

A. Importing libraries and connecting to WRDS:

```
import wrds
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

# Connect to WRDS
db = wrds.Connection()
```

B. Data Collection using SQL Queries:

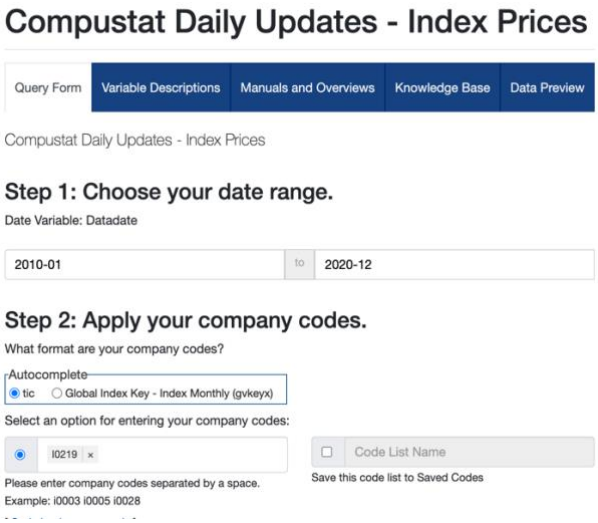
```
stock_symbol = 'AAPL'
index_gvkeyx = '165157'
start_date = '2000-01-01'
end_date = '2020-12-31'
```

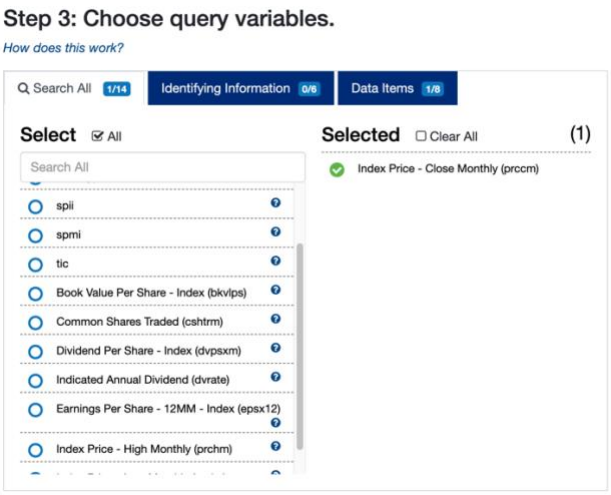
Found **Apple**'s stock symbol from **Google finance** and **S&P 500** gvkeyx from **WRDS**.

Steps to obtain S&P 500 gvkeyx:

1. Login to WRDS -> Get Data -> Compustat – Capital IQ -> North America -> Index Prices.

2. Then follow these steps:





3. Submit form as excel or csv: Here in the excel you will find the gvkeyx in 1st Column:

	Global Index Key - Index Monthly	Index Price - Close Monthly	Data Date - Index Monthly
2	165157	513.833591	1/31/10
3	165157	526.467995	2/28/10
4	165157	559.058688	3/31/10

C. SQL Queries to retrieve data:

1. **AAPL Data**: A SQL query fetched Apple Inc.'s (AAPL) daily closing prices over a specified period. The query selected the date and closing price from the appropriate dataset within WRDS.

```
apple_query = f"""SELECT date, prc as price
            FROM crsp.msf
            WHERE permno in (
              SELECT permno
              FROM crsp.msenames
              WHERE ticker = '{stock_symbol}'
            )
            AND date >= '{start_date}' AND date <= '{end_date}'"""
```

Query Explanation:

1. **Select Data Fields:** Retrieves the date and price (prc) from the CRSP Monthly Stock File (crsp.msf) in the WRDS database.
2. **Filter by Ticker**: Uses a subquery to find the permno (a unique identifier for stocks in CRSP) that matches Apple Inc.'s ticker symbol (stock_symbol).
3. **Date Range**: Specifies the data collection period by setting a start (start_date) and end date (end_date).
4. **Purpose**: The query is designed to extract Apple's stock price data within the given date range, ensuring the retrieved data is specific to Apple Inc.

2. **GSPC Data**: Similarly, another SQL query retrieved the daily closing prices for the S&P 500 Index (GSPC), aligning with the same dates as the AAPL data.

```
gspc_query = f"""SELECT datadate as date, prccm as price
            FROM comp.idx_mth
            WHERE gvkeyx = '{index_gvkeyx}'
            AND datadate >= '{start_date}' AND datadate <= '{end_date}'"""
```

Query Explanation:

1. **Select Data Fields:** Retrieves the date (datadate) and closing price (prccm) from the comp.idx_mth table, which contains monthly index data.
2. **Index Identification:** Filters the data for the S&P 500 Index using the provided Global Vantage Key for Index (gvkeyx), which uniquely identifies the GSPC in the database.
3. **Time Frame Specification:** Specifies the date range for data retrieval with a start (start_date) and end date (end_date).
4. **Purpose:** This query is tailored to extract the monthly closing price data for the S&P 500 Index within the specified time period.

Executing these queries and closing the WRDS connection:

```
apple_data = db.raw_sql(apple_query)
gspc_data = db.raw_sql(gspc_query)

db.close() # Close the WRDS connection.
```

D. Data Analysis:

```
# Ensure the dates are in datetime format
apple_data['date'] = pd.to_datetime(apple_data['date'])
gspc_data['date'] = pd.to_datetime(gspc_data['date'])

# Merge the datasets on date
merged_data = pd.merge(apple_data, gspc_data, on='date', suffixes=('_apple', '_gspc'))

# Defining independant and dependant variables
X = merged_data['price_apple'] # Independant
y = merged_data['price_gspc'] # Dependant

# Add a constant to the independent variable
X = sm.add_constant(X)

# Perform the regression
model = sm.OLS(y, X).fit()

# Print out the statistics
print(model.summary())

# Plot the regression
plt.scatter(X['price_apple'], y)
plt.plot(X['price_apple'], model.predict(X), color='red')
plt.title('Regression of GSPC Monthly Closing Price on Apple Monthly Closing Price')
plt.xlabel('AAPL Closing Price')
plt.ylabel('GSPC Closing Price')
plt.show()
```

Explanation:

1. Date Formatting: Converts 'date' columns in Apple and GSPC datasets to Python's datetime format.
2. Data Merging: Merges Apple and GSPC data on the 'date' column, differentiating columns using suffixes '_apple' and '_gspc'.
3. Variable Definition: Defines Apple's closing price as the independent variable (X) and GSPC's closing price as the dependent variable (y).
4. Regression Setup: Adds a constant to the independent variable (X) to include an intercept in the regression model.
5. Performing Regression and Output: Executes the linear regression, prints the statistical summary, and visualizes the relationship with a scatter plot and regression line.

4. Results

1) Python Console Result:

OLS Regression Results

Dep. Variable:	price_gspc	R-squared:	0.029
Model:	OLS	Adj. R-squared:	0.024
Method:	Least Squares	F-statistic:	5.306
Date:	Tue, 19 Dec 2023	Prob (F-statistic):	0.0224
Time:	20:08:46	Log-Likelihood:	-1223.7
No. Observations:	179	AIC:	2451.
Df Residuals:	177	BIC:	2458.
Df Model:	1		
Covariance Type:	nonrobust		

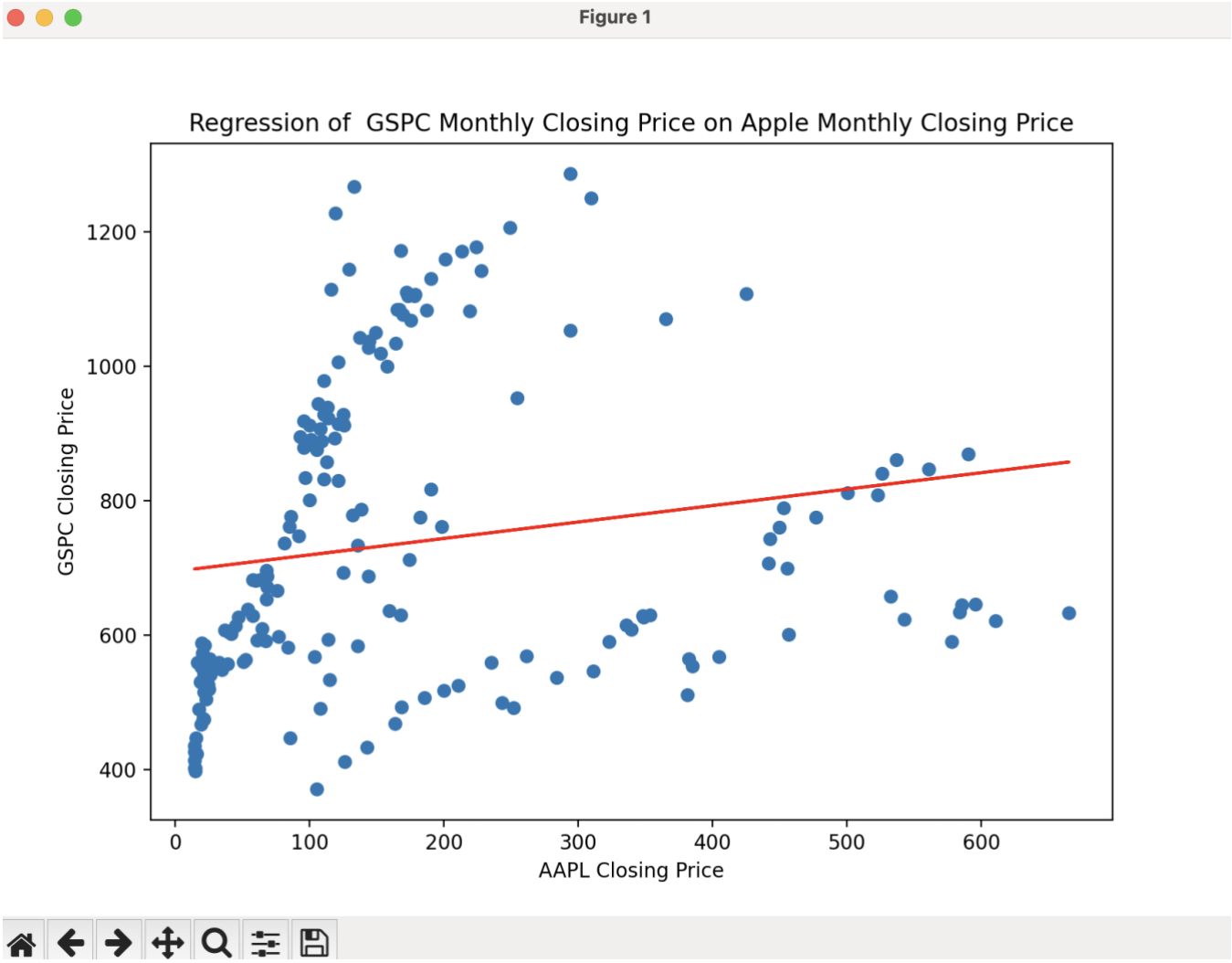
	coef	std err	t	P>|t|	[0.025	0.975]
const	694.7514	25.305	27.455	0.000	644.813	744.690
price_apple	0.2444	0.106	2.304	0.022	0.035	0.454

Omnibus:	23.959	Durbin-Watson:	0.035
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16.641
Skew:	0.625	Prob(JB):	0.000243
Kurtosis:	2.182	Cond. No.	356.

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

The regression analysis indicates a statistically significant but small positive relationship between Apple Inc.'s stock price and the S&P 500 Index, with an R-squared value of 0.029. This implies that changes in Apple's stock price are associated with changes in the S&P 500 Index, but Apple's stock only explains about 2.9% of the variation in the index's price. The low R-squared suggests that other factors besides Apple's stock price have a more substantial impact on the S&P 500 Index's price movements.

2) Graphical Plot Result:



The plot visualizes the regression analysis of the S&P 500 Index (GSPC) monthly closing prices against Apple Inc.'s (AAPL) monthly closing prices. Each point represents a month's paired closing prices of AAPL and GSPC. The red line depicts the regression line of best fit and indicates a slight positive relationship between AAPL's price and the GSPC's price. Despite this, the spread of the points suggests a weak correlation, aligning with the low R-squared value from the regression results.