**Stock Market Analysis using Python**

Stock Market Analysis means analyzing the current and historical trends in the stock market to make future buying and selling decisions. Stock market analysis is one of the best use cases of Data Science in finance.

## **Stock Market Analysis using Python**

To analyze the stock market, we will collect the stock price data of Google. At the end of this article, you will learn to analyze the stock market interactively using the Python programming language. Let’s start by collecting the stock price data of Google. We will use the **[yfinance](https://pypi.org/project/yfinance/" \t "_blank)** API of Yahoo Finance for collecting the stock price data.

Now here’s how to collect Google’s stock price data:

import pandas as pd

import yfinance as yf

import datetime

from datetime import date, timedelta

import plotly.graph\_objects as go

import plotly.express as px

today = date.today()

d1 = today.strftime("%Y-%m-%d")

end\_date = d1

d2 = date.today() - timedelta(days=365)

d2 = d2.strftime("%Y-%m-%d")

start\_date = d2

data = yf.download('GOOG',

                      start=start\_date,

                      end=end\_date,

                      progress=False)

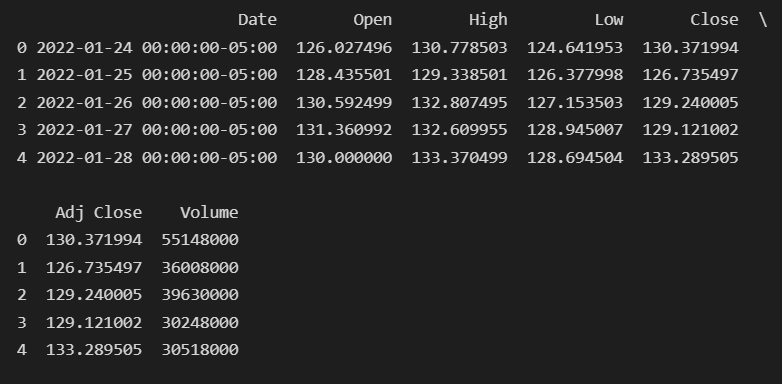
data["Date"] = data.index

data = data[["Date", "Open", "High", "Low",

             "Close", "Adj Close", "Volume"]]

data.reset\_index(drop=True, inplace=True)

print(data.head())



Whenever we analyze the stock market, always start with a candlestick chart. A candlestick chart is a handy tool to analyze the price movements of stock prices. Here’s how we can visualize a candlestick chart of Google’s stock prices:

figure = go.Figure(data=[go.Candlestick(x=data["Date"],

                                        open=data["Open"], high=data["High"],

                                        low=data["Low"], close=data["Close"])])

figure.update\_layout(title = "Google Stock Price Analysis", xaxis\_rangeslider\_visible=False)

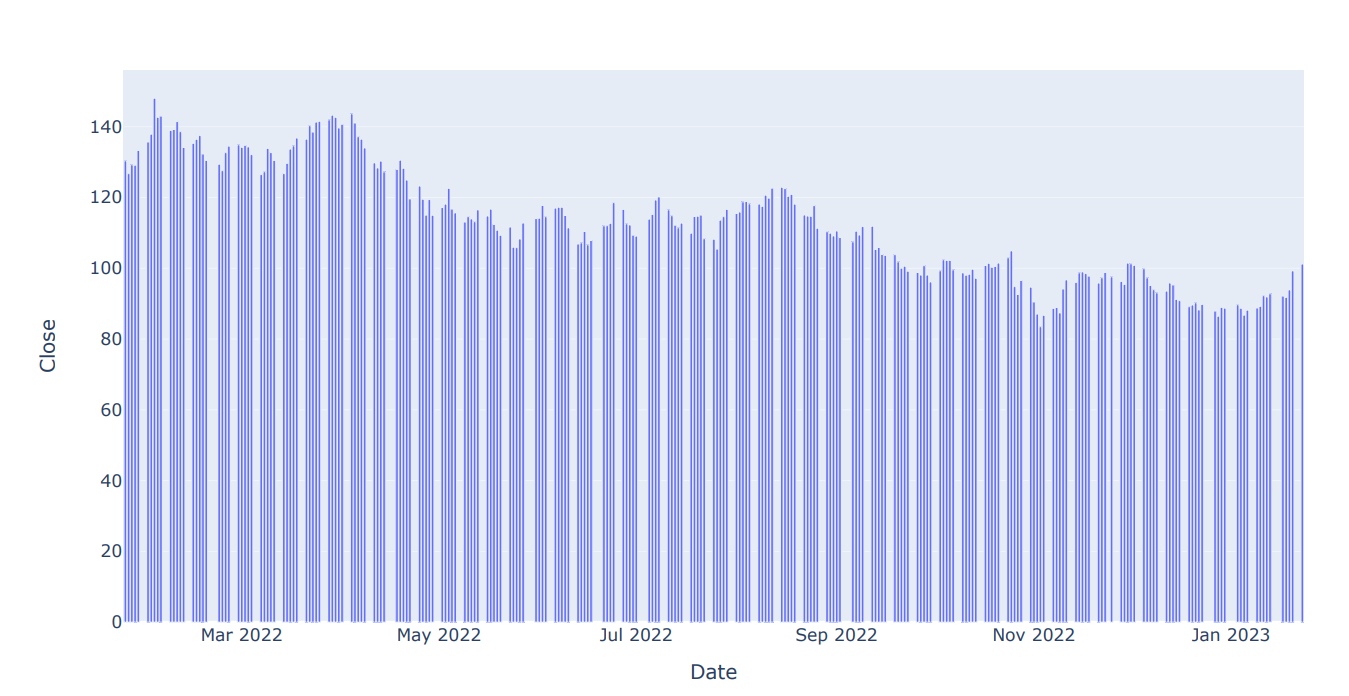
figure.show()



A bar plot is also a handy visualization to analyze the stock market, specifically in the long term. Here’s how to visualize the close prices of Google’s stock using a bar plot:

figure = px.bar(data, x = "Date", y= "Close")

figure.show()



One of the valuable tools to analyze the stock market is a range slider. It helps us to analyze the stock market between two specific points by interactively selecting the time period. Here’s how we can add a range-slider to analyze the stock market:

figure = px.line(data, x='Date', y='Close',

                 title='Stock Market Analysis with Rangeslider')

figure.update\_xaxes(rangeslider\_visible=True)

figure.show()



Another interactive feature we can add for stock market analysis is time period selectors. Time period selectors are like buttons that show us the graph of a specific time period. For example, a year, three months, six months, etc. Here is how we can add buttons for selecting the time period for stock market analysis:

figure = px.line(data, x='Date', y='Close',

                 title='Stock Market Analysis with Time Period Selectors')

figure.update\_xaxes(

    rangeselector=dict(

        buttons=list([

            dict(count=1, label="1m", step="month", stepmode="backward"),

            dict(count=6, label="6m", step="month", stepmode="backward"),

            dict(count=3, label="3m", step="month", stepmode="backward"),

            dict(count=1, label="1y", step="year", stepmode="backward"),

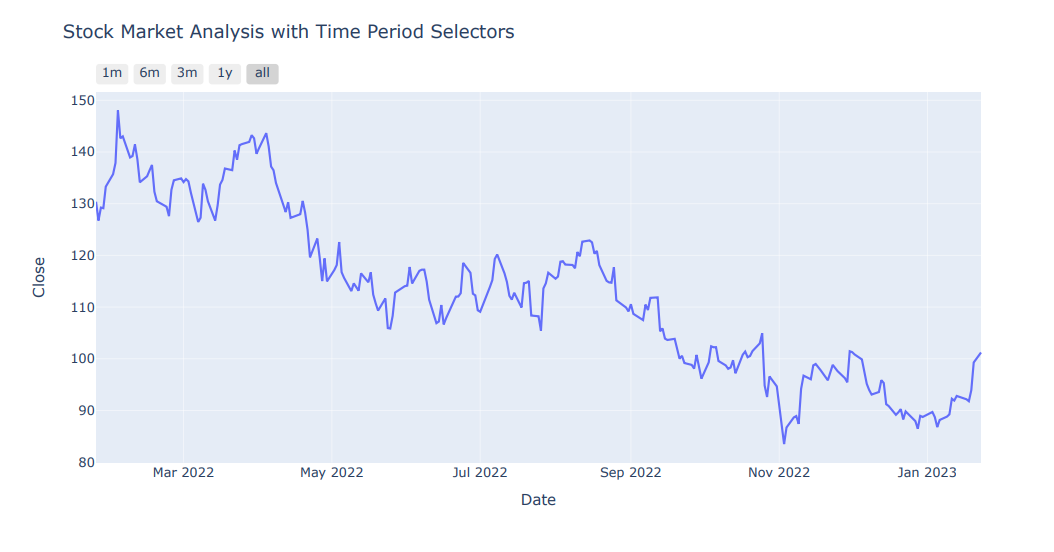
            dict(step="all")

        ])

    )

)

figure.show()



The weekend or holiday season always affects the stock market. So if we want to remove all the records of the weekend trends from our stock market visualization, below is how we can do it:

figure = px.scatter(data, x='Date', y='Close', range\_x=['2021-07-12', '2023-01-24'],

                 title="Stock Market Analysis by Hiding Weekend Gaps")

figure.update\_xaxes(

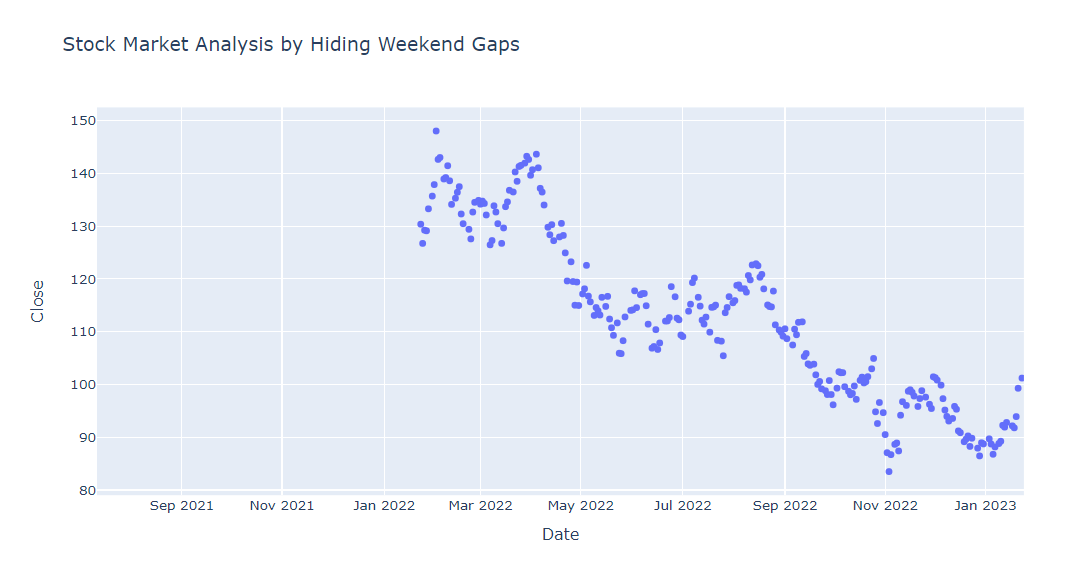
    rangebreaks=[

        dict(bounds=["sat", "sun"])

    ]

)

figure.show()



So that’s how we can analyze the stock market using Python.

### **Summary**

So this is how we can use the Python programming language to analyze the stock market interactively. Stock Market Analysis means analyzing the current and historical trends in the stock market to make future buying and selling decisions.