



# 빅데이터분석: 캔들차트

Younghoon Kim  
(nongaussian@hanyang.ac.kr)

**STOCK PRICE 분석하기**

# 오늘 그려볼 그래프



\* 출처: <https://python.plainenglish.io/a-simple-guide-to-plotly-for-plotting-financial-chart-54986c996682>

# 데이터 읽어들이기

- 코드 → 수업 홈페이지에서 복붙

```
import pandas as pd
import os, sys
from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv('drive/MyDrive/MSFT.csv')
df = df.drop(columns=["Adj Close"])

dt_all=pd.date_range(start=df.iloc[0]['Date'],
                     end=df.iloc[-1]['Date'])

dt_breaks=[d for d in dt_all
            if d.strftime("%Y-%m-%d") not in df['Date'].to_list()]
```

# 데이터 들여보기

날짜: 년-월-일 형식

일일최고가

최저가

시가

종가

거래량

Date

High

Low

Open

Close

Volume

Adj Close

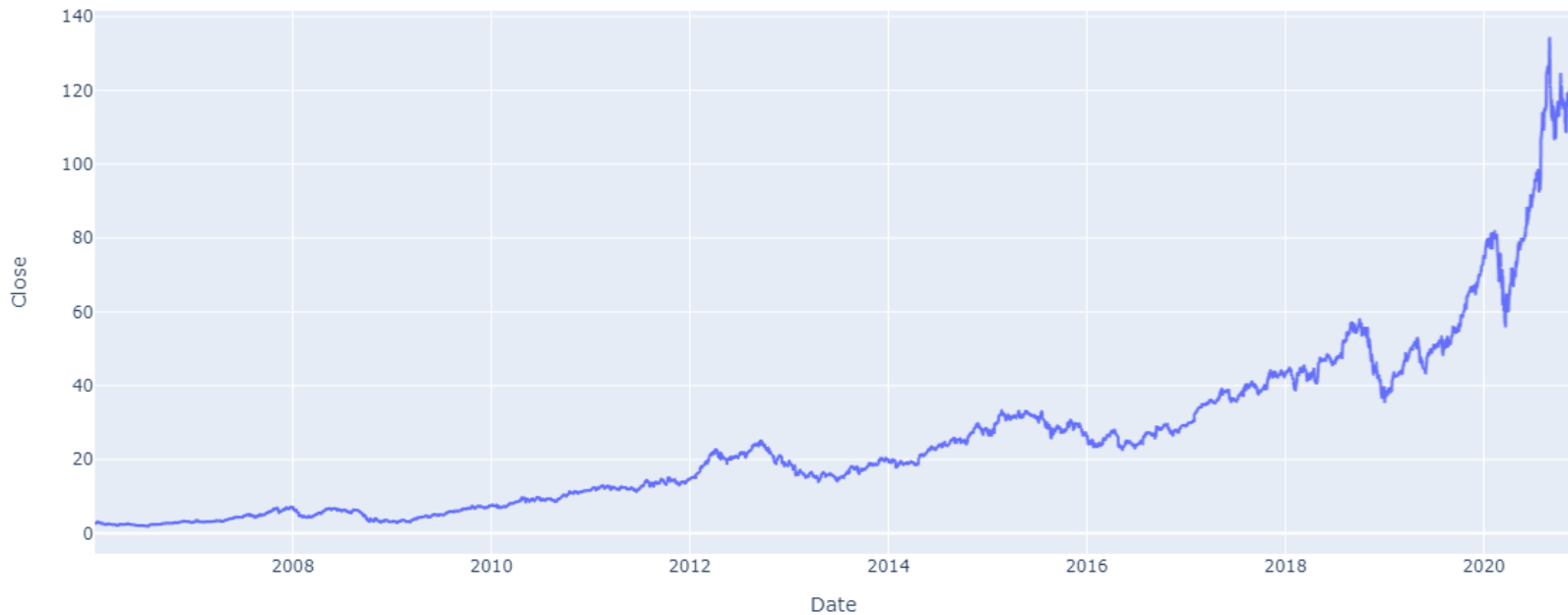
0	2006-01-03	2.669643	2.580357	2.585000	2.669643	807234400.0	2.284267
1	2006-01-04	2.713571	2.660714	2.683214	2.677500	619603600.0	2.290990
2	2006-01-05	2.675000	2.633929	2.672500	2.656429	449422400.0	2.272961
3	2006-01-06	2.739286	2.662500	2.687500	2.725000	704457600.0	2.331634
4	2006-01-09	2.757143	2.705000	2.740357	2.716071	675040800.0	2.323994
...	...	...	...	...	...	...	...
3744	2020-11-16	120.989998	118.150002	118.919998	120.300003	91183000.0	120.300003
3745	2020-11-17	120.669998	118.959999	119.550003	119.389999	74271000.0	119.389999
3746	2020-11-18	119.820000	118.000000	118.610001	118.029999	76322100.0	118.029999

# 첫 그래프: AAPL 종가 그리기



```
import plotly.express as px

fig = px.line(df, x='Date', y="Close")
fig.show()
```





# plotly.graph\_objects를 이용해 그래프 그리기



```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
```

```
graph = go.Scatter(x=df['Date'], y=df["Close"])
```

```
fig = make_subplots()
fig.add_trace(graph)
fig.show()
```





# plotly.graph\_objects를 이용해 그래프 그리기

```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
```

```
graph = go.Scatter(x=df['Date'], y=df["Close"])
```

```
fig = make_subplots()
fig.add_trace(graph)
fig.show()
```

그래프 여러 개를 조합하기 위한 API

비교

```
import plotly.express as px
```

```
fig = px.line(df, x='Date', y="Close")
fig.show()
```



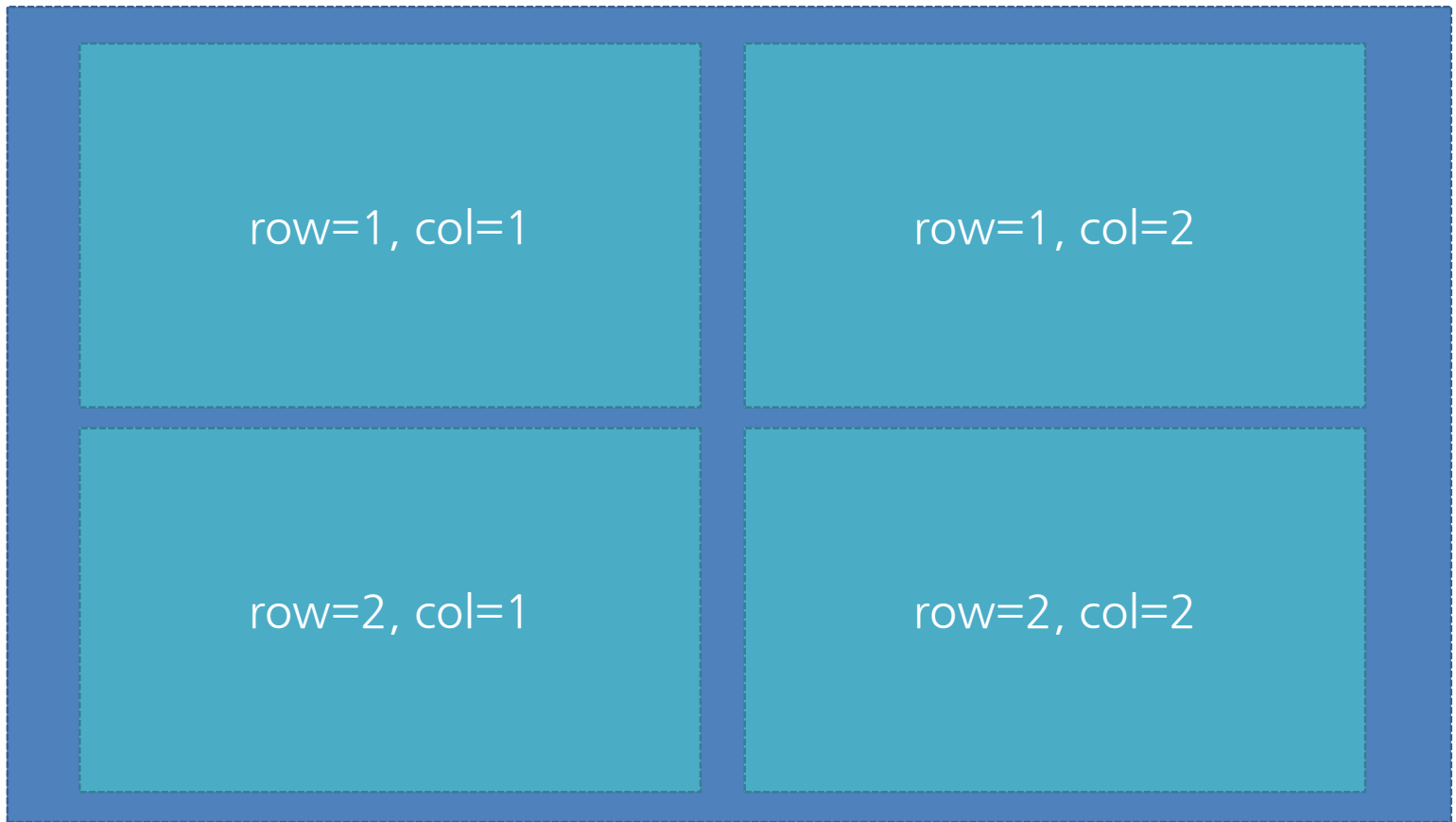


# make\_subplots?

- 그래프를 그릴 캔버스를 만든다!

cols = 2

rows = 2



fig



# 종가와 거래량 함께 그리기

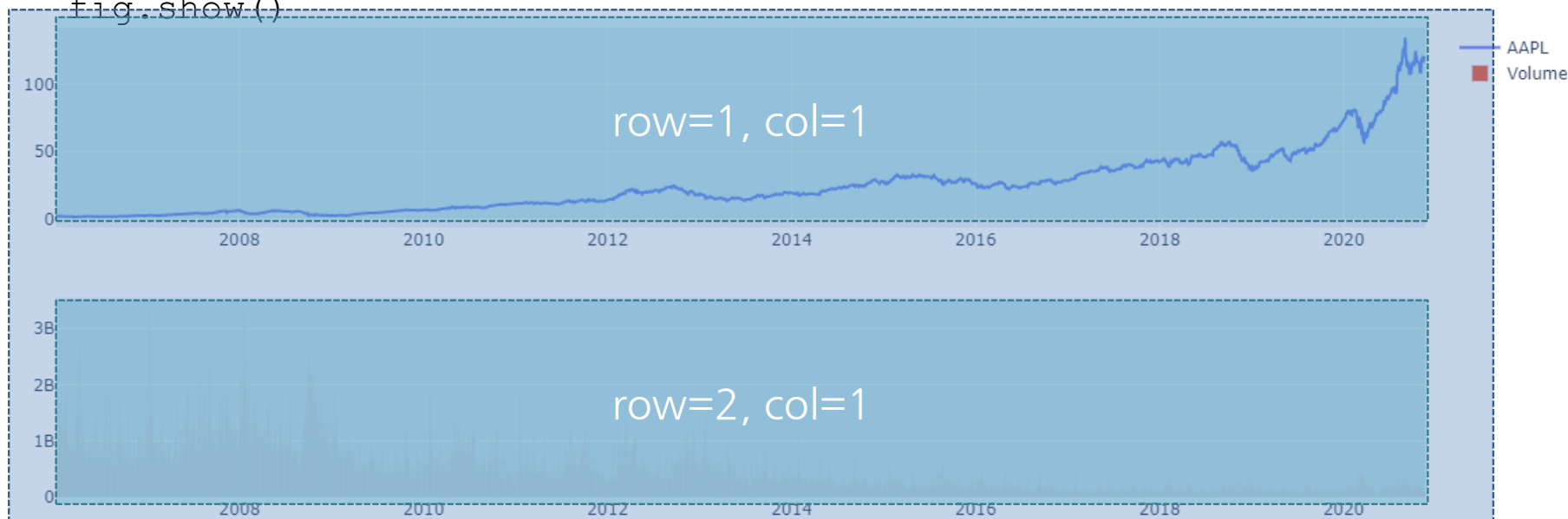
```
fig = make_subplots(rows=2, cols=1)
```

```
graph1 = go.Scatter(x=df['Date'], y=df["Close"], name='AAPL')  
graph2 = go.Bar(x=df['Date'], y=df["Volume"], name='Volume')
```

```
fig.add_trace(graph1, row=1, col=1)
```

```
fig.add_trace(graph2, row=2, col=1)
```

```
fig.show()
```



fig



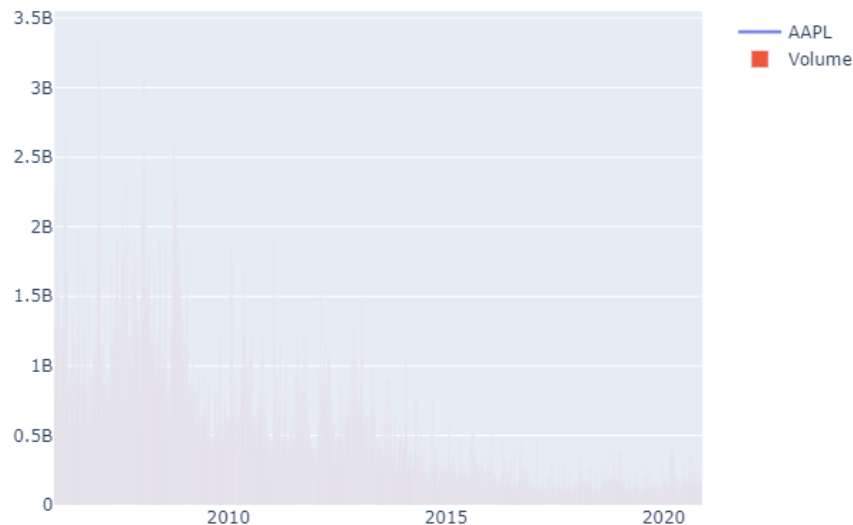
# 종가와 거래량 함께 그리기

```
fig = make_subplots(rows=1, cols=2)
```

```
graph1 = go.Scatter(x=df['Date'], y=df["Close"], name='AAPL')  
graph2 = go.Bar(x=df['Date'], y=df["Volume"], name='Volume')
```

```
fig.add_trace(graph1, row=1, col=1)  
fig.add_trace(graph2, row=1, col=2)
```

```
fig.show()
```

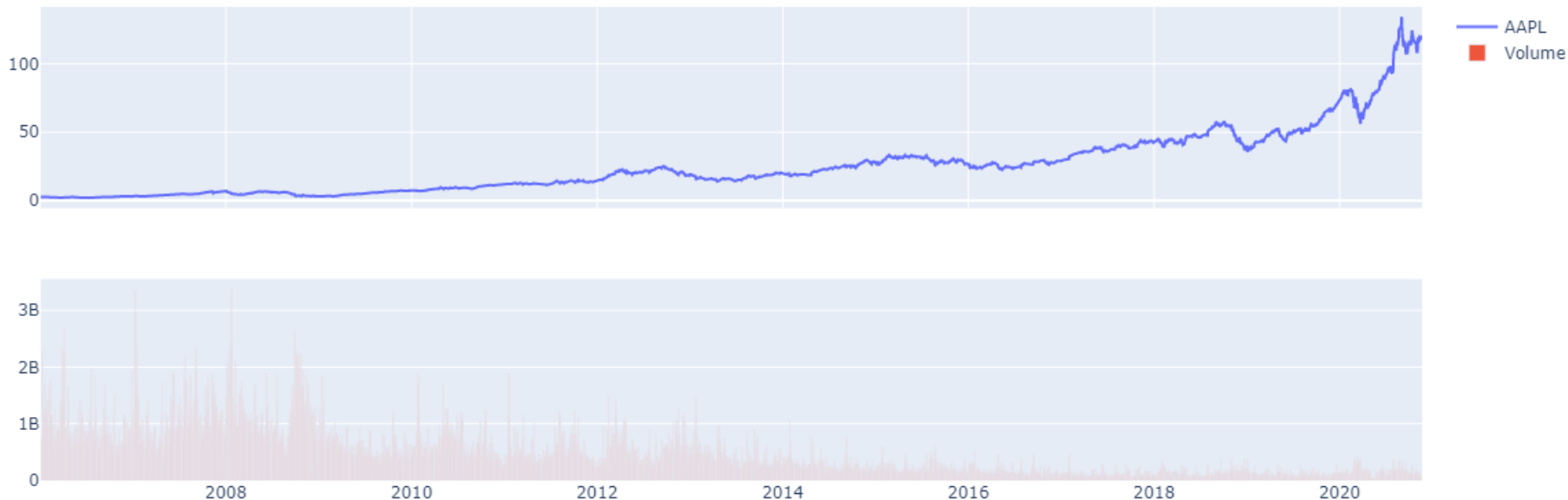


# X축 레이블 공유

```
fig = make_subplots(rows=2, cols=1, shared_xaxes=True)

graph1 = go.Scatter(x=df['Date'], y=df["Close"], name='AAPL')
graph2 = go.Bar(x=df['Date'], y=df["Volume"], name='Volume')

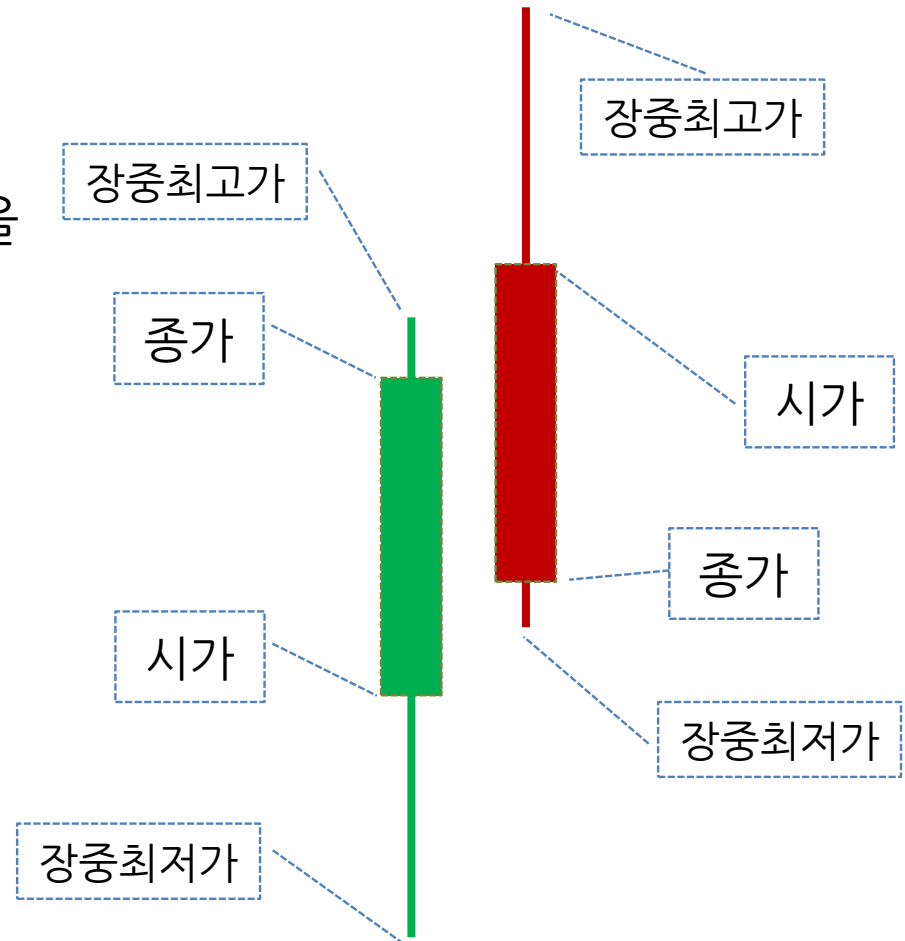
fig.add_trace(graph1, row=1, col=1)
fig.add_trace(graph2, row=2, col=1)
fig.show()
```



# 캔들차트 그리기

## ■ 캔들차트란?

- 사각형을 이용해 시가, 종가, 최고가, 최저가를 같이 표시
- 상승/하락을 사각형의 색깔을 이용해 시각화





# 캔들차트 그리기

```
fig = make_subplots()
```

```
graph = go.Candlestick(x=df['Date'],  
    open=df['Open'],  
    high=df['High'],  
    low=df['Low'],  
    close=df['Close'])  
fig.add_trace(graph)  
fig.show()
```

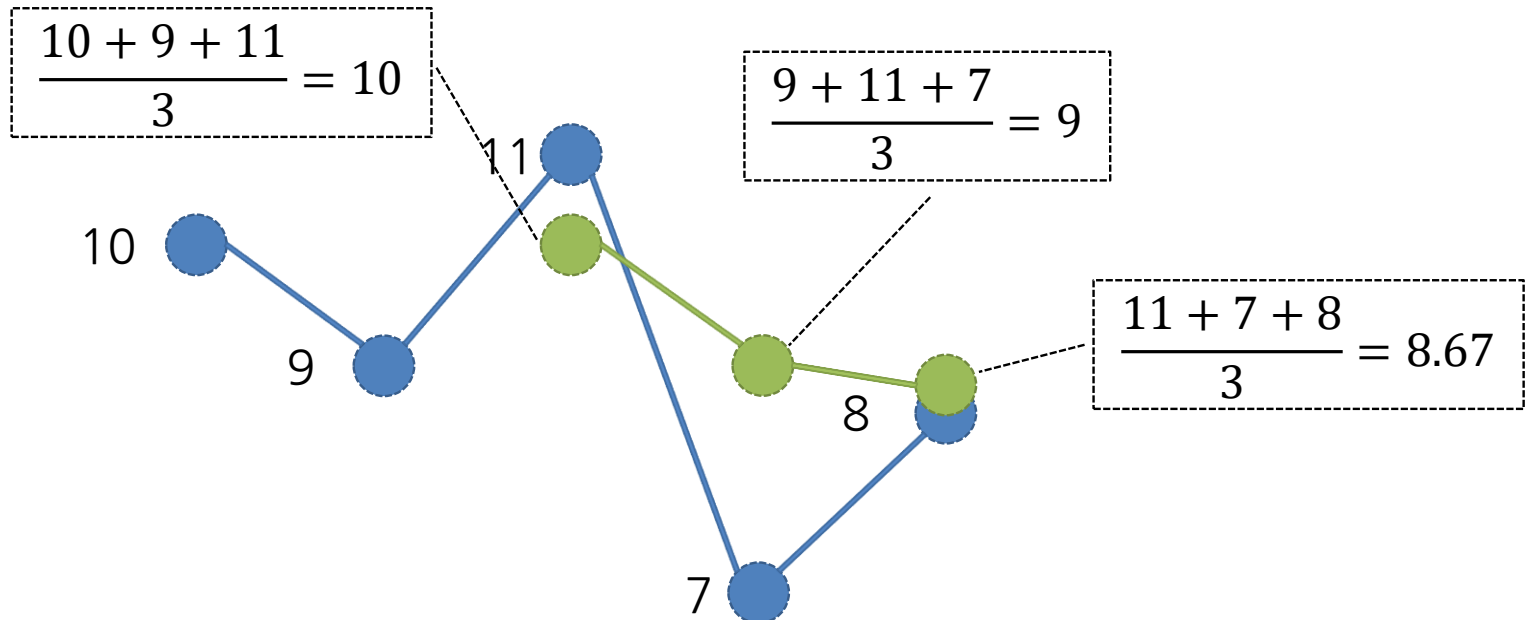
**범위 슬라이드:**  
시작과 끝 범위를 지정  
해서 확대해보세요.



# 이동평균

## ■ 이동평균이란?

- 매 날짜로부터 N-1일 전까지의, 총 N일의 값의 평균
- 예: 3일 이동평균





# .rolling() 함수

```
df["MA05"] = df['Close'].rolling(window=5).mean()
```

1. window 번 한 줄 씩 shift시키면서 데이터를 복사
2. .mean() 함수를 통해 평균값 계산

```
df['Close'].rolling(window=5).mean()
```

```
0          NaN
1          NaN
2          NaN
3          NaN
4      2.688929
```

...

```
3744    118.846001
3745    119.530000
3746    119.238000
3747    119.124001
3748    118.739999
```

```
Name: Close, Length: 3749, dtype: float64
```



# 그래프 그리기

```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
```

```
df["MA05"] = df['Close'].rolling(window=5).mean()
df["MA20"] = df['Close'].rolling(window=20).mean()
df["MA60"] = df['Close'].rolling(window=60).mean()
```

```
fig = make_subplots()
```

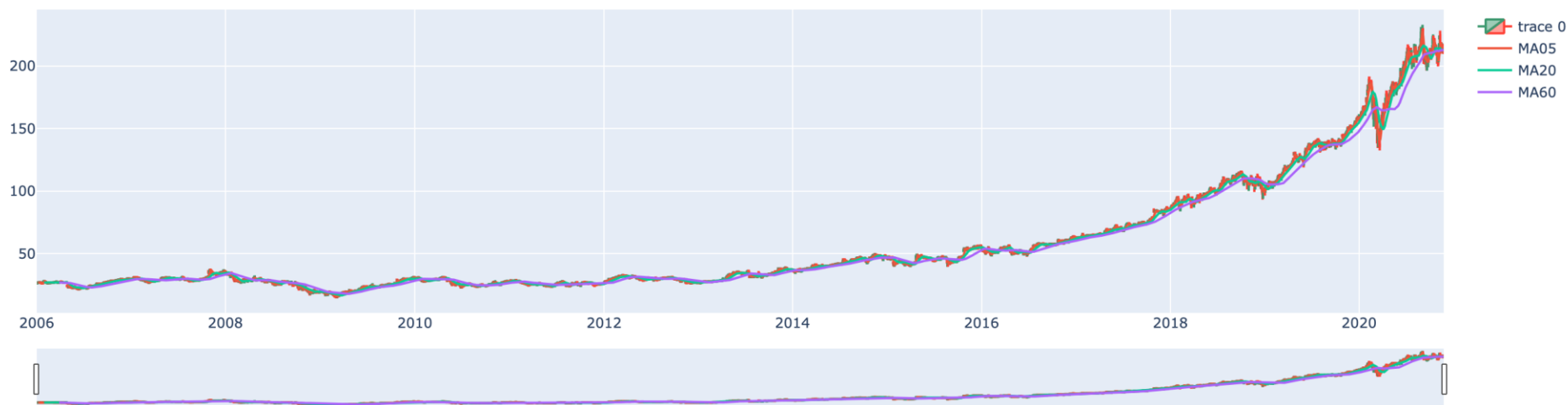
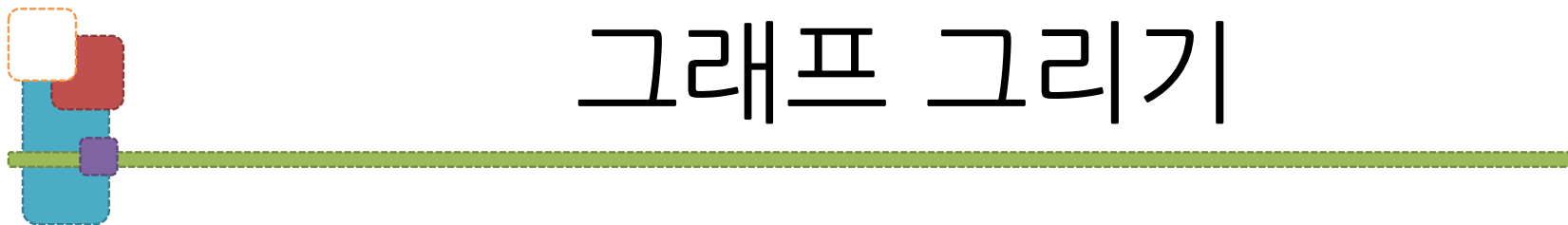
```
graph1=go.Candlestick(x=df['Date'],
    open=df['Open'],high=df['High'],low=df['Low'],close=df['Close'])
graph2=go.Scatter(x=df['Date'],y=df['MA05'],name='MA05')
graph3=go.Scatter(x=df['Date'],y=df['MA20'],name='MA20')
graph4=go.Scatter(x=df['Date'],y=df['MA60'],name='MA60')
```

```
fig.add_trace(graph1)
fig.add_trace(graph2)
fig.add_trace(graph3)
fig.add_trace(graph4)
```

```
fig.update_xaxes(rangebreaks=[dict(values=dt_breaks)])
```

```
fig.show()
```

# 그래프 그리기



# 범위 조정

X축 값의 위는 아래 슬라이드바에서 조정이 가능하나 Y축 값이 재 조정되지 않아 매우 좋다

**연습문제:** Dataframe에서 X축 값(즉, Date)의 범위를 2019-01-01에서 2019-12-31로 바꾸어 다시 그래프를 그려보시오.

