

Plotly.js 한국 주식 분봉 차트의 비거래 시간 공백 제거를 위한 종합 분석 보고서

I. 서론: 금융 차트에서의 비연속적 시간 문제

금융 시장은 정해진 시간에만 운영되지만, Plotly.js와 같은 표준 차트 라이브러리는 기본적으로 시간을 연속적인 선형 축으로 간주합니다. 이러한 불일치는 전문적인 금융 분석에 있어 시각적, 분석적 문제를 야기하며, 이를 해결하는 것은 필수적입니다.

1.1 기술적 분석에 대한 시각적 공백의 영향

금융 시계열 데이터를 시각화할 때 가장 먼저 직면하는 문제는 비거래 시간으로 인한 차트의 공백입니다. Plotly는 기본적으로 시간 축을 선형적으로 렌더링하기 때문에 주말, 공휴일, 야간 등 거래가 없는 기간이 거대한 빈 공간으로 표시됩니다.¹ 이 문제는 특히 30분봉, 1시간봉과 같은 분봉 데이터에서 더욱 두드러집니다. 야간과 주말의 공백이 실제 거래 데이터를 압도하여 차트의 가독성을 심각하게 저해하고 분석 도구로서의 가치를 상실하게 만듭니다.³

이러한 공백은 단순한 미관상의 문제를 넘어 기술적 분석을 방해하는 핵심적인 요인으로 작용합니다. 이동평균선과 같은 보조지표가 왜곡될 수 있으며, 헤드앤숄더, 플래그 등과 같은 주요 차트 패턴을 식별하기 어려워집니다. 분석가들이 기대하는 시각적 연속성이 깨지기 때문입니다.¹ 이 문제는 Plotly 커뮤니티에서 오랫동안 해결이 요구되어 온 핵심 기능으로, 이는 금융 애플리케이션에서의 중요성을 방증합니다.¹ 궁극적인 목표는 TradingView와 같은 전문 차트 플랫폼처럼 이러한 공백을 기본적으로 처리하여 데이터의 본질에 집중할 수 있는 환경을 구축하는 것입니다.¹

1.2 한국거래소(KRX)의 비거래 시간 정의

정확한 공백 제거를 위해서는 대상 시장의 거래 시간을 명확히 정의해야 합니다. 한국 주식 시장의 경우, 한국거래소(KRX)의 운영 기준은 다음과 같습니다.

- **정규 거래 시간:** KRX의 정규 거래 세션은 평일 오전 9시부터 오후 3시 30분까지(한국 표준시, KST, UTC+9)입니다.⁶
- **시간 외 세션:** KRX는 장전 시간 외(08:00-09:00) 및 장후 시간 외(15:40-18:00) 세션도 운영합니다.⁶ 시각화하려는 데이터셋이 이러한 시간 외 거래를 포함하는지에 따라 공백 제거 로직이 달라질 수 있어 복잡성을 더합니다.
- **주말 및 공휴일:** 토요일, 일요일 및 공식 공휴일에는 거래가 없습니다. 완전한 솔루션을 위해서는 2024년, 2025년 등 대상 연도의 공휴일 목록을 확보하는 것이 필수적입니다.⁹

이러한 거래 시간 정보를 명확히 하기 위해 아래와 같이 표로 정리할 수 있습니다.

표 1: 한국거래소(KRX) 거래 세션 (KST 기준)

세션 유형	시작 시간 (KST)	종료 시간 (KST)	비고
장전 시간 외	08:00	09:00	단일가 매매 방식
정규 세션	09:00	15:30	일반적인 접속 매매 방식
장후 시간 외	15:40	18:00	당일 종가 또는 단일가 매매

이러한 '공백 문제'는 단순히 차트를 예쁘게 만드는 것을 넘어, 시장 이벤트의 이산적(discrete) 성격과 데이터 시각화의 연속적(continuous) 시간 표현 사이의 근본적인 충돌을 해결하는 과정입니다. 이 문제를 해결하는 것은 범용 차트 라이브러리를 전문 금융 분석 도구로 전환하기 위한 전제 조건입니다. 금융 시장은 규칙 기반의 세션 내에서 이벤트 중심으로 움직이며, 이 세션 밖의 시간은 대부분의 가격 움직임 분석에서 의미가 없습니다. 차트에 이러한 무의미한 시간을 표시하는 것은 분석의 신호 대 잡음비를 낮추고 혼란을 야기합니다. 따라서 공백을 제거하는 행위는 X축의 맥락을 '선형적인 시간'에서 '시장 이벤트 시간'으로 재정의하는 중요한 개념적 전환이며, 이는 트레이더와 분석가에게 훨씬 더 의미 있는 분석 환경을 제공합니다.

II. 기존 해결책과 그 명백한 한계

Plotly에 공식적인 공백 제거 기능이 도입되기 전, 개발자들은 임시방편적인 해결책을 사용해야 했습니다. 이 섹션에서는 가장 널리 알려진 방법인 X축 타임을 변경하는 기법을 소개하고, 왜 이

방법이 상호작용이 중요한 금융 분석 환경에서 부적합한지를 체계적으로 분석합니다.

2.1 `axis.type = 'category'` 방식: 피상적으로 간단한 해결책

가장 초기의 해결책은 X축의 타입(type)을 기본값인 'date'에서 'category'로 변경하는 것이었습니다.⁴ 이 방법은 Plotly가 각 타임스탬프를 연속적인 시간 위의 한 점이 아닌, 고유하고 등간격의 문자열 레이블로 취급하도록 강제합니다.¹³ 그 결과, 주말이나 공휴일로 인한 시각적 공백은 사라집니다. 차트는 더 이상 데이터 포인트 간의 시간적 거리를 인지하지 않고, 단순히 리스트의 순차적인 항목처럼 하나의 캔들 다음에 다른 캔들을 나열하기 때문입니다.¹³

2.2 단점 분석: 시계열 상호작용 기능의 붕괴

이 방법은 표면적으로 문제를 해결하는 것처럼 보이지만, 시계열 데이터가 가진 본질적인 속성을 파괴하여 심각한 부작용을 초래합니다.

- **시간적 정보의 상실:** 가장 큰 문제는 축이 시간에 대한 모든 감각을 잃어버린다는 점입니다. 축은 단순히 문자열의 나열이 되며¹⁴, 이로 인해 60시간의 주말 공백과 30분의 장중 데이터 간격이 동일하게 취급됩니다. 즉, 하나의 카테고리에서 다음 카테고리로 넘어가는 한 단계로 축소됩니다.¹⁵
- **Rangeslider 및 Zoom/Pan 기능의 실패:** 이는 금융 차트에서 가장 치명적인 단점입니다. 시계열 데이터를 탐색하는 핵심 도구인 `rangeslider`는 카테고리 축과 근본적으로 호환되지 않습니다. 상호작용적인 확대/축소(`zoom`) 및 이동(`pan`) 기능 역시 비정상적으로 작동하거나 완전히 깨집니다.¹⁶ 커뮤니티에서는 확대/축소 시 Y축 자동 스케일링이 실패하여 차트를 사용할 수 없게 되는 문제가 지속적으로 보고되었습니다.¹⁹ 데이터 포인트가 화면 밖으로 밀려나거나 차트가 납작하게 변형되는 현상도 발생합니다.²²
- **부적절한 레이블링 및 포매팅:** Plotly의 정교한 날짜 포매팅 로직(`tickformat`, `dtick` 등)이 카테고리 축에는 적용되지 않기 때문에, 축 레이블이 복잡하게 겹치거나 부적절한 형식으로 표시되는 경우가 많습니다.¹³

2.3 시간적 척도의 왜곡과 분석적 함의

모든 시간 간격을 동일하게 처리함으로써, 이 차트는 시장이 지속적으로 활동하는 듯한 잘못된 인상을 줄 수 있습니다. 이는 야간의 가격 변동이나 주말 동안 발생한 이벤트의 중요성을

회석시킬 수 있습니다. 또한, 24시간 수익률 계산과 같이 일정한 시간 간격에 의존하는 모든 기술적 지표는 왜곡된 타임라인으로 인해 그 의미를 잃게 됩니다. 이러한 이유로 개발자 커뮤니티에서는 이 방법을 심각한 애플리케이션에 사용하기에는 부적합한, 어쩔 수 없는 타협 또는 "패배"로 간주합니다.³

category 축 방식의 실패는 데이터 시각화의 중요한 설계 원칙을 드러냅니다. 즉, 축의 기본 데이터 모델은 플로팅되는 데이터의 개념적 모델과 일치해야 합니다. 시계열 데이터는 순서, 방향성, 그리고 '기간'이라는 고유한 속성을 가집니다. **'date'** 축 모델은 이 세 가지를 모두 존중하지만, **'category'** 축 모델은 순서만 존중하고 기간의 개념은 무시합니다. 사용자가 시계열 데이터에 **category** 축을 적용하는 것은 '기간' 정보를 의도적으로 버리는 손실 압축과 같습니다. **rangeslider**나 **zoom**과 같은 상호작용 기능은 바로 이 '기간' 속성을 기반으로 작동하도록 설계되었습니다. 따라서 기간 정보가 제거된 데이터 모델에 연속적인 시간 기반의 연산을 시도하면 기능이 치명적으로 실패하는 것은 당연한 결과입니다. 이는 단순한 버그가 아니라, 데이터의 본질과 시각화 도구의 모델 간의 근본적인 불일치에서 비롯된 문제입니다.

III. 현대적 해결책: **layout.xaxis.rangebreaks** 심층 분석

Plotly는 비거래 시간 공백 문제를 해결하기 위해 **rangebreaks**라는 공식적이고 강력한 기능을 도입했습니다. 이 섹션에서는 **rangebreaks** 속성을 체계적으로 분해하고, 각 구성 요소의 기능과 사용법을 명확한 예시와 함께 설명하여 이것이 가장 올바르고 견고한 접근법임을 제시합니다.

3.1 **rangebreaks** 속성의 구조

layout.xaxis.rangebreaks 기능은 **'date'** 타입 축의 모든 상호작용 기능을 그대로 유지하면서 비거래 시간 공백 문제를 해결하기 위해 특별히 설계되었습니다.¹ 이 기능의 핵심 원리는 사용자가 축에서 숨기거나 "끊어낼" 특정 시간 구간을 직접 정의할 수 있도록 하는 것입니다.

rangebreaks 속성은 각각의 '끊김' 유형을 정의하는 객체들의 배열로 구성되며, 주요 속성으로는 **pattern**, **bounds**, **values**, **dvalue**가 있습니다.²³

표 2: **layout.xaxis.rangebreaks** 속성 상세

속성	타입	설명	사용 예시
enabled	Boolean	해당 rangebreak 규칙의 활성화	<code>dict(bounds=['sat', 'mon'],</code>

		여부를 결정합니다. 기본값은 true입니다.	enabled=True)
pattern	String	'day of week' 또는 'hour'와 같이 반복되는 시간 패턴을 지정합니다.	dict(pattern='day of week',...)
bounds	Array	pattern과 함께 사용되어 끊어낼 시간의 시작과 끝 경계를 정의합니다.	dict(bounds=['sat', 'mon']) 또는 dict(bounds=[15.5, 9])
values	Array	불규칙적인 특정 날짜나 시간 목록을 직접 지정하여 끊어냅니다. 공휴일 처리에 유용합니다.	dict(values=['2024- 05-01', '2024-05-06'])
dvalue	Number	values에 지정된 각 항목이 차지하는 시간의 길이(밀리초 단위)를 설정합니다.	dict(values=[...], dvalue=24*60*60*1 000) (하루)

3.2 정적 공백 제거: 주말 및 표준 비거래 시간 제외

가장 일반적인 사용 사례는 주말과 정규 거래 시간 외의 야간 시간을 제거하는 것입니다.

- **주말 제외:** 토요일과 일요일을 제거하는 것은 패턴 기반의 `rangebreak`를 통해 간단히 구현할 수 있습니다. `dict(pattern='day of week', bounds=)` 또는 더 직관적인 `dict(bounds=['sat', 'mon'])` 코드를 사용합니다.²³ 이는 Plotly에게 토요일(요일 인덱스 6)부터 월요일(요일 인덱스 1) 시작 전까지의 시간을 축에서 숨기도록 지시합니다.
- **비거래 시간 제외:** 분봉 차트에서는 야간 공백을 제거해야 합니다. KRX 정규 거래 시간(09:00-15:30)에 맞춰 `dict(pattern='hour', bounds=[15.5, 9])`와 같이 설정할 수 있습니다.²³ 이는 오후 3시 30분부터 다음 날 오전 9시까지의 시간을 숨깁니다.
- **정적 규칙 결합:** `rangebreaks` 배열에 여러 개의 `dict` 객체를 포함하여 주말 제외 규칙과

비거래 시간 제외 규칙을 동시에 적용할 수 있습니다.²⁴

`rangebreaks` 기능의 설계는 문제의 본질에 대한 깊은 이해를 보여줍니다. 이 기능은 `pattern/bounds`를 이용한 규칙 기반 방식과 `values/dvalue`를 이용한 데이터 기반 방식을 모두 제공합니다. 이는 비거래 기간이 두 가지 유형으로 나뉜다는 점을 반영한 것입니다. 첫째는 주말이나 야간처럼 규칙적으로 반복되는 예측 가능한 공백이고, 둘째는 공휴일이나 임시 휴장일처럼 불규칙적인 예측 불가능한 공백입니다. `Plotly` 개발자들은 이 차이를 인식하고, 예측 가능한 공백은 `pattern`과 `bounds`를 통해 간단하고 효율적으로 처리할 수 있도록 했습니다. 반면, 예측 불가능한 공백은 규칙 기반 시스템으로는 처리할 수 없으므로 `values`와 `dvalue`를 통해 명시적으로 지정할 수 있는 유연성을 제공했습니다. 이러한 이중 모드 설계는 일반적인 사례에 대한 사용 편의성과 복잡한 실제 시나리오에 대한 유연성을 모두 만족시키는 정교한 아키텍처이며, 단일 모드 솔루션보다 훨씬 더 강력한 기능을 제공합니다.

IV. 고빈도 데이터를 위한 고급 구현: 동적 Range Break 계산

이 섹션에서는 한국 주식 '분봉' 데이터를 시각화하는 사용자의 특정 요구에 대한 궁극적인 해결책을 제시합니다. 정적인 규칙을 넘어, 공휴일과 비정기적인 시장 휴장을 포함한 모든 비거래 기간을 처리할 수 있는 완전한 동적, 데이터 주도 알고리즘을 소개합니다.

4.1 알고리즘적 접근: 연속적 타임라인과 실제 데이터 비교를 통한 공백 식별

정적인 `pattern` 규칙은 공휴일이나 기타 비정기적인 비거래일을 처리할 수 없습니다. 모든 공휴일을 수동으로 나열하는 것은 번거롭고 오류가 발생하기 쉽습니다. 분봉 데이터의 경우, 데이터가 누락되면 장중에도 공백이 발생할 수 있어 시간 단위 패턴만으로는 불충분합니다.

가장 견고한 방법은 누락된 타임스탬프 목록을 프로그래밍 방식으로 동적으로 생성하는 것입니다.²⁶ 알고리즘의 논리는 다음과 같습니다.

1. 전체 데이터셋의 시작과 종료 타임스탬프를 확인합니다.
2. 시작과 종료 지점 사이에 원하는 빈도(예: 30분 간격)로 완벽하게 연속적인 시간 범위를 생성합니다. 이는 모든 가능한 캔들 슬롯을 나타냅니다.
3. 실제 거래 데이터에 존재하는 타임스탬프 집합을 만듭니다.
4. '모든 가능한 캔들 슬롯' 집합과 '실제 데이터' 집합의 차집합을 구하면, 이것이 바로 모든 비거래 타임스탬프의 정확한 목록이 됩니다.

이 방법은 `rangebreaks`의 `values` 속성을 사용하여 계산된 비거래 시간 목록을 전달하고, `dvalue` 속성을 통해 각 간격의 길이를 밀리초 단위로 지정합니다(예: 30분봉의 경우 $\$30 * 60 * 1000\$$).²⁶

4.2 Python과 Pandas를 이용한 단계별 구현

아래는 위에서 설명한 알고리즘을 구현한 상세한 Python 코드 예제입니다.

- 1단계: 데이터 준비
분봉 또는 시간봉 주식 데이터를 Pandas DataFrame으로 로드하고, 타임스탬프 컬럼이 `datetime` 형식인지 확인합니다.
- 2단계: 연속적인 타임라인 생성
`pd.date_range()` 함수를 사용하여 데이터의 최소/최대 타임스탬프와 적절한 빈도(`freq='30min'`, `freq='H'`)를 지정하여 완전한 시간 인덱스를 생성합니다.²⁶
- 3단계: 실제 거래 타임스탬프 식별
원본 데이터에서 실제 타임스탬프 목록을 추출합니다.
- 4단계: 공백(Breaks) 계산
Pandas 인덱싱이나 집합 연산을 사용하여 연속적인 타임라인에는 있지만 실제 거래 타임스탬프에는 없는 시간들을 찾아냅니다. 이 목록이 `dt_breaks`가 됩니다.
- 5단계: Plotly Figure 설정
`go.Candlestick` 차트를 생성하고, `fig.update_xaxes()` 내에서 `rangebreaks=[dict(values=dt_breaks, dvalue=...)]`와 같이 계산된 목록을 전달합니다.²⁶

Python

```
import pandas as pd
import plotly.graph_objects as go

# 1단계: 데이터 준비 (예시 데이터 생성)
# 실제로는 파일에서 데이터를 로드합니다.
data = {
    'timestamp': pd.to_datetime(['2024-05-20 09:00', '2024-05-20 09:30', '2024-05-20 15:00',
                                  '2024-05-21 09:00', '2024-05-21 09:30']),
    'open': ,
    'high': ,
    'low': ,
    'close':
```

```

}
df = pd.DataFrame(data)
df.set_index('timestamp', inplace=True)

# 2단계: 연속적인 타임라인 생성
freq_minutes = 30
full_time_range = pd.date_range(start=df.index.min(), end=df.index.max(),
freq=f'{freq_minutes}T')

# 3단계 & 4단계: 공백 계산
# full_time_range에 없는 실제 데이터의 인덱스를 찾습니다.
dt_breaks = full_time_range.difference(df.index)

# 5단계: Plotly Figure 설정
fig = go.Figure(data=[go.Candlestick(
    x=df.index,
    open=df['open'],
    high=df['high'],
    low=df['low'],
    close=df['close']
)])

# dvalue를 밀리초 단위로 설정합니다.
dvalue_ms = freq_minutes * 60 * 1000

fig.update_xaxes(
    rangebreaks=[
        dict(values=dt_breaks, dvalue=dvalue_ms) # 동적으로 계산된 공백
    ]
)

fig.update_layout(title_text='KRX 30분봉 차트 (비거래 시간 제거)', xaxis_title='날짜',
yaxis_title='가격')
fig.show()

```

4.3 공휴일 캘린더 통합

동적 알고리즘은 데이터에 공백이 있는 모든 기간(공휴일 포함)을 자동으로 포착합니다. 대안적인 접근법은 KRX 휴장일 목록을 명시적으로 정의하여 `rangebreaks`에 추가하는

것입니다.⁹ 이 방법은 주말을 제거하는

pattern 기반 규칙과 결합할 수 있습니다. 일봉 데이터에서는 더 간단할 수 있지만, 장중 데이터 누락까지 처리해야 하는 분봉 데이터에는 동적 계산 방식이 더 포괄적이고 정확합니다.

4.4 대용량 데이터셋에 대한 성능 고려사항

한 가지 중요한 경고는 매우 긴 기간에 걸쳐 고빈도(예: 초 또는 밀리초 단위)의 연속적인 타임라인을 생성할 경우 메모리 부족(**memory overflow**)이 발생할 수 있다는 점입니다.²⁶ 이를 완화하기 위해, 거대한 단일

date_range를 생성하는 대신, 하루 단위로 반복하며 공백을 계산하는 방식을 사용할 수 있습니다. 각 날짜가 주말이나 공휴일인지 먼저 확인하고, 거래일인 경우에만 야간 시간과 같은 장중 비거래 시간을 계산하여 **breaks** 목록에 추가하는 것입니다. 이 방식은 메모리 사용량을 크게 줄일 수 있습니다.

동적 계산 방법은 선언적 설정(Plotly에게 공백의 '규칙'을 알려주는 것)에서 명령적 생성(Plotly에게 공백의 정확한 '데이터'를 제공하는 것)으로의 패러다임 전환을 의미합니다. 이 접근법은 계산 비용이 더 높지만, 실제 거래 데이터에 거의 완벽한 충실도를 제공하여 고빈도 데이터 시각화를 위한 유일한 전문가급 솔루션입니다. 이는 시각화 로직이 데이터 자체와 긴밀하게 결합되어야 최고의 정확성을 얻을 수 있음을 시사하며, "기능 사용법"을 넘어 "견고한 시각화를 위한 데이터 처리 파이프라인 설계"라는 공학적 과제로 논의를 격상시킵니다.

V. 종합 및 전략적 권장사항

지금까지 논의된 모든 방법론을 종합하여 명확한 비교 분석을 제공하고, 사용자의 요구에 맞는 실행 가능한 권장사항을 제시합니다.

5.1 방법론 비교 분석

각 방법론의 장단점을 한눈에 파악할 수 있도록 아래의 비교 매트릭스를 제시합니다. 이를 통해 사용자는 자신의 특정 요구사항과 제약 조건에 가장 적합한 방법을 선택할 수 있습니다.

표 3: 방법론 비교 매트릭스

기준	xaxis.type='category'	정적 rangebreaks (pattern/bounds)	동적 rangebreaks (values/dvalue)
구현 복잡도	매우 낮음	낮음	중간
rangeslider/Zoom 지원	불가	완전 지원	완전 지원
주말 처리	가능	가능	가능
비거래 시간 처리	불가	가능	가능
공휴일 처리	불가	불가 (수동 추가 필요)	자동 처리
분봉 데이터 적합성	부적합	제한적	매우 높음
성능 영향	낮음	매우 낮음	중간 (데이터 크기에 따라 다름)
권장사항	사용 금지	일봉 또는 주봉 차트에 적합	분봉/시간봉 등 모든 시계열에 강력 권장

5.2 한국 주식 시장 분봉 데이터 시각화를 위한 모범 사례

- **최우선 권장사항:** 사용자가 명시한 '한국 주식 분봉' 차트 시각화 목표를 위해서는 동적 **Range Break** 계산(IV 섹션) 방식이 명백히 가장 우수한 접근법입니다. 이는 정확성을 보장하고 모든 상호작용 기능을 완벽하게 보존하는 유일한 방법입니다.
- **데이터 전처리:** 최종 시각화의 품질은 데이터 전처리 파이프라인의 품질에 달려있음을 강조해야 합니다. 공백을 계산하는 로직은 재사용 가능한 함수로 캡슐화하는 것이 좋습니다.
- **시간대 인식:** 모든 계산은 시간대를 인식하는 **datetime** 객체(특히 KRX 데이터의 경우 KST)를 사용하여 수행해야 합니다. 이는 오차나 부정확한 공백 계산을 방지하는 데 필수적입니다.

5.3 재사용 가능한 차트 함수를 위한 최종 코드 아키텍처

모범 사례를 종합하여, `create_krx_candlestick(df, freq='30min', holidays=[...])`와 같은 형태의 Python 함수 시그니처를 구상할 수 있습니다. 이 함수는 동적 공백 계산 로직, 캔들스틱 트레이스 생성, `rangebreaks`를 포함한 레이아웃 설정을 하나의 모듈화되고 재사용 가능한 코드로 통합하여 제공할 수 있습니다.

VI. 결론: Plotly.js를 활용한 전문가 수준의 금융 시각화 달성

본 보고서는 Plotly.js를 사용하여 한국 주식 분봉 차트의 비거래 시간 공백을 제거하는 다양한 방법론을 심층적으로 분석했습니다. 결론적으로, Plotly.js는 강력한 범용 라이브러리이지만, 전문가 수준의 금융 차트를 구현하기 위해서는 고급 기능에 대한 깊은 이해와 데이터 중심의 사례 깊은 접근 방식이 필요합니다.

분석 결과, 결함이 많은 임시방편(category 축)에서 시작하여 강력하고 유연한 `rangebreaks` 기능으로 기술이 발전해왔음을 확인했습니다. 특히, 실제 데이터를 기반으로 비거래 기간을 동적으로 계산하는 방법론은 시각적으로 깔끔하고 분석적으로 건전하며, 완전한 상호작용을 지원하는 차트를 생성할 수 있게 해줍니다. 이는 전문 금융 차트 플랫폼의 표준을 충족하는 결과물입니다.

Plotly.js와 같은 라이브러리가 계속 발전함에 따라, 향후에는 시장별 캘린더 통합이나 자동화된 공백 처리 기능이 표준으로 자리 잡을 수도 있을 것입니다. 그러나 현재로서는 본 보고서에서 설명한 동적 접근법이 최첨단 기술을 대표하며, 개발자가 Plotly.js의 잠재력을 최대한 활용하여 정교하고 실용적인 금융 시각화 솔루션을 구축할 수 있는 가장 확실한 경로를 제공합니다.

참고 자료

1. Should remove breaks for OHLC, Candlestick · Issue #1382 · plotly/plotly.js - GitHub, 9월 13, 2025에 액세스, <https://github.com/plotly/plotly.js/issues/1382>
2. Discontinue time in candle stick chart - Plotly Community Forum, 9월 13, 2025에 액세스, <https://community.plotly.com/t/discontinue-time-in-candle-stick-chart/28895>
3. How to remove these empty spaces in candle stick? - Plotly Community Forum, 9월 13, 2025에 액세스, <https://community.plotly.com/t/how-to-remove-these-empty-spaces-in-candle-s>

[tick/55566](#)

4. Candlesticks - how to avoid weekend gaps in financial data - Plotly ..., 9월 13, 2025에 액세스,
<https://community.plotly.com/t/candlesticks-how-to-avoid-weekend-gaps-in-financial-data/3749>
5. How to graph time-series data with a discontinuous x-axis? - Plotly Community Forum, 9월 13, 2025에 액세스,
<https://community.plotly.com/t/how-to-graph-time-series-data-with-a-discontinuous-x-axis/423>
6. Korea Exchange - Wikipedia, 9월 13, 2025에 액세스,
https://en.wikipedia.org/wiki/Korea_Exchange
7. South Korea Stock Exchange - Twelve Data, 9월 13, 2025에 액세스,
<https://twelvedata.com/exchanges/xkrx>
8. KRX eyes 12-hour trading to counter Nextrade's rapid rise - The Korea Herald, 9월 13, 2025에 액세스, <https://www.koreaherald.com/article/10548345>
9. Korea Exchange (KRX) | Key trading information, 9월 13, 2025에 액세스,
<https://globalexchangesdirectory.com/exchange/korea-exchange-krx-xkrx>
10. 2025 Market Holidays: South Korea - RYT9, 9월 13, 2025에 액세스,
<https://www.ryt9.com/en/hol/278011>
11. Korea Stock Exchange KRX Holidays 2025 | Trading Hours. - ForexChurch.com, 9월 13, 2025에 액세스,
<https://www.forexchurch.com/stock-market-holidays/korea-stock-exchange>
12. Stock Market Holiday Calendar - Investing.com, 9월 13, 2025에 액세스,
<https://www.investing.com/holiday-calendar/>
13. Plotly: How to style a plotly figure so that it doesn't display gaps for missing dates?, 9월 13, 2025에 액세스,
<https://stackoverflow.com/questions/61346100/plotly-how-to-style-a-plotly-figure-so-that-it-doesnt-display-gaps-for-missing>
14. X axis formatting in category type - Plotly Community Forum, 9월 13, 2025에 액세스, <https://community.plotly.com/t/x-axis-formatting-in-category-type/33973>
15. Highlight areas of interactive time series plotly plot where y is greater than defined threshold and annotate them - Stack Overflow, 9월 13, 2025에 액세스,
<https://stackoverflow.com/questions/78761758/highlight-areas-of-interactive-time-series-plotly-plot-where-y-is-greater-than-d>
16. Plotly - Autorescaling y axis range when range slider used - Stack Overflow, 9월 13, 2025에 액세스,
<https://stackoverflow.com/questions/75876460/plotly-autorescaling-y-axis-range-when-range-slider-used>
17. Interaction with rangeslider causes my plot data points to shift up, 9월 13, 2025에 액세스,
<https://community.plotly.com/t/interaction-with-rangeslider-causes-my-plot-data-points-to-shift-up/91235>
18. Rangeslider on Candlestick/OHLC charts - Plotly Community Forum, 9월 13, 2025에 액세스,
<https://community.plotly.com/t/rangeslider-on-candlestick-ohlc-charts/3912>

19. Update y-axis range when x-axis rangeslider is used · Issue #912 · plotly/plotly.R - GitHub, 9월 13, 2025에 액세스, <https://github.com/ropensci/plotly/issues/912>
20. candlestick chart - Plotly vertical range is not automatically adjusted - Stack Overflow, 9월 13, 2025에 액세스, <https://stackoverflow.com/questions/70916985/plotly-vertical-range-is-not-automatically-adjusted>
21. Python: Plot candlesticks with automatic Y zoom - Stack Overflow, 9월 13, 2025에 액세스, <https://stackoverflow.com/questions/51162010/python-plot-candlesticks-with-automatic-y-zoom>
22. Autoscale y-axis on zoom for candlestick chart - Plotly Community Forum, 9월 13, 2025에 액세스, <https://community.plotly.com/t/autoscale-y-axis-on-zoom-for-candlestick-chart/53148>
23. Layout.xaxis in JavaScript - Plotly, 9월 13, 2025에 액세스, <https://plotly.com/javascript/reference/layout/axis/>
24. How remove weekends from plot using Plotly? - Stack Overflow, 9월 13, 2025에 액세스, <https://stackoverflow.com/questions/62824368/how-remove-weekends-from-plot-using-plotly>
25. Candle Stick Chart - Removing Time & Date breaks candle chart - Plotly Community Forum, 9월 13, 2025에 액세스, <https://community.plotly.com/t/candle-stick-chart-removing-time-date-breaks-candle-chart/88988>
26. python - Ignore Non Trading days (Holidays / remove gaps) in Plotly ..., 9월 13, 2025에 액세스, <https://stackoverflow.com/questions/70716384/ignore-non-trading-days-holidays-remove-gaps-in-plotly-candlesticks-for-min>