

주식 시세 현황 분석



소프트웨어융합대학원
진혜진



1. 프로젝트 개요

2. 웹 기본 지식 이해하기

3. 웹 스크래핑

4. 데이터 수집 및 가공

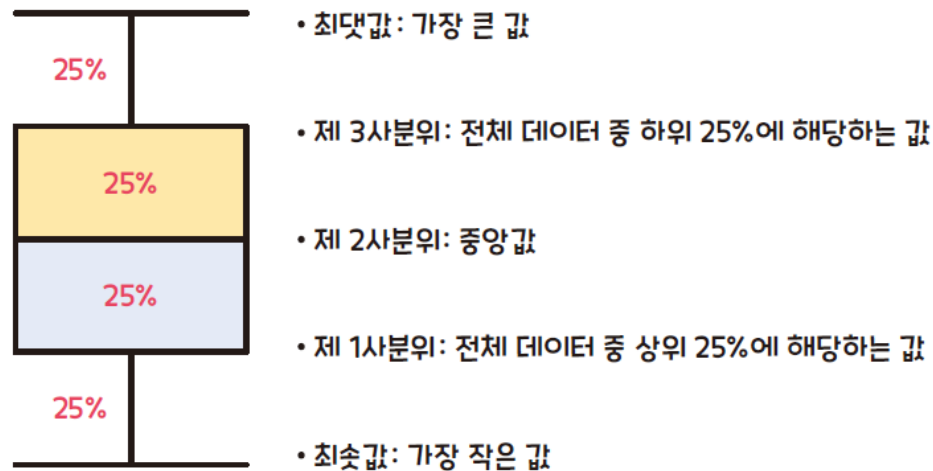
5. 데이터 시각화

데이터 시각화

1. 상자 수염 그래프

■ 상자 수염 그래프(Box-and-whisker plot)

- 주식 시세 그래프를 확인할 때 등장하는 그래프→ 영어명 : 박스 플랏(Box plot)
- 데이터를 요약한 다섯 가지로 지표를 표기함
- boxplot() 메서드를 사용
 - 일일 시세 데이터를 월별로 취합→ 상자 수염 그래프에 다섯 가지 요약 수치를 표현하는 방식으로 시각화 진행



상자 수염 그래프 표기 방법

2. 시각화 대상 데이터 추출하기

■ 한글 열 이름을 영문으로 변경하기

- 한글을 그래프로 표현 했을 때 문제점
 - 폰트에 따라 그래프 라벨이 깨져서 보이는 경우가 있음
 - 데이터 처리 시 문자 세트(charset) 불일치로 인해 문제가 생기기도 함
- rename() 메서드
 - 데이터를 안전하게 처리하고 시각화→ 필요한 열 이름만 영문으로 변경
- 날짜: date
- 종가: end_price
- 시가: start_price
- 고가: highest_price
- 저가: lowest_price

2. 시각화 대상 데이터 추출하기

■ 한글 열 이름을 영문으로 변경하기

- rename() 메서드 사용 : 다음 소스 코드 실행

```
# 열 이름 영문으로 변경
all_tables.rename(columns={'날짜': 'date',
                           '종가': 'end_price',
                           '시가': 'start_price',
                           '고가': 'highest_price',
                           '저가': 'lowest_price'},
                  inplace=True)

# 데이터 확인하기
all_tables
```

	date	end_price	전일비	start_price	highest_price	lowest_price	거래량
1	2025.04.17	55100.0	상승 400	54700.0	55300.0	54500.0	10862598.0
2	2025.04.16	54700.0	하락 1,900	56000.0	56200.0	54500.0	14437025.0
3	2025.04.15	56600.0	상승 400	56300.0	57100.0	56200.0	8998640.0
4	2025.04.14	56200.0	상승 1,000	56300.0	56700.0	55800.0	12852613.0
5	2025.04.11	55200.0	하락 1,200	55600.0	55700.0	54800.0	13930480.0
...
9	2024.11.25	57900.0	상승 1,900	57400.0	57900.0	56700.0	36237325.0
10	2024.11.22	56000.0	하락 400	56000.0	56700.0	55900.0	15281543.0
11	2024.11.21	56400.0	상승 1,100	54900.0	56900.0	54700.0	19096850.0
12	2024.11.20	55300.0	하락 1,000	56100.0	56500.0	54800.0	20864667.0
13	2024.11.19	56300.0	하락 400	56500.0	57500.0	55900.0	31539632.0

2. 시각화 대상 데이터 추출하기

■ 필요 없는 열 제거하기

- drop() 메서드

→ 전일비, 거래량 열 제거

```
# 필요 없는 열 제거
all_tables.drop(['전일비', '거래량'], axis=1, inplace=True)

# 데이터 확인하기
all_tables
```



	date	end_price	start_price	highest_price	lowest_price
1	2025.04.17	55100.0	54700.0	55300.0	54500.0
2	2025.04.16	54700.0	56000.0	56200.0	54500.0
3	2025.04.15	56600.0	56300.0	57100.0	56200.0
4	2025.04.14	56200.0	56300.0	56700.0	55800.0
5	2025.04.11	55200.0	55600.0	55700.0	54800.0
...
9	2024.11.25	57900.0	57400.0	57900.0	56700.0
10	2024.11.22	56000.0	56000.0	56700.0	55900.0
11	2024.11.21	56400.0	54900.0	56900.0	54700.0
12	2024.11.20	55300.0	56100.0	56500.0	54800.0
13	2024.11.19	56300.0	56500.0	57500.0	55900.0

100 rows × 5 columns

2. 시각화 대상 데이터 추출하기

■ 중앙값 추가하기

- 상자 수염 그래프에 표기할 데이터 : 최고가와 최저가의 중앙값

```
# 중앙값 열 추가
all_tables['median_price'] = all_tables['highest_price'] - ((all_tables['highest_price'] - all_tables['lowest_price']) / 2)

# 값 확인하기
all_tables
```

	date	end_price	start_price	highest_price	lowest_price	median_price
1	2025.04.17	55100.0	54700.0	55300.0	54500.0	54900.0
2	2025.04.16	54700.0	56000.0	56200.0	54500.0	55350.0
3	2025.04.15	56600.0	56300.0	57100.0	56200.0	56650.0
4	2025.04.14	56200.0	56300.0	56700.0	55800.0	56250.0
5	2025.04.11	55200.0	55600.0	55700.0	54800.0	55250.0
...
9	2024.11.25	57900.0	57400.0	57900.0	56700.0	57300.0
10	2024.11.22	56000.0	56000.0	56700.0	55900.0	56300.0
11	2024.11.21	56400.0	54900.0	56900.0	54700.0	55800.0
12	2024.11.20	55300.0	56100.0	56500.0	54800.0	55650.0
13	2024.11.19	56300.0	56500.0	57500.0	55900.0	56700.0

100 rows × 6 columns

2. 시각화 대상 데이터 추출하기

■ 집계를 위한 달 정보 추가하기

- date 열에 있는 일자 정보를 문자열로 변경
- 파이썬 슬라이싱으로 월까지 내용을 잘라서 추가 : 다음 소스 코드를 실행

```
# 월 정보 열 추가
all_tables['month'] = all_tables['date'].str[0:7]

# 값 확인하기
all_tables
```



	date	end_price	start_price	highest_price	lowest_price	midian_price	month
1	2025.04.17	55100.0	54700.0	55300.0	54500.0	54900.0	2025.04
2	2025.04.16	54700.0	56000.0	56200.0	54500.0	55350.0	2025.04
3	2025.04.15	56600.0	56300.0	57100.0	56200.0	56650.0	2025.04
4	2025.04.14	56200.0	56300.0	56700.0	55800.0	56250.0	2025.04
5	2025.04.11	55200.0	55600.0	55700.0	54800.0	55250.0	2025.04
...
9	2024.11.25	57900.0	57400.0	57900.0	56700.0	57300.0	2024.11
10	2024.11.22	56000.0	56000.0	56700.0	55900.0	56300.0	2024.11
11	2024.11.21	56400.0	54900.0	56900.0	54700.0	55800.0	2024.11
12	2024.11.20	55300.0	56100.0	56500.0	54800.0	55650.0	2024.11
13	2024.11.19	56300.0	56500.0	57500.0	55900.0	56700.0	2024.11

100 rows × 7 columns

2. 시각화 대상 데이터 추출하기

■ 색인 변경 및 정렬하기

- set_index() 메서드
 - 그래프 x축을 숫자로 설정 : 색인을 date로 변경
- sort_index() 메서드
 - 데이터를 색인 오름차순으로 정렬 : 왼쪽에서 오른쪽으로 날씨가 증가하는 그래프



```
# 날짜를 색인으로 지정하기
all_tables.set_index('date', inplace=True)

# 날짜 오름차순으로 정렬하기
all_tables.sort_index(inplace=True)

# 값 확인하기
all_tables
```

2. 시각화 대상 데이터 추출하기

■ 시각화를 위한 데이터 2차 가공이 완료됨

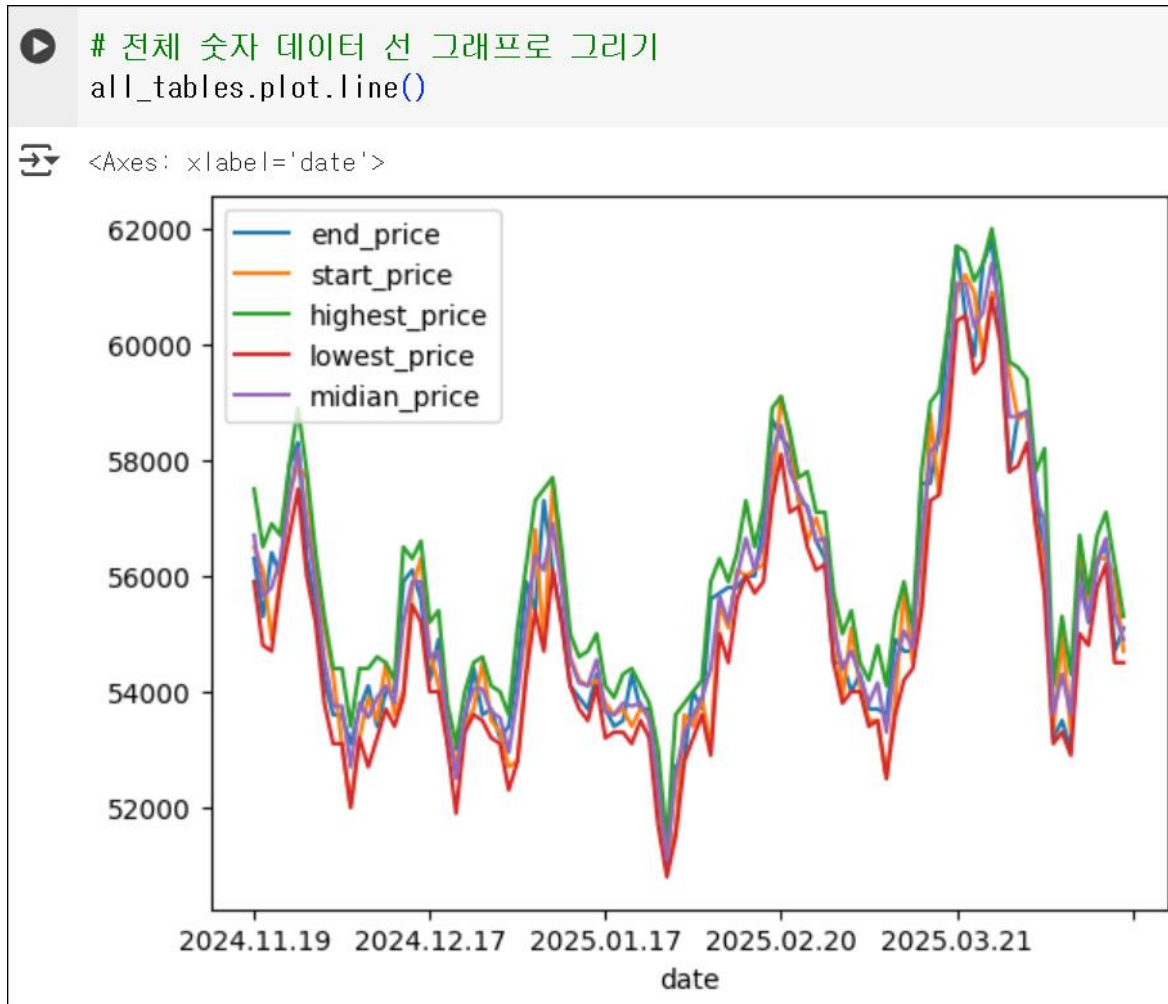
date	end_price	start_price	highest_price	lowest_price	midian_price	month
2024.11.19	56300.0	56500.0	57500.0	55900.0	56700.0	2024.11
2024.11.20	55300.0	56100.0	56500.0	54800.0	55650.0	2024.11
2024.11.21	56400.0	54900.0	56900.0	54700.0	55800.0	2024.11
2024.11.22	56000.0	56000.0	56700.0	55900.0	56300.0	2024.11
2024.11.25	57900.0	57400.0	57900.0	56700.0	57300.0	2024.11
...
2025.04.11	55200.0	55600.0	55700.0	54800.0	55250.0	2025.04
2025.04.14	56200.0	56300.0	56700.0	55800.0	56250.0	2025.04
2025.04.15	56600.0	56300.0	57100.0	56200.0	56650.0	2025.04
2025.04.16	54700.0	56000.0	56200.0	54500.0	55350.0	2025.04
2025.04.17	55100.0	54700.0	55300.0	54500.0	54900.0	2025.04

100 rows × 6 columns

3. 그래프 그리기

■ 기본 선 그래프

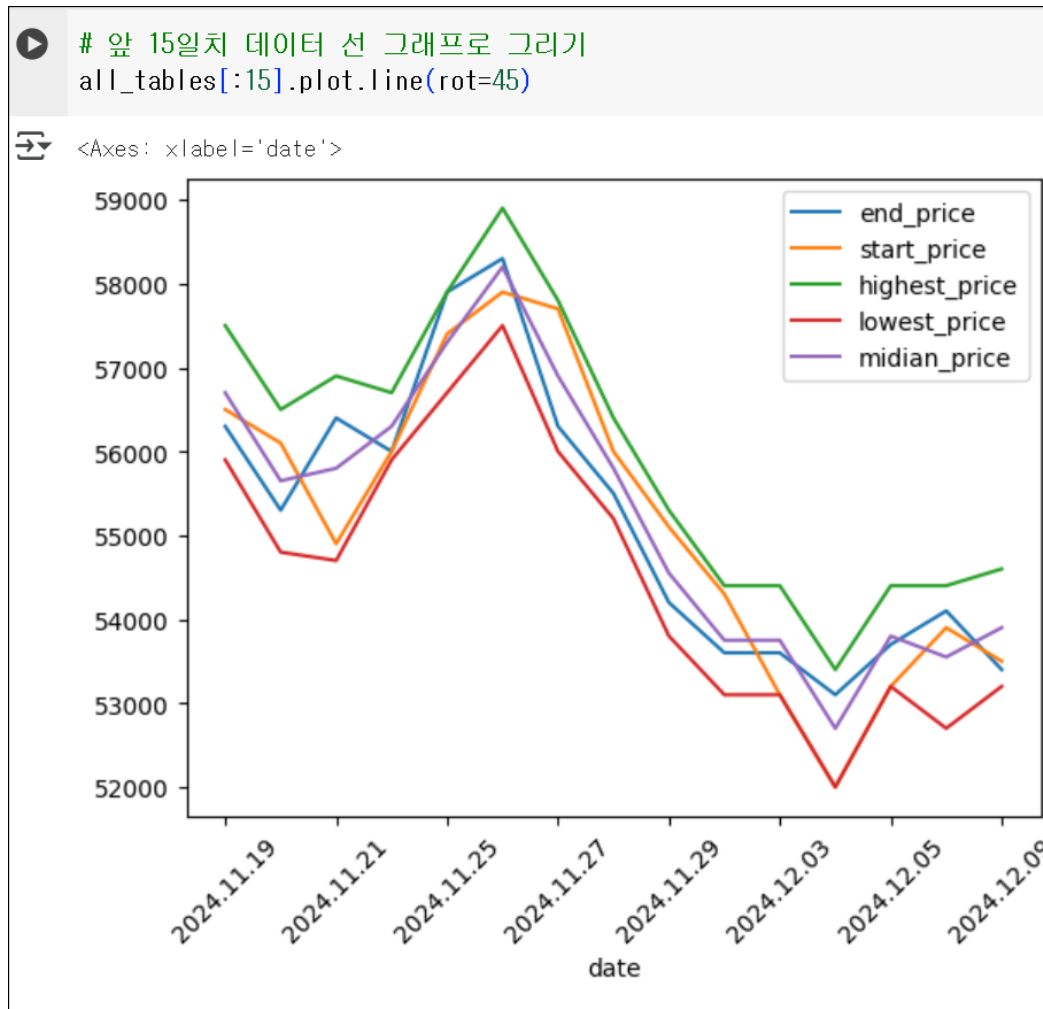
- 지금까지 준비한 데이터가 시각화 목적에 맞게 가공되었는지 확인



3. 그래프 그리기

■ 선 그래프 구역별로 나눠서 확인하기

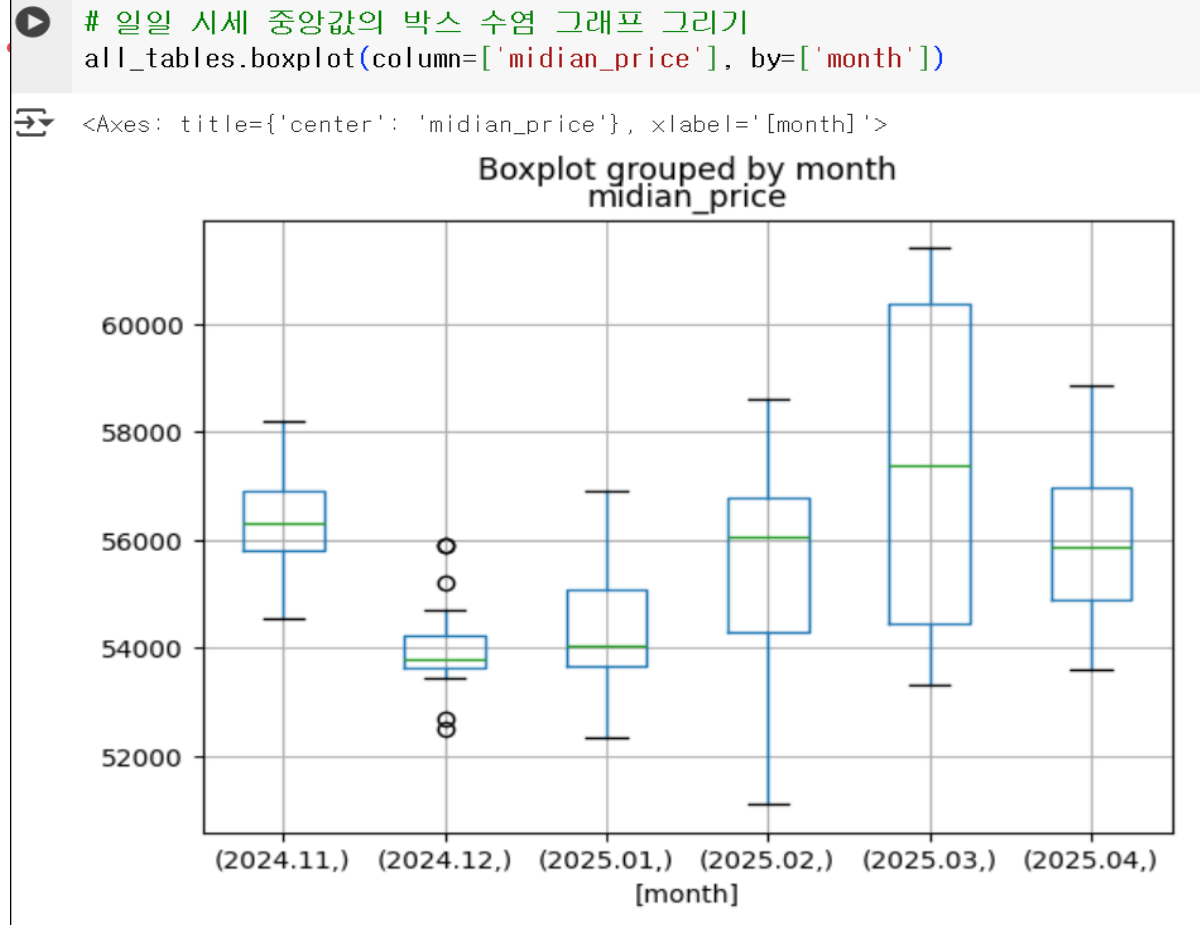
- 슬라이싱으로 앞 15개 데이터만 선택하고, x축 라벨은 45도 돌려서 표현



3. 그래프 그리기

■ 상자 수염 그래프

- 전체 데이터를 월 단위로 집계 → 중앙값의 월 단위 최댓값, 최솟값, 중앙값 확인



3. 그래프 그리기

- `boxplot`: 상자 수염 그래프를 그리기 위해 데이터프레임의 `boxplot()` 메서드 사용
 - 지금까지 사용한 `plot()` 메서드와는 다른 메서드임
- `'midian_price'`: 그래프를 그릴 열을 선택
 - 인수 `column`에 `midian_price` 열을 설정함
- `'month'`: 월 단위로 데이터를 집계
 - `by` 인수에 `month` 열의 값을 설정

4. 마무리

■ 문제 해결

- 영업일 기준 100일 분량의 데이터만 확인한 결과
 - 더 긴 기간의 데이터가 있다면 더 정확한 분석을 할 수 있음
 - ✓ 또한 거래량과 같은 추가 지표를 추가해 검색하면 더욱 좋음