In [1]:
```
import pandas as pd
import numpy as np
file_path ='./cafedata/chungnam-pricedata.csv'
df = pd.read_csv(file_path)
df.head()
```

Out[1]:

	뚜레쥬르 지점	뚜레쥬르 공주신월	뚜레쥬르 공주중앙	뚜레쥬르 논산하나로장군마트	뚜레쥬르 논산반월	뚜레쥬르 당진수청	뚜레쥬르 충남부여	뚜레쥬르 서산예천	뚜레쥬르 아산북수	뚜레쥬르 아산터미널	...	뚜레쥬르 홍성삼성마트	뚜레쥬르 천안두정아크로텔	뚜레쥬르 천안쌍용역	뚜레쥬르 대전둔산법원	뚜레쥬르 대전둔산샘머리	뚜레쥬르 대전사학연금	뚜레쥬르 대전도안상대	뚜레쥬르 대전오류중앙	뚜레쥬르 대전버드내	뚜레쥬르 대전태평
0	마늘 단짝 고구마	NaN	NaN	NaN	NaN	4900.0	4900.0	4900.0	4900.0	4900.0	...	4900.0	5400.0	4900.0	4900.0	4900.0	4900.0	4900.0	4900.0	NaN	4900.0
1	깊은 밤 뺑스위스	4900.0	NaN	4900.0	4900.0	4300.0	4300.0	4300.0	4300.0	4300.0	...	4300.0	4700.0	4300.0	4300.0	4300.0	4300.0	4300.0	4300.0	4000.0	4300.0
2	BELT 샌드위치	4300.0	4300.0	4300.0	4300.0	6900.0	6900.0	6900.0	NaN	6900.0	...	7300.0	8300.0	6900.0	6900.0	6900.0	6900.0	NaN	7200.0	NaN	6900.0
3	BLT콥 샐러드	NaN	6900.0	6900.0	6900.0	NaN	NaN	NaN	NaN	8500.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	9500.0	8200.0	8500.0
4	쉬림프 에그 샐러드	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	7300.0	NaN	NaN	10500.0	8200.0	NaN

5 rows × 24 columns

In [2]:
```
print(df.columns)
```

```
Index(['뚜레쥬르 지점', '뚜레쥬르 공주신월', '뚜레쥬르 공주중앙', '뚜레쥬르 논산하나로장군마트', '뚜레쥬르 논산반월',
       '뚜레쥬르 당진수청', '뚜레쥬르 충남부여', '뚜레쥬르 서산예천', '뚜레쥬르 아산북수', '뚜레쥬르 아산터미널',
       '뚜레쥬르 아산탕정', '뚜레쥬르 천안신부', '뚜레쥬르 천안성성신도시', '뚜레쥬르 홍성의료원', '뚜레쥬르 홍성삼성마트',
       '뚜레쥬르 천안두정아크로텔', '뚜레쥬르 천안쌍용역', '뚜레쥬르 대전둔산법원', '뚜레쥬르 대전둔산샘머리',
       '뚜레쥬르 대전사학연금', '뚜레쥬르 대전도안상대', '뚜레쥬르 대전오류중앙', '뚜레쥬르 대전버드내', '뚜레쥬르 대전태평'],
      dtype='object')
```

In [3]:
```
import re

def categorize_menu(df):
    # 키워드 기반 카테고리 매핑 딕셔너리
    category_keywords = {
        '샌드위치류': ['샌드위치', 'BELT', 'BLT', 'V.E.L.T'],
        '샐러드류': ['샐러드'],
        '식빵류': ['식빵', '우유롤', '우유 브레드', '소버식빵'],
        '크림빵': ['크림가득 메론빵','마담 얼그레이 크림번','순진우유크림빵','겹겹이 연유 크림 데니쉬','사르르 고구마케이크빵','사르르 우유크림빵','빵속에리얼초코','카페모카크림빵','까까웨뜨'],
        '피자빵,고로케': ['고로케', '소시지브레드','피자토스트','NEW어니언소시지포카치아'],
        '파이/패스트리': ['바통쉬크레','크라상','애플파이','유자파이'],
        '간식빵': ['소금버터롤','치즈방앗간','깨찰빵','소보로빵','오리지널 커피번','카페모카빵','꽈배기','옛날 단팥 도넛',r'^단팥빵$','단팥소보로빵'],
        '신제품': ['마구마구', '단짝', '뺑스위스']
    }

    # 새로운 카테고리 컬럼 생성
    df['카테고리'] = '기타'  # 기본값

    # 각 메뉴명에 대해 카테고리 매핑
    for idx, menu_name in enumerate(df['뚜레쥬르 지점']):
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
        if pd.isna(menu_name):  # null 체크
            continue

        menu_name = str(menu_name).lower()  # 소문자 변환

        # 각 카테고리의 키워드 체크
        for category, keywords in category_keywords.items():
            if any(keyword.lower() in menu_name for keyword in keywords):
                df.loc[idx, '카테고리'] = category
                break

    return df

def analyze_categories_by_store(df):
    # 매장별 카테고리별 기본 통계
    stores = df.columns[1:-1]  # 첫 번째 열(메뉴명)과 마지막 열(카테고리) 제외

    # 카테고리별 기본 통계
    category_stats = pd.DataFrame()

    for store in stores:
        # 매장별 데이터 숫자로 변환 (오류 방지)
        df[store] = pd.to_numeric(df[store], errors='coerce')

        temp = df.groupby('카테고리').agg({store: 'mean'})
        temp.reset_index(inplace=True)
        temp.rename(columns={store: '평균 가격'}, inplace=True)
        temp['매장명'] = store
        category_stats = pd.concat([category_stats, temp], axis=0)

    return category_stats

def pivot_store_category(stats):
    # 피벗 테이블 생성
    pivot_table = stats.pivot_table(index='매장명', columns='카테고리', values='평균 가격', aggfunc='mean')
    pivot_table=pivot_table.round(1)
    pivot_table.reset_index(inplace=True)
    return pivot_table

# 데이터 로드 및 처리
def process_bakery_data(filepath):
    # csv 파일 읽기
    df = pd.read_csv(filepath)

    # 카테고리 지정
    df = categorize_menu(df)

    # 매장별 카테고리별 분석
    stats = analyze_categories_by_store(df)

    # 피벗 테이블 생성
    pivot_table = pivot_store_category(stats)

    return df, pivot_table

# 파일 처리 및 결과 생성
df, pivot_table = process_bakery_data(file_path)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```python
# 카테고리화된 데이터 및 매장별 통계 표시
from IPython.display import display

# print("카테고리화된 가격 데이터 (처음 5개 행)")
# display(df.head())
#########################################################################################################################

storeinfo_filepath='./adress_process/chungnam_adress.csv'

def process_address(address):
    try:
        # 수동 수정
        if address == '경기도 동탄지성로469번길 60 5단지 상가1동107호,108호,109호':
            return '경기도 화성시'

        # 정규표현식으로 '충청남도 XX시' 추출
        match = re.match(r'충청남도 \w+시', address) or re.match(r'충청남도 \w+군', address) or re.match(r'대전광역시\s+\w+구', address) or re.match(r'대전광역시\s+\w+군', address)

        if match:
            return match.group()

        # 기본값 반환
        return address
    except Exception as e:
        print(f"주소 처리 중 오류 발생: {address}, {e}")
        return address

def load_store_info(storeinfo_filepath):
    store_info = pd.read_csv(storeinfo_filepath)
    # 주소 컬럼 처리
    store_info['주소'] = store_info['주소'].apply(process_address)
    return store_info



def process_bakery_data(price_filepath, store_info_filepath):
    # 가격 데이터 로드
    df = pd.read_csv(price_filepath)

    # 매장 정보 데이터 로드
    store_info = load_store_info(store_info_filepath)

    # 카테고리 지정
    df = categorize_menu(df)

    # 매장별 카테고리별 분석
    stats = analyze_categories_by_store(df)

    # 피벗 테이블 생성 후 매장 정보 병합
    pivot_table = pivot_store_category(stats)
    result = pd.merge(pivot_table, store_info,
                      left_on='매장명',
                      right_on='매장',
                      how='left')

    # 컬럼 순서 재정렬
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    columns = ['매장명', '주소', '지역'] + [col for col in result.columns
              if col not in ['매장명', '매장', '주소', '지역']]
    result = result[columns]

    return df, result

# 실제 파일 경로로 호출
df, result = process_bakery_data('./cafedata/chungnam-pricedata.csv',
                                 './adress_process/chungnam_adress.csv')

# 결과 출력
print("\n매장별 카테고리별 평균 가격 (주소 정보 포함)")
display(result)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

매장별 카테고리별 평균 가격 (주소 정보 포함)

	매장명	주소	지역	간식빵	기타	샌드위치류	샐러드류	식빵류	신제품	크림빵	파이/패스트리	피자빵,고로케
0	뚜레쥬르 공주신월	충청남도 공주시	충남	3660.0	3597.1	3250.0	4700.0	4250.0	4900.0	4175.0	4233.3	2875.0
1	뚜레쥬르 공주중앙	충청남도 공주시	충남	3783.3	3842.9	6640.0	5300.0	4591.7	NaN	3525.0	3200.0	3133.3
2	뚜레쥬르 논산반월	충청남도 논산시	충남	3862.5	3923.9	6880.0	5300.0	5150.0	5250.0	3200.0	4075.0	3133.3
3	뚜레쥬르 논산하나로장군마트	충청남도 논산시	충남	3922.2	4407.8	6600.0	NaN	5191.7	4900.0	3928.6	3400.0	3133.3
4	뚜레쥬르 당진수청	충청남도 당진시	충남	3215.4	4209.4	7160.0	8400.0	5030.8	5033.3	3571.4	2900.0	3183.3
5	뚜레쥬르 대전도안상대	대전광역시 유성구	충남	2920.0	3992.5	7100.0	NaN	4630.0	4600.0	3280.0	3000.0	3260.0
6	뚜레쥬르 대전둔산법원	대전광역시 서구	충남	2922.2	4506.5	7261.5	9300.0	5050.0	4966.7	3583.3	3200.0	3200.0
7	뚜레쥬르 대전둔산샘머리	대전광역시 서구	충남	3011.1	4229.8	7166.7	7300.0	4760.0	4600.0	3700.0	3033.3	3266.7
8	뚜레쥬르 대전버드내	대전광역시 중구	충남	2972.7	3512.1	7766.7	8200.0	5071.4	4000.0	3260.0	2950.0	3166.7
9	뚜레쥬르 대전사학연금	대전광역시 서구	충남	3450.0	3943.8	7185.7	8300.0	5080.0	4600.0	3300.0	NaN	3100.0
10	뚜레쥬르 대전오류중앙	대전광역시 중구	충남	3172.7	4124.1	7490.0	10500.0	4910.0	4600.0	3385.7	2933.3	3383.3
11	뚜레쥬르 대전태평	대전광역시 중구	충남	2870.0	4300.0	7227.3	NaN	5212.5	4933.3	3557.1	3100.0	3142.9
12	뚜레쥬르 서산예천	충청남도 서산시	충남	3266.7	4076.6	7166.7	NaN	4725.0	4933.3	3775.0	3000.0	3240.0
13	뚜레쥬르 아산북수	충청남도 아산시	충남	3480.0	4767.4	7400.0	NaN	5316.7	4600.0	3850.0	3466.7	3640.0
14	뚜레쥬르 아산탕정	충청남도 아산시	충남	2966.7	4285.7	7250.0	8300.0	5283.3	4600.0	3233.3	2933.3	3200.0
15	뚜레쥬르 아산터미널	충청남도 아산시	충남	3137.5	4396.6	7275.0	8500.0	4985.7	4600.0	4028.6	3250.0	3150.0
16	뚜레쥬르 천안두정아크로텔	충청남도 천안시	충남	4166.7	5008.2	8342.9	11000.0	5487.5	5050.0	5166.7	3666.7	3820.0
17	뚜레쥬르 천안성성신도시	충청남도 천안시	충남	3430.0	4456.0	8318.2	8900.0	5090.0	5400.0	3757.1	3200.0	3420.0
18	뚜레쥬르 천안신부	충청남도 천안시	충남	3170.0	4142.1	7142.9	8350.0	4950.0	4600.0	3622.2	3100.0	3025.0
19	뚜레쥬르 천안쌍용역	충청남도 천안시	충남	3166.7	4456.7	7200.0	NaN	4837.5	4600.0	2833.3	2933.3	3300.0
20	뚜레쥬르 충남부여	충청남도 부여군	충남	3027.3	4267.7	7100.0	NaN	4912.5	4933.3	3660.0	2966.7	3183.3
21	뚜레쥬르 홍성삼성마트	충청남도 홍성군	충남	2777.8	4283.7	7320.0	NaN	4740.0	4600.0	3480.0	2933.3	3200.0
22	뚜레쥬르 홍성의료원	충청남도 홍성군	충남	2862.5	4152.1	7166.7	NaN	5192.3	4600.0	3614.3	3100.0	3225.0

In [4]:
```
grouped_data = result.groupby('주소')[['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']].mean().mean(axis=1).sort_values(ascending=False)

# groupby 결과를 데이터프레임으로 변환
grouped_df = pd.DataFrame(grouped_data).reset_index()

# 컬럼명 변경
grouped_df.columns = ['주소', '평균가격']

# CSV 파일로 저장
grouped_df.to_csv('anal_chungnam/구별_빵_평균가격.csv', index=False, encoding='utf-8-sig')
grouped_df
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[4]:

	주소	평균가격
0	충청남도 천안시	5139.480208
1	대전광역시 중구	4881.483333
2	충청남도 아산시	4868.616667
3	충청남도 당진시	4811.775000
4	대전광역시 서구	4768.910417
5	충청남도 논산시	4576.662500
6	충청남도 서산시	4300.957143
7	충청남도 부여군	4254.728571
8	충청남도 홍성군	4200.850000
9	충청남도 공주시	4194.787500
10	대전광역시 유성구	4112.857143

In [5]:
```
categories = ['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']

# 각 카테고리별로 구의 평균 가격 계산
grouped_data = {}
for category in categories:
    grouped_data[category] = result.groupby('주소')[category].mean().round(2)

# 데이터프레임 생성
grouped_df = pd.DataFrame(grouped_data)

# CSV 파일로 저장
grouped_df.to_csv('anal_chungnam/시군별_카테고리_평균가격.csv', encoding='utf-8-sig')
grouped_df
```

Out[5]:

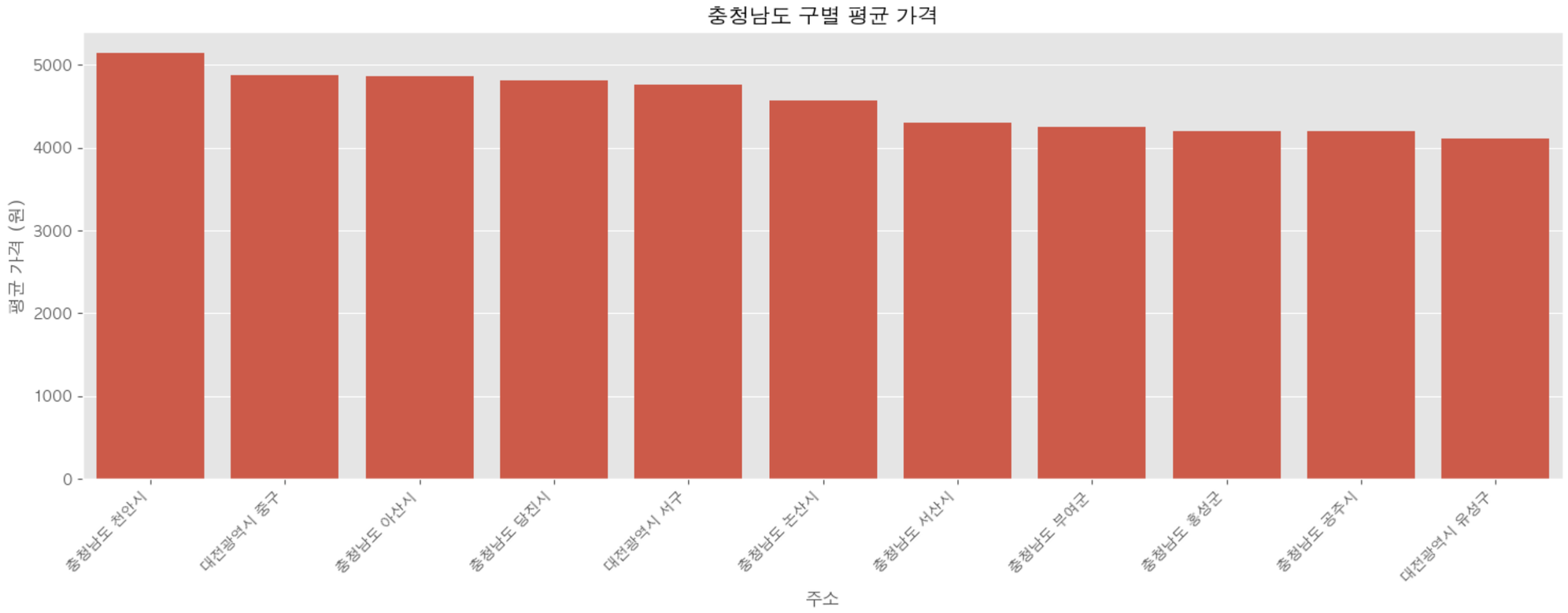
주소	간식빵	샌드위치류	샐러드류	식빵류	신제품	크림빵	파이/패스트리	피자빵,고로케
대전광역시 서구	3127.77	7204.63	8300.00	4963.33	4722.23	3527.77	3116.65	3188.90
대전광역시 유성구	2920.00	7100.00	NaN	4630.00	4600.00	3280.00	3000.00	3260.00
대전광역시 중구	3005.13	7494.67	9350.00	5064.63	4511.10	3400.93	2994.43	3230.97
충청남도 공주시	3721.65	4945.00	5000.00	4420.85	4900.00	3850.00	3716.65	3004.15
충청남도 논산시	3892.35	6740.00	5300.00	5170.85	5075.00	3564.30	3737.50	3133.30
충청남도 당진시	3215.40	7160.00	8400.00	5030.80	5033.30	3571.40	2900.00	3183.30
충청남도 부여군	3027.30	7100.00	NaN	4912.50	4933.30	3660.00	2966.70	3183.30
충청남도 서산시	3266.70	7166.70	NaN	4725.00	4933.30	3775.00	3000.00	3240.00
충청남도 아산시	3194.73	7308.33	8400.00	5195.23	4600.00	3703.97	3216.67	3330.00
		7751.00	9416.67	5091.25	4912.50	3844.82	3225.00	3391.25

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

충청남도 홍성군	2820.15	7243.35	NaN	4966.15	4600.00	3547.15	3016.65	3212.50

In [6]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from matplotlib import font_manager, rc

# Mac OS 용 폰트 설정
plt.rc('font', family='AppleGothic') # 맥용 폰트 설정

# 그래프 기본 설정
plt.rcParams['axes.unicode_minus'] = False
plt.style.use('ggplot')


# 1. 구별 전체 평균 가격 분석
plt.figure(figsize=(15, 6))
grouped_data = result.groupby('주소')[['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']].mean().mean(axis=1).sort_values(ascending=False)

sns.barplot(x=grouped_data.index, y=grouped_data.values)
plt.title('충청남도 구별 평균 가격')
plt.xticks(rotation=45, ha='right')
plt.ylabel('평균 가격 (원)')
plt.tight_layout()
plt.show()

# 2. 카테고리별 가격 분포 (박스플롯)
plt.figure(figsize=(12, 6))
categories = ['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']
data_melted = pd.melt(result, value_vars=categories)

sns.boxplot(x='variable', y='value', data=data_melted)
plt.title('카테고리별 가격 분포')
plt.xticks(rotation=45, ha='right')
plt.xlabel('카테고리')
plt.ylabel('가격 (원)')
plt.tight_layout()
plt.show()
```

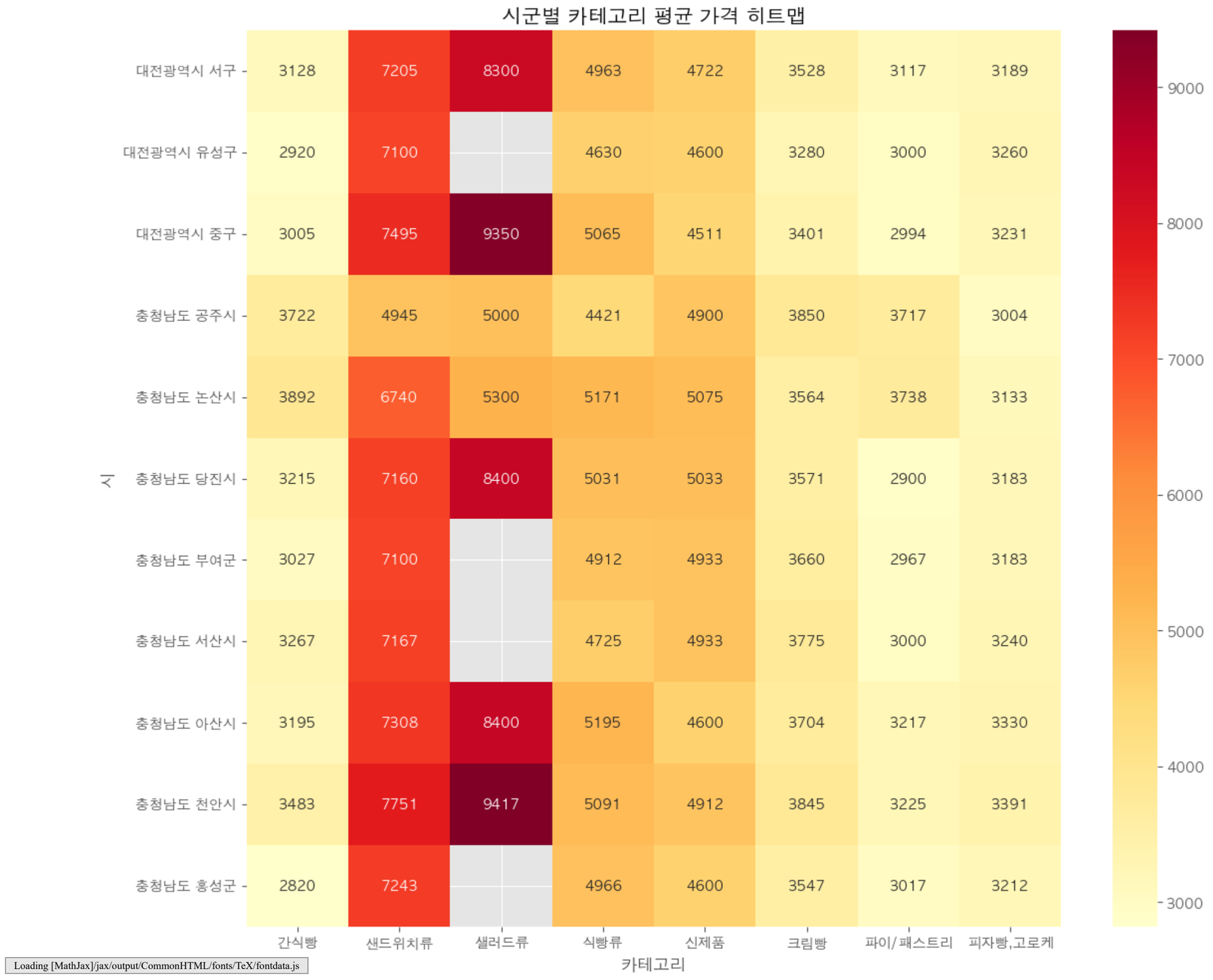
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



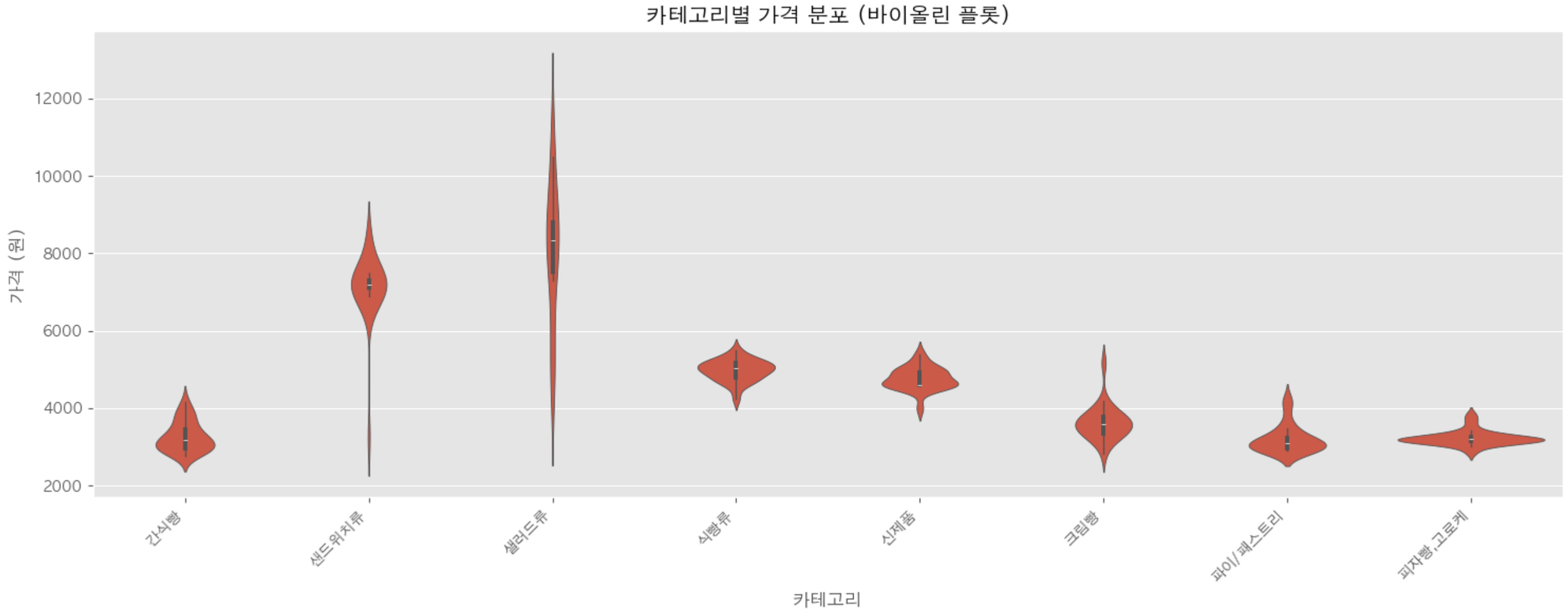
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [7]:
```python
# 3. 구별/카테고리별 평균 가격 히트맵
plt.figure(figsize=(12, 10))
pivot_data = result.groupby('주소')[categories].mean()
sns.heatmap(pivot_data, annot=True, fmt='.0f', cmap='YlOrRd')
plt.title('시군별 카테고리 평균 가격 히트맵')
plt.ylabel('시')
plt.xlabel('카테고리')
plt.tight_layout()
plt.show()

# 5. 카테고리별 가격 분포 (바이올린 플롯)
plt.figure(figsize=(15, 6))
sns.violinplot(x='variable', y='value', data=data_melted)
plt.title('카테고리별 가격 분포 (바이올린 플롯)')
plt.xticks(rotation=45, ha='right')
plt.xlabel('카테고리')
plt.ylabel('가격 (원)')
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [8]:
```python
# 1. 구별 평균 빵가격 계산
categories = ['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']
bread_price_by_district = result.groupby('주소')[categories].mean().mean(axis=1).reset_index()
bread_price_by_district.columns = ['구분', '평균_빵가격']
# '충청남도'제거
bread_price_by_district['구분'] = bread_price_by_district['구분'].str.replace('충청남도', '').str.strip()

# 아파트 가격 데이터 전처리
apt_price = pd.read_csv('anal_chungnam/chungnam_APT_PRICE.csv')
# '충청남도'와 '구' 제거
apt_price['구분'] = apt_price['구분'].str.replace('충청남도', '').str.strip()

apt_price['매매'] = pd.to_numeric(apt_price['매매'].str.replace(',', ''), errors='coerce')
apt_price = apt_price.dropna()  # 결측치 제거
apt_price = apt_price[~apt_price['구분'].str.contains('대전광역시 유성구')]


# 데이터 확인
print("전처리 후 구별 빵가격 데이터:")
print(bread_price_by_district)
print("전처리 후 아파트 가격 데이터:")
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

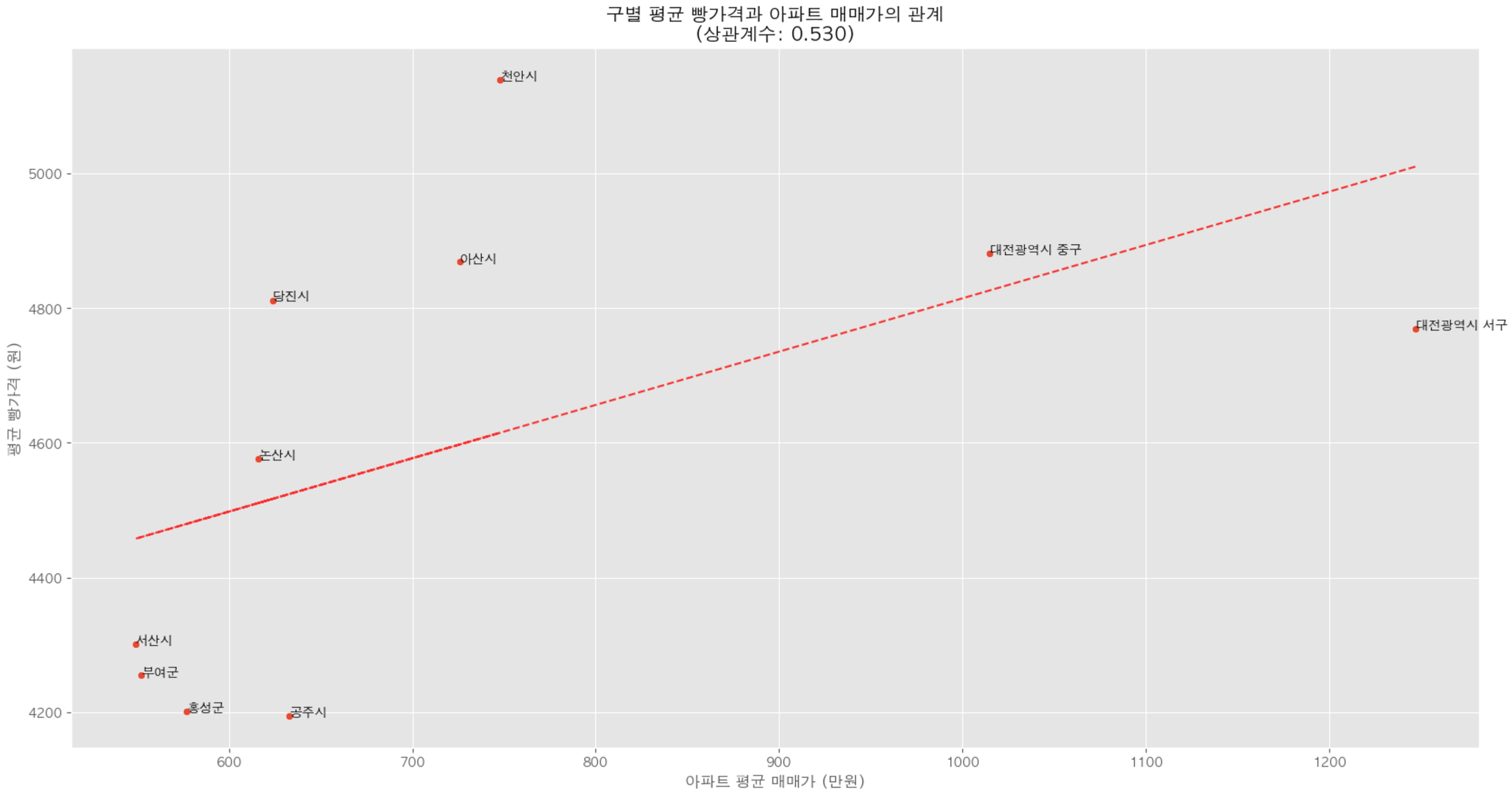
```python
print(apt_price)

# 데이터 병합
merged_df = pd.merge(bread_price_by_district, apt_price[['구분', '매매']], on='구분', how='inner')
print("\n병합된 데이터:")
print(merged_df)

# 시각화
if not merged_df.empty:
    plt.figure(figsize=(20, 10))
    sns.scatterplot(data=merged_df, x='매매', y='평균_빵가격')

    # 추세선 추가
    x = merged_df['매매'].values
    y = merged_df['평균_빵가격'].values
    z = np.polyfit(x, y, 1)
    p = np.poly1d(z)
    plt.plot(x, p(x), "r--", alpha=0.8)

    # 각 점에 구 이름 표시
    for idx, row in merged_df.iterrows():
        plt.annotate(row['구분'], (row['매매'], row['평균_빵가격']))

    correlation = merged_df['평균_빵가격'].corr(merged_df['매매'])
    plt.title(f'구별 평균 빵가격과 아파트 매매가의 관계\n(상관계수: {correlation:.3f})')
    plt.xlabel('아파트 평균 매매가 (만원)')
    plt.ylabel('평균 빵가격 (원)')

    print(f"\n상관계수: {correlation:.3f}")
    if correlation > 0:
        print("양의 상관관계가 있습니다: 아파트 가격이 높은 구일수록 빵 가격도 높은 경향이 있습니다.")
    else:
        print("음의 상관관계가 있습니다: 아파트 가격이 높은 구일수록 빵 가격이 낮은 경향이 있습니다.")
```

```
전처리 후 구별 빵가격 데이터:
          구분        평균_빵가격
0    대전광역시 서구  4768.910417
1   대전광역시 유성구  4112.857143
2    대전광역시 중구  4881.483333
3         공주시  4194.787500
4         논산시  4576.662500
5         당진시  4811.775000
6         부여군  4254.728571
7         서산시  4300.957143
8         아산시  4868.616667
9         천안시  5139.480208
10        홍성군  4200.850000

전처리 후 아파트 가격 데이터:
          구분    매매   전세
0         계룡시   714  526
1         공주시   633  442
2         금산군   514  372
3         논산시   616  446
4         당진시   624  496
5         보령시   573  433
6         부여군   552  407
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
8         서천군   418  278
9         아산시   726  539
10        예산군   480  324
11        천안시   748  574
12        청양군   481  196
13        태안군   454  327
14        홍성군   577  429
15  대전광역시 대덕구   811  582
16   대전광역시 동구   880  640
17   대전광역시 서구  1247  829
19   대전광역시 중구  1015  700

병합된 데이터:
        구분        평균_빵가격    매매
0  대전광역시 서구  4768.910417  1247
1  대전광역시 중구  4881.483333  1015
2       공주시  4194.787500   633
3       논산시  4576.662500   616
4       당진시  4811.775000   624
5       부여군  4254.728571   552
6       서산시  4300.957143   549
7       아산시  4868.616667   726
8       천안시  5139.480208   748
9       홍성군  4200.850000   577

상관계수: 0.530
양의 상관관계가 있습니다: 아파트 가격이 높은 구일수록 빵 가격도 높은 경향이 있습니다.
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [9]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

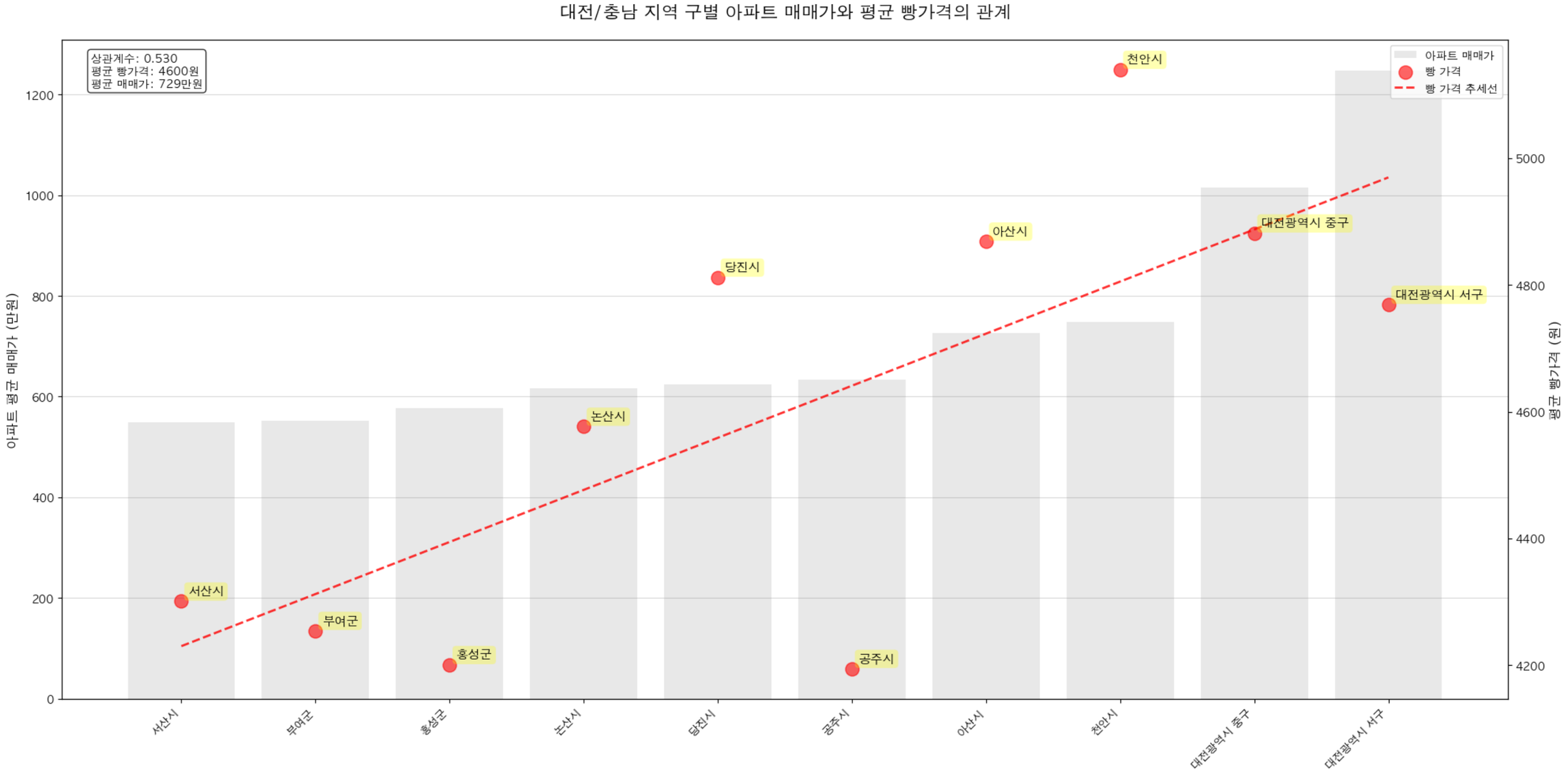
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```python
# 시각화 기본 설정
plt.style.use('default')  # 기본 스타일 사용
plt.rcParams['figure.figsize'] = (20, 10)
plt.rcParams['font.family'] = 'AppleGothic'

if not merged_df.empty:
    # 데이터 정렬
    sorted_df = merged_df.sort_values(by='매매')

    # 메인 그래프 생성
    fig, ax1 = plt.subplots(figsize=(20, 10))

    # 기본 그리드 제거
    ax1.grid(False)

    # 바차트 배경 (매매가)
    ax1.bar(range(len(sorted_df)), sorted_df['매매'], alpha=0.5, color='lightgray', label='아파트 매매가')

    # 산점도를 위한 두 번째 y축 생성
    ax2 = ax1.twinx()
    ax2.grid(False)

    # 산점도 그리기
    scatter = ax2.scatter(range(len(sorted_df)), sorted_df['평균_빵가격'],
                          s=150, alpha=0.6, color='red', label='빵 가격')

    # 추세선 추가
    z = np.polyfit(range(len(sorted_df)), sorted_df['평균_빵가격'], 1)
    p = np.poly1d(z)
    ax2.plot(range(len(sorted_df)), p(range(len(sorted_df))), "r--",
             linewidth=2, alpha=0.8, label='빵 가격 추세선')

    # x축 레이블 설정 (45도 회전)
    ax1.set_xticks(range(len(sorted_df)))
    ax1.set_xticklabels(sorted_df['구분'], rotation=45, ha='right')

    # 각 점에 구 이름 표시
    for i, row in enumerate(sorted_df.itertuples()):
        ax2.annotate(row.구분,
                     (i, row.평균_빵가격),
                     xytext=(6, 6),
                     textcoords='offset points',
                     fontsize=11,
                     bbox=dict(boxstyle='round,pad=0.3', fc='yellow', alpha=0.3, ec='none'))
                     # arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))

    # 평균선 추가
    # ax2.axhline(y=sorted_df['평균_빵가격'].mean(), color='g', linestyle='--', alpha=0.3, label='평균 빵가격')
    # ax1.axhline(y=sorted_df['매매'].mean(), color='b', linestyle='--', alpha=0.3, label='평균 매매가')

    # y축 그리드만 추가
    ax1.grid(True, axis='y', alpha=0.3, linestyle='-', color='gray')

    # 축 레이블 설정
    ax1.set_ylabel('아파트 평균 매매가 (만원)', fontsize=12)
    ax2.set_ylabel('평균 빵가격 (원)', fontsize=12)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```python
# 상관계수 계산
correlation = sorted_df['평균_빵가격'].corr(sorted_df['매매'])

# 그래프 제목 설정
plt.title('대전/충남 지역 구별 아파트 매매가와 평균 빵가격의 관계', fontsize=16, pad=20)

# 통계 정보 추가
stats_text = f'상관계수: {correlation:.3f}\n'
stats_text += f'평균 빵가격: {sorted_df["평균_빵가격"].mean():.0f}원\n'
stats_text += f'평균 매매가: {sorted_df["매매"].mean():.0f}만원'
ax1.text(0.02, 0.98, stats_text,
         transform=ax1.transAxes,
         verticalalignment='top',
         bbox=dict(boxstyle='round', facecolor='white', alpha=0.8))

# 범례 추가
lines1, labels1 = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax1.legend(lines1 + lines2, labels1 + labels2, loc='upper right')

# 여백 조정
plt.tight_layout()

# 그래프 표시
plt.show()

# 추가 분석 출력
print("\n=== 지역별 상세 데이터 ===")
analysis_df = merged_df.copy()
analysis_df['가격_차이_순위'] = analysis_df['매매'].rank(ascending=False)
analysis_df['빵가격_순위'] = analysis_df['평균_빵가격'].rank(ascending=False)
analysis_df['순위_차이'] = abs(analysis_df['가격_차이_순위'] - analysis_df['빵가격_순위'])

print("\n아파트 가격 상위 5개 지역:")
print(analysis_df.nlargest(5, '매매')[['구분', '매매', '평균_빵가격']])
print("\n빵 가격 상위 5개 지역:")
print(analysis_df.nlargest(5, '평균_빵가격')[['구분', '매매', '평균_빵가격']])
print("\n순위 차이가 가장 큰 5개 지역 (불일치도가 높은 지역):")
print(analysis_df.nlargest(5, '순위_차이')[['구분', '매매', '평균_빵가격', '순위_차이']])

print(f"\n상관계수: {correlation:.3f}")
if correlation > 0:
    print("양의 상관관계가 있습니다: 아파트 가격이 높은 지역일수록 빵 가격도 높은 경향이 있습니다.")
else:
    print("음의 상관관계가 있습니다: 아파트 가격이 높은 지역일수록 빵 가격이 낮은 경향이 있습니다.")
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js




```
=== 지역별 상세 데이터 ===

아파트 가격 상위 5개 지역:
          구분     매매       평균_빵가격
0  대전광역시 서구  1247  4768.910417
1  대전광역시 중구  1015  4881.483333
2         천안시   740  5130.420208
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
7       아산시   726  4868.616667
2       공주시   633  4194.787500

빵 가격 상위 5개 지역:
         구분     매매       평균_빵가격
8       천안시   748  5139.480208
1  대전광역시 중구  1015  4881.483333
7       아산시   726  4868.616667
4       당진시   624  4811.775000
0  대전광역시 서구  1247  4768.910417

순위 차이가 가장 큰 5개 지역 (불일치도가 높은 지역):
         구분     매매       평균_빵가격  순위_차이
2       공주시   633  4194.787500    5.0
0  대전광역시 서구  1247  4768.910417    4.0
6       서산시   549  4300.957143    3.0
4       당진시   624  4811.775000    2.0
8       천안시   748  5139.480208    2.0

상관계수: 0.530
양의 상관관계가 있습니다: 아파트 가격이 높은 지역일수록 빵 가격도 높은 경향이 있습니다.
```

In [10]:
```python
import pandas as pd
import numpy as np
cbile_path ='./cafedata/chungbuk-pricedata.csv'
cbdf = pd.read_csv(cbile_path)
cbdf.head()
```

Out[10]:

	뚜레쥬르 지점	뚜레쥬르 충북음성	뚜레쥬르 제천하소로	뚜레쥬르 증평하나로마트	뚜레쥬르 진천성모병원	뚜레쥬르 청원내수	뚜레쥬르 청주용암현대	뚜레쥬르 청주용암부영	뚜레쥬르 청주모충	뚜레쥬르 청주율량	뚜레쥬르 청주가경	뚜레쥬르 청주테크노폴리스	뚜레쥬르 서충주파라뷰	뚜레쥬르 충주용산호암	뚜레쥬르 충북옥천	뚜레쥬르 충북금왕중앙
0	마늘 단짝 고구마	4900.0	4900.0	4900.0	4900.0	4900.0	4900.0	4900.0	4900.0	4900.0	NaN	4900.0	4900.0	4900.0	4900.0	4900.0
1	깊은 밤 뺑스위스	4300.0	4300.0	4300.0	4300.0	4300.0	4300.0	4300.0	4300.0	4300.0	4300.0	NaN	4300.0	4300.0	4300.0	4300.0
2	BELT 샌드위치	6900.0	6900.0	6900.0	6900.0	6900.0	6900.0	6900.0	6900.0	6900.0	6900.0	NaN	6900.0	6900.0	NaN	6900.0
3	BLT콥 샐러드	NaN	8500.0	NaN	NaN	8500.0	NaN	NaN	NaN	NaN	NaN	8500.0	NaN	NaN	NaN	8500.0
4	쉬림프 에그 샐러드	NaN	NaN	NaN	NaN	NaN	10500.0	NaN	NaN	NaN	NaN	NaN	10500.0	NaN	NaN	NaN

In [11]:
```python
import re

def categorize_menu(gbdf):
    # 키워드 기반 카테고리 매핑 딕셔너리
    category_keywords = {
        '샌드위치류': ['샌드위치', 'BELT', 'BLT', 'V.E.L.T'],
        '샐러드류': ['샐러드'],
        '식빵류': ['식빵', '우유롤', '우유 브레드', '소버식빵'],
        '크림빵': ['크림가득 메론빵','마담 얼그레이 크림번','순진우유크림빵','겹겹이 연유 크림 데니쉬','사르르 고구마케이크빵','사르르 우유크림빵','빵속에리얼초코','카페모카크림빵','까까웨뜨'],
        '피자빵,고로케': ['고로케', '소시지브레드','피자토스트','NEW어니언소시지포카치아'],
        '파이/패스트리': ['바통쉬크레','크라상','애플파이','유자파이'],
        '간식빵': ['소금버터롤','치즈방앗간','깨찰빵','소보로빵','오리지널 커피번','카페모카빵','꽈배기','옛날 단팥 도넛',r'^단팥빵$','단팥소보로빵'],
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
        '신제품': ['마구마구', '단짝', '뺑스위스']
    }

    # 새로운 카테고리 컬럼 생성
    gbdf['카테고리'] = '기타'  # 기본값

    # 각 메뉴명에 대해 카테고리 매핑
    for idx, menu_name in enumerate(gbdf['뚜레쥬르 지점']):
        if pd.isna(menu_name):  # null 체크
            continue

        menu_name = str(menu_name).lower()  # 소문자 변환

        # 각 카테고리의 키워드 체크
        for category, keywords in category_keywords.items():
            if any(keyword.lower() in menu_name for keyword in keywords):
                gbdf.loc[idx, '카테고리'] = category
                break

    return gbdf

def analyze_categories_by_store(gbdf):
    # 매장별 카테고리별 기본 통계
    stores = gbdf.columns[1:-1]  # 첫 번째 열(메뉴명)과 마지막 열(카테고리) 제외

    # 카테고리별 기본 통계
    category_stats = pd.DataFrame()

    for store in stores:
        # 매장별 데이터 숫자로 변환 (오류 방지)
        gbdf[store] = pd.to_numeric(gbdf[store], errors='coerce')

        temp = gbdf.groupby('카테고리').agg({store: 'mean'})
        temp.reset_index(inplace=True)
        temp.rename(columns={store: '평균 가격'}, inplace=True)
        temp['매장명'] = store
        category_stats = pd.concat([category_stats, temp], axis=0)

    return category_stats

def pivot_store_category(stats):
    # 피벗 테이블 생성
    pivot_table = stats.pivot_table(index='매장명', columns='카테고리', values='평균 가격', aggfunc='mean')
    pivot_table=pivot_table.round(1)
    pivot_table.reset_index(inplace=True)
    return pivot_table

# 데이터 로드 및 처리
def process_bakery_data(cbile_path):
    # csv 파일 읽기
    gbdf = pd.read_csv(cbile_path)

    # 카테고리 지정
    gbdf = categorize_menu(gbdf)

    # 매장별 카테고리별 분석
    stats = analyze_categories_by_store(gbdf)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    # 피벗 테이블 생성
    pivot_table = pivot_store_category(stats)

    return gbdf, pivot_table

# 파일 처리 및 결과 생성
gbdf, pivot_table = process_bakery_data(cbile_path)

# 카테고리화된 데이터 및 매장별 통계 표시
from IPython.display import display

# print("카테고리화된 가격 데이터 (처음 5개 행)")
# display(gbdf.head())
#########################################################################################################################

storeinfo_filepath='./adress_process/뚜레쥬르_매장정보_충청북도_수정.csv'

def process_address(address):
    try:
        # 수동 수정
        if address == '경기도 동탄지성로469번길 60 5단지 상가1동107호,108호,109호':
            return '경기도 화성시'

        # 정규표현식으로 '충청북도 XX시' 추출
        match = re.match(r'충청북도\s+\w+시', address) or re.match(r'충청북도\s+\w+군', address) or re.match(r'충청북도\s+\w+구', address)

        if match:
            return match.group()

        # 기본값 반환
        return address
    except Exception as e:
        print(f"주소 처리 중 오류 발생: {address}, {e}")
        return address

def load_store_info(storeinfo_filepath):
    store_info = pd.read_csv(storeinfo_filepath)
    # 주소 컬럼 처리
    store_info['주소'] = store_info['주소'].apply(process_address)
    return store_info



def process_bakery_data(price_filepath, store_info_filepath):
    # 가격 데이터 로드
    gbdf = pd.read_csv(price_filepath)

    # 매장 정보 데이터 로드
    store_info = load_store_info(store_info_filepath)

    # 카테고리 지정
    gbdf = categorize_menu(gbdf)

    # 매장별 카테고리별 분석
    stats = analyze_categories_by_store(gbdf)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    # 피벗 테이블 생성 후 매장 정보 병합
    pivot_table = pivot_store_category(stats)
    result = pd.merge(pivot_table, store_info,
                      left_on='매장명',
                      right_on='매장',
                      how='left')

    # 컬럼 순서 재정렬
    columns = ['매장명', '주소', '지역'] + [col for col in result.columns
              if col not in ['매장명', '매장', '주소', '지역']]
    result = result[columns]

    return gbdf, result

# 실제 파일 경로로 호출
gbdf, result = process_bakery_data('./cafedata/chungbuk-pricedata.csv',
                                   './adress_process/chungbuk_adress.csv')

# 결과 출력
print("\n매장별 카테고리별 평균 가격 (주소 정보 포함)")
display(result)

result.to_csv('./anal_chungbuk/?시별_카테고리_평균가격.csv', encoding='utf-8-sig')
```

매장별 카테고리별 평균 가격 (주소 정보 포함)

	매장명	주소	지역	간식빵	기타	샌드위치류	샐러드류	식빵류	신제품	크림빵	파이/패스트리	피자빵,고로케
0	뚜레쥬르 서충주파라뷰	충청북도 충주시	충북	3161.5	4208.9	6230.0	8600.0	5620.0	4600.0	3587.5	3000.0	3100.0
1	뚜레쥬르 제천하소로	충청북도 제천시	충북	3033.3	4275.0	6438.9	10200.0	5294.1	4600.0	3500.0	3066.7	3220.0
2	뚜레쥬르 증평하나로마트	충청북도 증평군	충북	3100.0	4726.7	6700.0	NaN	4983.3	4600.0	3500.0	2933.3	2750.0
3	뚜레쥬르 진천성모병원	충청북도 진천군	충북	3075.0	4340.5	7150.0	3000.0	5010.0	4600.0	3433.3	3200.0	3025.0
4	뚜레쥬르 청원내수	충청북도 청원군	충북	2925.0	4327.5	6733.3	3000.0	5023.1	4600.0	3300.0	3066.7	3100.0
5	뚜레쥬르 청주가경	충청북도 청주시	충북	2920.0	3854.8	7085.7	NaN	4714.3	4300.0	3066.7	2400.0	3180.0
6	뚜레쥬르 청주모충	충청북도 청주시	충북	2975.0	4176.5	7050.0	NaN	5077.8	4600.0	3250.0	2933.3	3025.0
7	뚜레쥬르 청주용암부영	충청북도 청주시	충북	3112.5	4320.0	6542.9	4700.0	5041.7	4600.0	3240.0	3200.0	3025.0
8	뚜레쥬르 청주용암현대	충청북도 청주시	충북	3236.4	4460.9	7220.0	9066.7	5176.9	4600.0	3357.1	2800.0	3040.0
9	뚜레쥬르 청주율량	충청북도 청주시	충북	3020.0	4044.2	7100.0	NaN	5086.7	4600.0	3266.7	2933.3	3040.0
10	뚜레쥬르 청주테크노폴리스	충청북도 청주시	충북	2775.0	3734.3	7550.0	NaN	5340.0	4900.0	3675.0	2800.0	3120.0
11	뚜레쥬르 충북금왕중앙	충청북도 음성군	충북	3125.0	4107.1	7038.5	NaN	5500.0	4600.0	2400.0	2400.0	2850.0
12	뚜레쥬르 충북옥천	충청북도 옥천군	충북	3142.9	4633.3	7220.0	NaN	4975.0	4600.0	3800.0	2400.0	3250.0
13	뚜레쥬르 충북음성	충청북도 음성군	충북	2970.0	4346.3	6085.7	NaN	4961.5	5066.7	3483.3	2800.0	3025.0
14	뚜레쥬르 충주용산호암	충청북도 충주시	충북	3155.6	4228.9	7050.0	NaN	5283.3	4600.0	3525.0	2933.3	3333.3

In [12]:
```
grouped_data = result.groupby('주소')[['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']].mean().mean(axis=1).sort_values(ascending=False)

으로 변환
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
grouped_df = pd.DataFrame(grouped_data).reset_index()

# 컬럼명 변경
grouped_df.columns = ['주소', '평균가격']

# CSV 파일로 저장
grouped_df.to_csv('anal_chungbuk/시군별_빵_평균가격.csv', index=False, encoding='utf-8-sig')
grouped_df
```

Out[12]:

	주소	평균가격
0	충청북도 제천시	4919.125000
1	충청북도 충주시	4773.718750
2	충청북도 청주시	4484.939583
3	충청북도 옥천군	4198.271429
4	충청북도 증평군	4080.942857
5	충청북도 진천군	4061.662500
6	충청북도 음성군	4021.835714
7	충청북도 청원군	3968.512500

In [13]:

```
categories = ['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']

# 각 카테고리별로 구의 평균 가격 계산
grouped_data = {}
for category in categories:
    grouped_data[category] = result.groupby('주소')[category].mean().round(2)

# 데이터프레임 생성
grouped_df = pd.DataFrame(grouped_data)

# CSV 파일로 저장
grouped_df.to_csv('anal_chungbuk/시군별_카테고리_평균가격.csv', encoding='utf-8-sig')
grouped_df
```

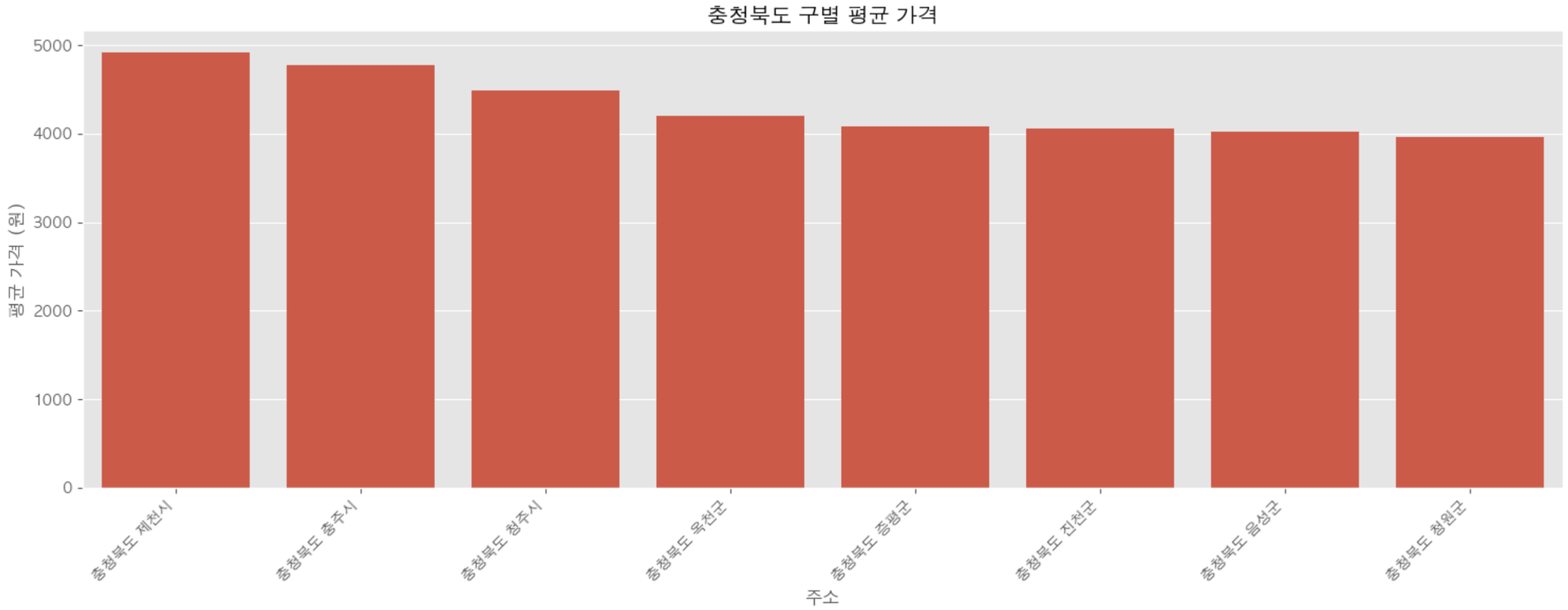
Out[13]:

주소	간식빵	샌드위치류	샐러드류	식빵류	신제품	크림빵	파이/패스트리	피자빵,고로케
충청북도 옥천군	3142.90	7220.00	NaN	4975.00	4600.00	3800.00	2400.00	3250.00
충청북도 음성군	3047.50	6562.10	NaN	5230.75	4833.35	2941.65	2600.00	2937.50
충청북도 제천시	3033.30	6438.90	10200.00	5294.10	4600.00	3500.00	3066.70	3220.00
충청북도 증평군	3100.00	6700.00	NaN	4983.30	4600.00	3500.00	2933.30	2750.00
충청북도 진천군	3075.00	7150.00	3000.00	5010.00	4600.00	3433.30	3200.00	3025.00
충청북도 청원군	2925.00	6733.30	3000.00	5023.10	4600.00	3300.00	3066.70	3100.00
충청북도 청주시	3006.48	7091.43	6883.35	5072.90	4600.00	3309.25	2844.43	3071.67
충청북도 충주시	3150.55	6640.00	8600.00	5451.65	4600.00	3556.25	2966.65	3216.65

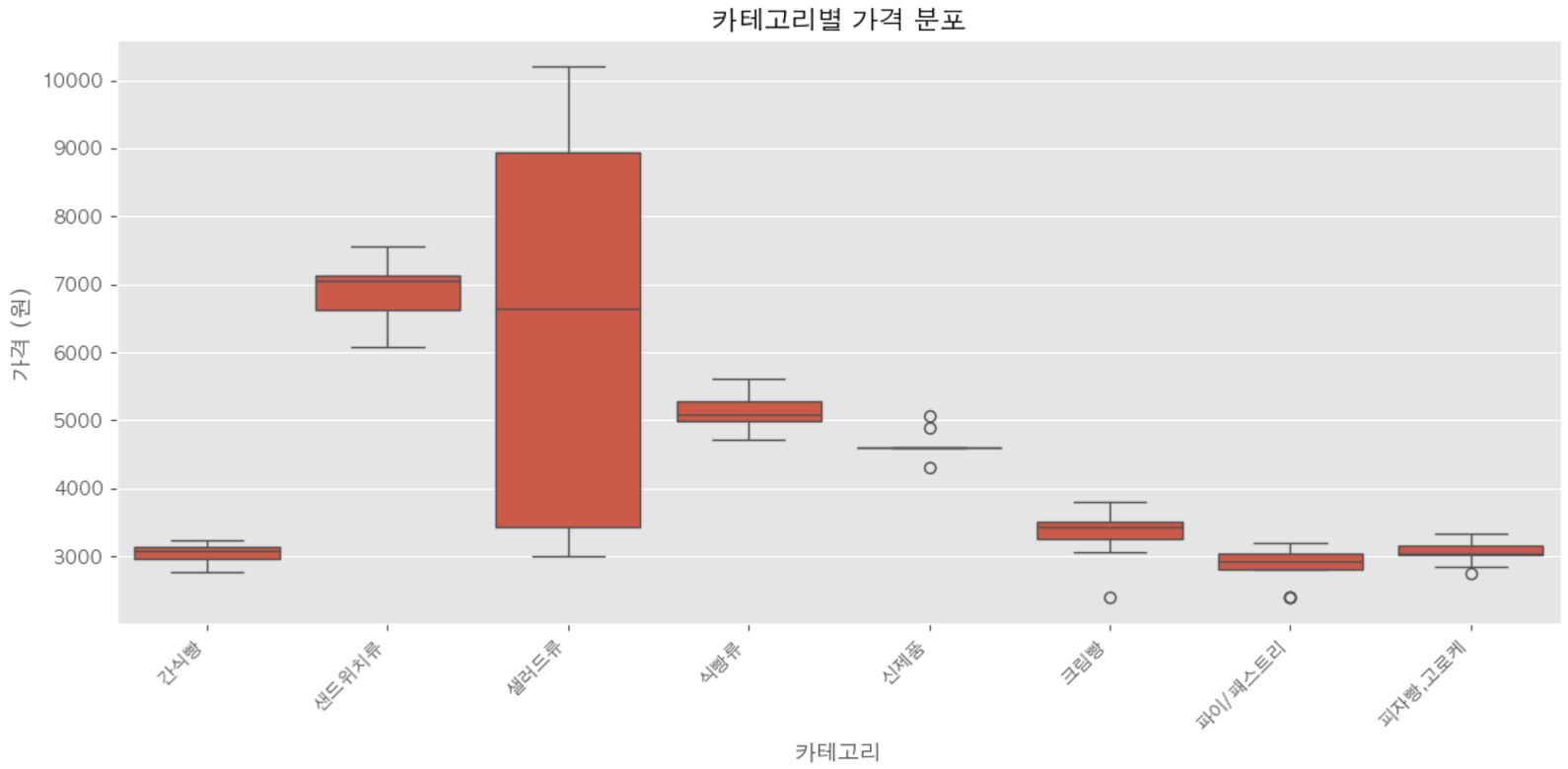
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [14]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from matplotlib import font_manager, rc

# Mac OS 용 폰트 설정
plt.rc('font', family='AppleGothic') # 맥용 폰트 설정

# 그래프 기본 설정
plt.rcParams['axes.unicode_minus'] = False
plt.style.use('ggplot')


# 1. 구별 전체 평균 가격 분석
plt.figure(figsize=(15, 6))
grouped_data = result.groupby('주소')[['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']].mean().mean(axis=1).sort_values(ascending=False)

sns.barplot(x=grouped_data.index, y=grouped_data.values)
plt.title('충청북도 구별 평균 가격')
plt.xticks(rotation=45, ha='right')
plt.ylabel('평균 가격 (원)')
plt.tight_layout()
plt.show()

# 2. 카테고리별 가격 분포 (박스플롯)
plt.figure(figsize=(12, 6))
categories = ['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']
data_melted = pd.melt(result, value_vars=categories)

sns.boxplot(x='variable', y='value', data=data_melted)
plt.title('카테고리별 가격 분포')
plt.xticks(rotation=45, ha='right')
plt.xlabel('카테고리')
plt.ylabel('가격 (원)')
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

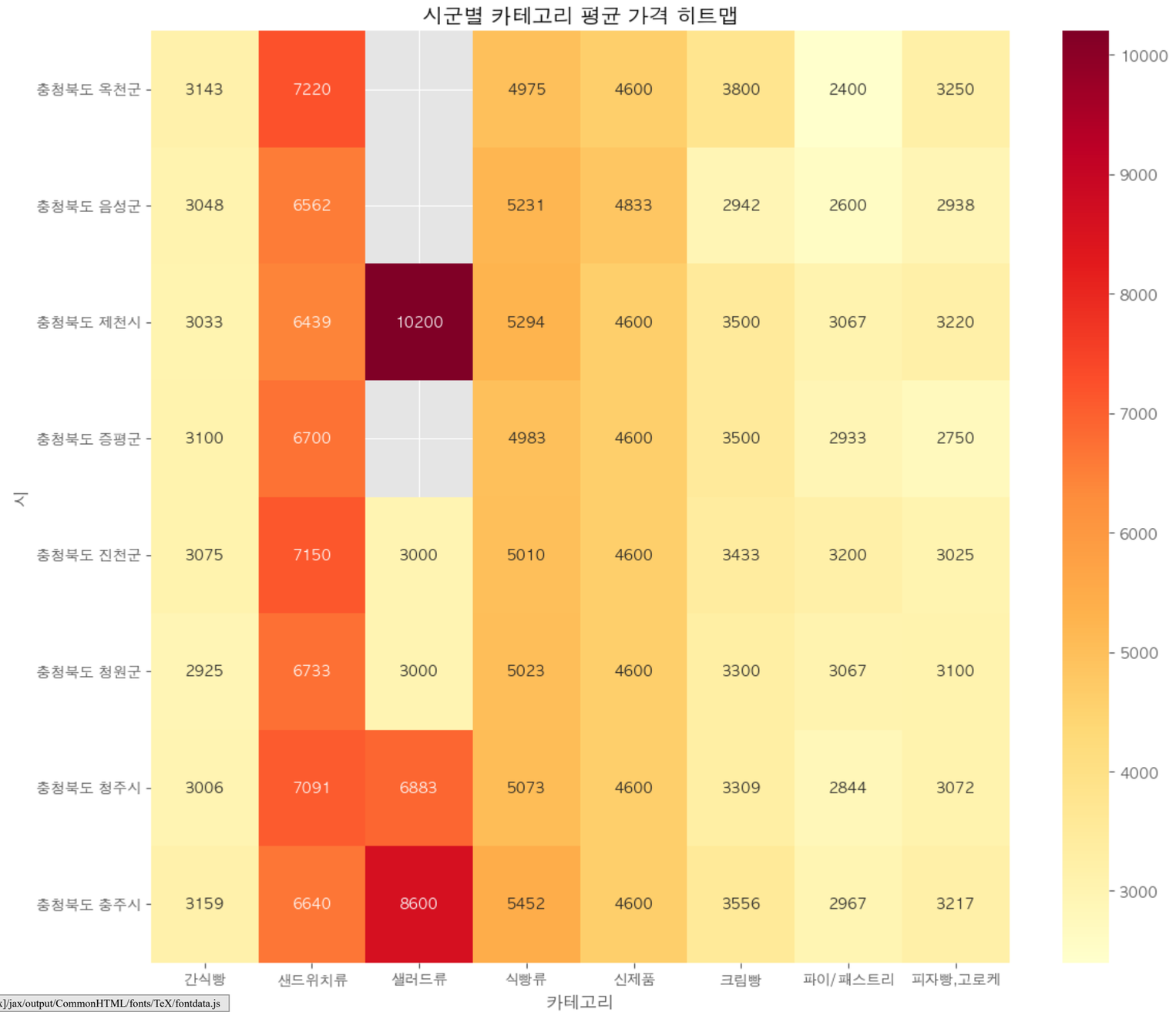


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

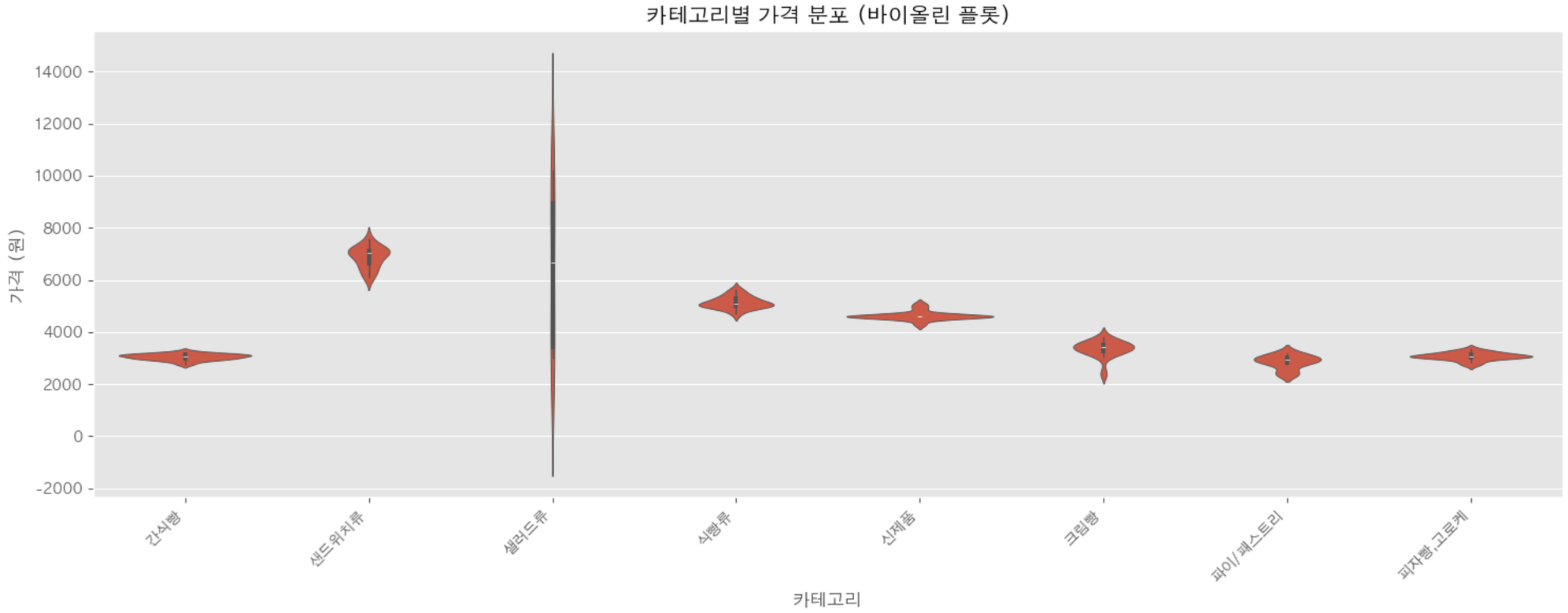


In [15]:
```python
# 3. 구별/카테고리별 평균 가격 히트맵
plt.figure(figsize=(12, 10))
pivot_data = result.groupby('주소')[categories].mean()
sns.heatmap(pivot_data, annot=True, fmt='.0f', cmap='YlOrRd')
plt.title('시군별 카테고리 평균 가격 히트맵')
plt.ylabel('시')
plt.xlabel('카테고리')
plt.tight_layout()
plt.show()

# 5. 카테고리별 가격 분포 (바이올린 플롯)
plt.figure(figsize=(15, 6))
sns.violinplot(x='variable', y='value', data=data_melted)
plt.title('카테고리별 가격 분포 (바이올린 플롯)')
plt.xticks(rotation=45, ha='right')
plt.xlabel('카테고리')
plt.ylabel('가격 (원)')
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [16]:
```python
# 1. 구별 평균 빵가격 계산
categories = ['간식빵', '샌드위치류', '샐러드류', '식빵류', '신제품', '크림빵', '파이/패스트리', '피자빵,고로케']
bread_price_by_district = result.groupby('주소')[categories].mean().mean(axis=1).reset_index()
bread_price_by_district.columns = ['구분', '평균_빵가격']
# '경상북도'제거
bread_price_by_district['구분'] = bread_price_by_district['구분'].str.replace('충청북도', '').str.strip()

# 아파트 가격 데이터 전처리
apt_price = pd.read_csv('./anal_chungbuk/chungbuk_apt_price.csv')
# '충청북도'와 '구' 제거
apt_price['구분'] = apt_price['구분'].str.replace('충청북도', '').str.strip()

apt_price = apt_price.dropna()  # 결측치 제거


# 데이터 확인
print("전처리 후 구별 빵가격 데이터:")
print(bread_price_by_district)
print("\n전처리 후 아파트 가격 데이터:")
print(apt_price)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

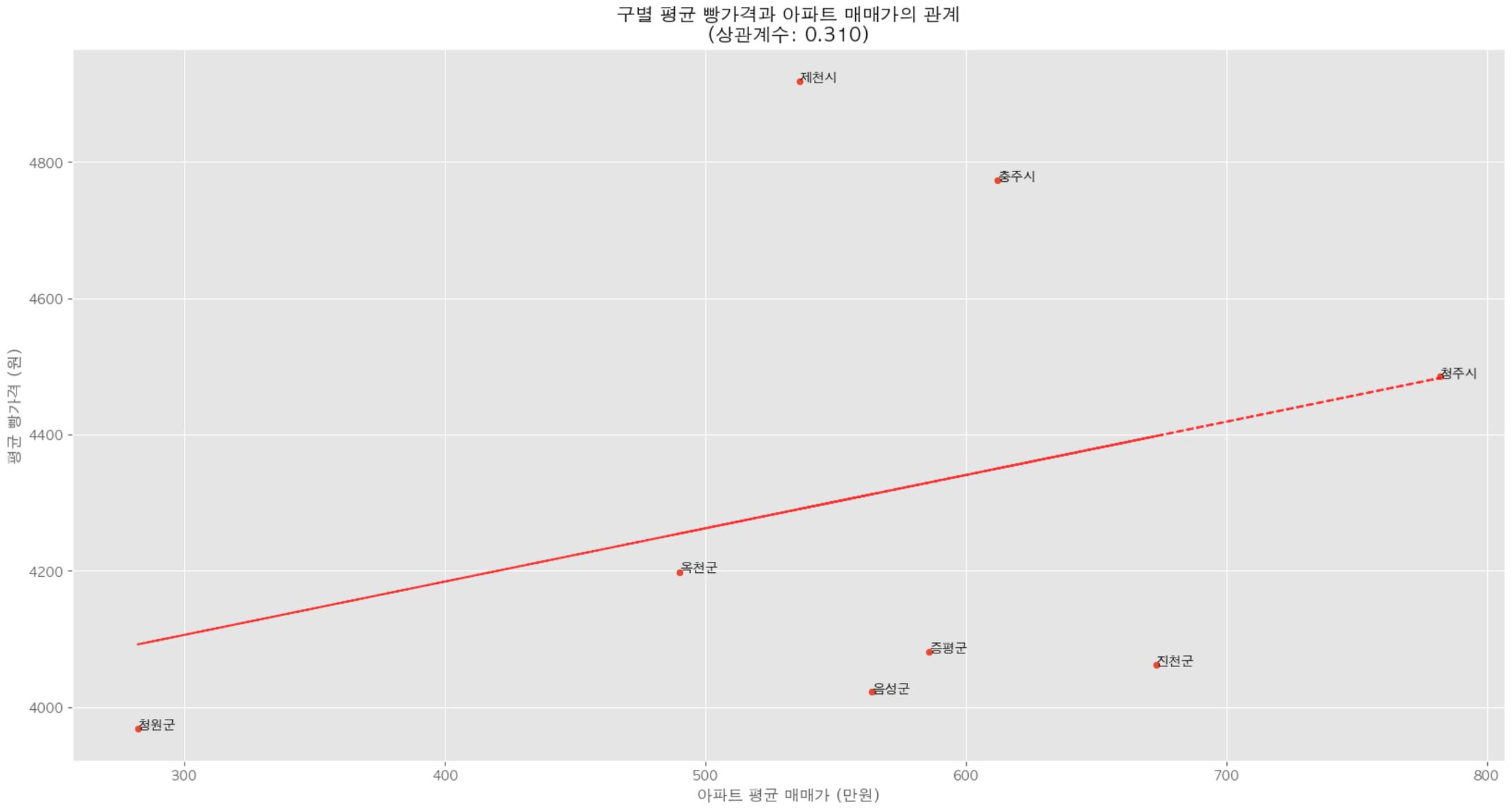
```python
# 데이터 병합
merged_df = pd.merge(bread_price_by_district, apt_price[['구분', '매매']], on='구분', how='inner')
print("\n병합된 데이터:")
print(merged_df)

# 시각화
if not merged_df.empty:
    plt.figure(figsize=(20, 10))
    sns.scatterplot(data=merged_df, x='매매', y='평균_빵가격')

    # 추세선 추가
    x = merged_df['매매'].values
    y = merged_df['평균_빵가격'].values
    z = np.polyfit(x, y, 1)
    p = np.poly1d(z)
    plt.plot(x, p(x), "r--", alpha=0.8)

    # 각 점에 구 이름 표시
    for idx, row in merged_df.iterrows():
        plt.annotate(row['구분'], (row['매매'], row['평균_빵가격']))

    correlation = merged_df['평균_빵가격'].corr(merged_df['매매'])
    plt.title(f'구별 평균 빵가격과 아파트 매매가의 관계\n(상관계수: {correlation:.3f})')
    plt.xlabel('아파트 평균 매매가 (만원)')
    plt.ylabel('평균 빵가격 (원)')

    print(f"\n상관계수: {correlation:.3f}")
    if correlation > 0:
        print("양의 상관관계가 있습니다: 아파트 가격이 높은 구일수록 빵 가격도 높은 경향이 있습니다.")
    else:
        print("음의 상관관계가 있습니다: 아파트 가격이 높은 구일수록 빵 가격이 낮은 경향이 있습니다.")
```

```
전처리 후 구별 빵가격 데이터:
    구분       평균_빵가격
0  옥천군  4198.271429
1  음성군  4021.835714
2  제천시  4919.125000
3  증평군  4080.942857
4  진천군  4061.662500
5  청원군  3968.512500
6  청주시  4484.939583
7  충주시  4773.718750

전처리 후 아파트 가격 데이터:
     구분   매매   전세
0   단양군  476  481
1   보은군  543  532
2   영동군  353  355
3   옥천군  490  490
4   음성군  564  563
5   제천시  536  539
6   증평군  586  587
7   진천군  673  673
8   청원군  282  282
9   청주시  782  782
10  충주시  612  611
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    구분       평균_빵가격   매매
0  옥천군  4198.271429  490
1  음성군  4021.835714  564
2  제천시  4919.125000  536
3  증평군  4080.942857  586
4  진천군  4061.662500  673
5  청원군  3968.512500  282
6  청주시  4484.939583  782
7  충주시  4773.718750  612

상관계수: 0.310
양의 상관관계가 있습니다: 아파트 가격이 높은 구일수록 빵 가격도 높은 경향이 있습니다.
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [17]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

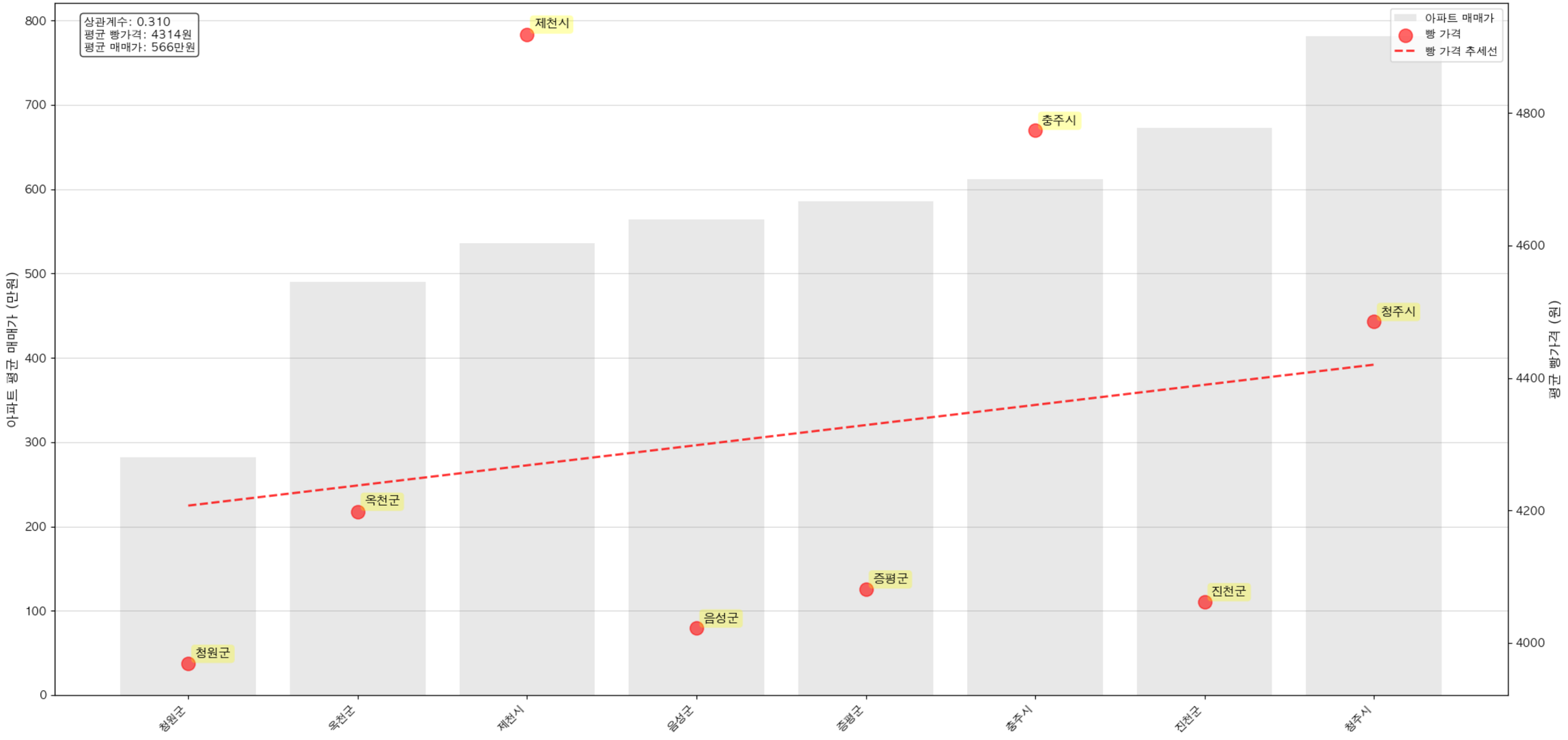
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```python
# 시각화 기본 설정
plt.style.use('default')  # 기본 스타일 사용
plt.rcParams['figure.figsize'] = (20, 10)
plt.rcParams['font.family'] = 'AppleGothic'

if not merged_df.empty:
    # 데이터 정렬
    sorted_df = merged_df.sort_values(by='매매')

    # 메인 그래프 생성
    fig, ax1 = plt.subplots(figsize=(20, 10))

    # 기본 그리드 제거
    ax1.grid(False)

    # 바차트 배경 (매매가)
    ax1.bar(range(len(sorted_df)), sorted_df['매매'], alpha=0.5, color='lightgray', label='아파트 매매가')

    # 산점도를 위한 두 번째 y축 생성
    ax2 = ax1.twinx()
    ax2.grid(False)

    # 산점도 그리기
    scatter = ax2.scatter(range(len(sorted_df)), sorted_df['평균_빵가격'],
                          s=150, alpha=0.6, color='red', label='빵 가격')

    # 추세선 추가
    z = np.polyfit(range(len(sorted_df)), sorted_df['평균_빵가격'], 1)
    p = np.poly1d(z)
    ax2.plot(range(len(sorted_df)), p(range(len(sorted_df))), "r--",
             linewidth=2, alpha=0.8, label='빵 가격 추세선')

    # x축 레이블 설정 (45도 회전)
    ax1.set_xticks(range(len(sorted_df)))
    ax1.set_xticklabels(sorted_df['구분'], rotation=45, ha='right')

    # 각 점에 구 이름 표시
    for i, row in enumerate(sorted_df.itertuples()):
        ax2.annotate(row.구분,
                     (i, row.평균_빵가격),
                     xytext=(6, 6),
                     textcoords='offset points',
                     fontsize=11,
                     bbox=dict(boxstyle='round,pad=0.3', fc='yellow', alpha=0.3, ec='none'))
                     # arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))

    # 평균선 추가
    # ax2.axhline(y=sorted_df['평균_빵가격'].mean(), color='g', linestyle='--', alpha=0.3, label='평균 빵가격')
    # ax1.axhline(y=sorted_df['매매'].mean(), color='b', linestyle='--', alpha=0.3, label='평균 매매가')

    # y축 그리드만 추가
    ax1.grid(True, axis='y', alpha=0.3, linestyle='-', color='gray')

    # 축 레이블 설정
    ax1.set_ylabel('아파트 평균 매매가 (만원)', fontsize=12)
    ax2.set_ylabel('평균 빵가격 (원)', fontsize=12)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```python
# 상관계수 계산
correlation = sorted_df['평균_빵가격'].corr(sorted_df['매매'])

# 그래프 제목 설정
plt.title('충청북도 지역 구별 아파트 매매가와 평균 빵가격의 관계', fontsize=16, pad=20)

# 통계 정보 추가
stats_text = f'상관계수: {correlation:.3f}\n'
stats_text += f'평균 빵가격: {sorted_df["평균_빵가격"].mean():.0f}원\n'
stats_text += f'평균 매매가: {sorted_df["매매"].mean():.0f}만원'
ax1.text(0.02, 0.98, stats_text,
         transform=ax1.transAxes,
         verticalalignment='top',
         bbox=dict(boxstyle='round', facecolor='white', alpha=0.8))

# 범례 추가
lines1, labels1 = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax1.legend(lines1 + lines2, labels1 + labels2, loc='upper right')

# 여백 조정
plt.tight_layout()

# 그래프 표시
plt.show()

# 추가 분석 출력
print("\n=== 지역별 상세 데이터 ===")
analysis_df = merged_df.copy()
analysis_df['가격_차이_순위'] = analysis_df['매매'].rank(ascending=False)
analysis_df['빵가격_순위'] = analysis_df['평균_빵가격'].rank(ascending=False)
analysis_df['순위_차이'] = abs(analysis_df['가격_차이_순위'] - analysis_df['빵가격_순위'])

print("\n아파트 가격 상위 5개 지역:")
print(analysis_df.nlargest(5, '매매')[['구분', '매매', '평균_빵가격']])
print("\n빵 가격 상위 5개 지역:")
print(analysis_df.nlargest(5, '평균_빵가격')[['구분', '매매', '평균_빵가격']])
print("\n순위 차이가 가장 큰 5개 지역 (불일치도가 높은 지역):")
print(analysis_df.nlargest(5, '순위_차이')[['구분', '매매', '평균_빵가격', '순위_차이']])

print(f"\n상관계수: {correlation:.3f}")
if correlation > 0:
    print("양의 상관관계가 있습니다: 아파트 가격이 높은 지역일수록 빵 가격도 높은 경향이 있습니다.")
else:
    print("음의 상관관계가 있습니다: 아파트 가격이 높은 지역일수록 빵 가격이 낮은 경향이 있습니다.")
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js




```
=== 지역별 상세 데이터 ===

아파트 가격 상위 5개 지역:
    구분   매매       평균_빵가격
6  청주시  782  4484.939583
4  진천군  673  4061.662500
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
3  증평군  586  4080.942857
1  음성군  564  4021.835714

빵 가격 상위 5개 지역:
    구분   매매       평균_빵가격
2  제천시  536  4919.125000
7  충주시  612  4773.718750
6  청주시  782  4484.939583
0  옥천군  490  4198.271429
3  증평군  586  4080.942857

순위 차이가 가장 큰 5개 지역 (불일치도가 높은 지역):
    구분   매매       평균_빵가격  순위_차이
2  제천시  536  4919.125000    5.0
4  진천군  673  4061.662500    4.0
0  옥천군  490  4198.271429    3.0
1  음성군  564  4021.835714    2.0
6  청주시  782  4484.939583    2.0

상관계수: 0.310
양의 상관관계가 있습니다: 아파트 가격이 높은 지역일수록 빵 가격도 높은 경향이 있습니다.
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js