

DX컨설턴트 트랙 미니프로젝트 1차

# 조별 발표

DX 10반 23조



# 1. 데이터 처리 및 분석

## 서초구 행정동별 주차장 공급과 불법 주정차 단속 건수 비교

# 여기에 코드를 작성해보세요!

```
seocho_crackdown_count = seocho_crackdown['단속동'].value_counts().reset_index()
seocho_crackdown_count.columns = ['주소_동', '단속건수']

seocho_merged = pd.merge(seocho_crackdown_count, seocho_supply, on='주소_동', how='inner')

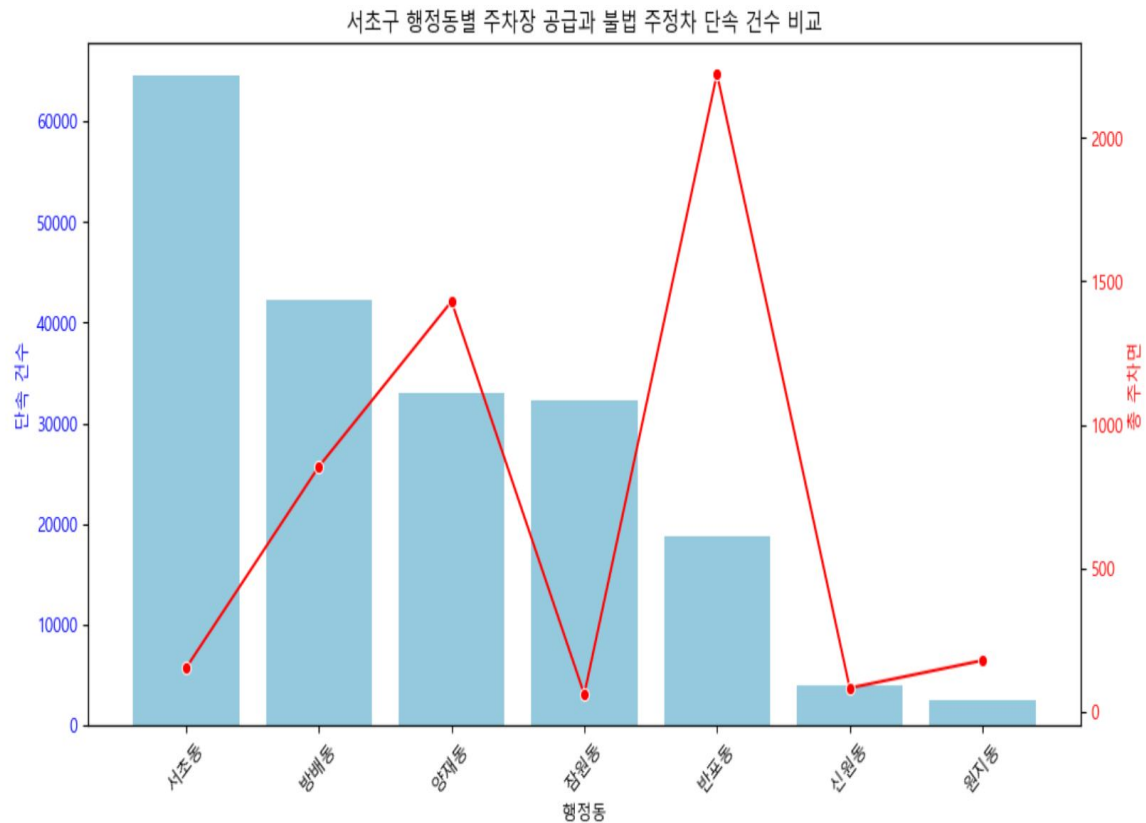
print(seocho_merged.head())

#그래프
fig, ax1 = plt.subplots(figsize=(12,6))

#막대그래프
sns.barplot(data=seocho_merged, x='주소_동', y='단속건수', ax=ax1, color='skyblue')
ax1.set_ylabel('단속 건수', color='blue')
ax1.set_xlabel('행정동')
ax1.tick_params(axis='y', labelcolor='blue')
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=45)

#꺾은선그래프
ax2 = ax1.twinx()
sns.lineplot(data=seocho_merged, x='주소_동', y='총 주차면', ax=ax2, color='red', marker='o')
ax2.set_ylabel('총 주차면', color='red')
ax2.tick_params(axis='y', labelcolor='red')

plt.title('서초구 행정동별 주차장 공급과 불법 주정차 단속 건수 비교')
plt.show()
```



# 1. 데이터 처리 및 분석

## 양천구 행정동별 주차장 공급과 불법 주정차 단속 건수 비교

```
#단속건수 집계
yangcheon_crackdown_count = yangcheon_crackdown['단속동'].value_counts().reset_index()
yangcheon_crackdown_count.columns = ['주소_동', '단속건수']

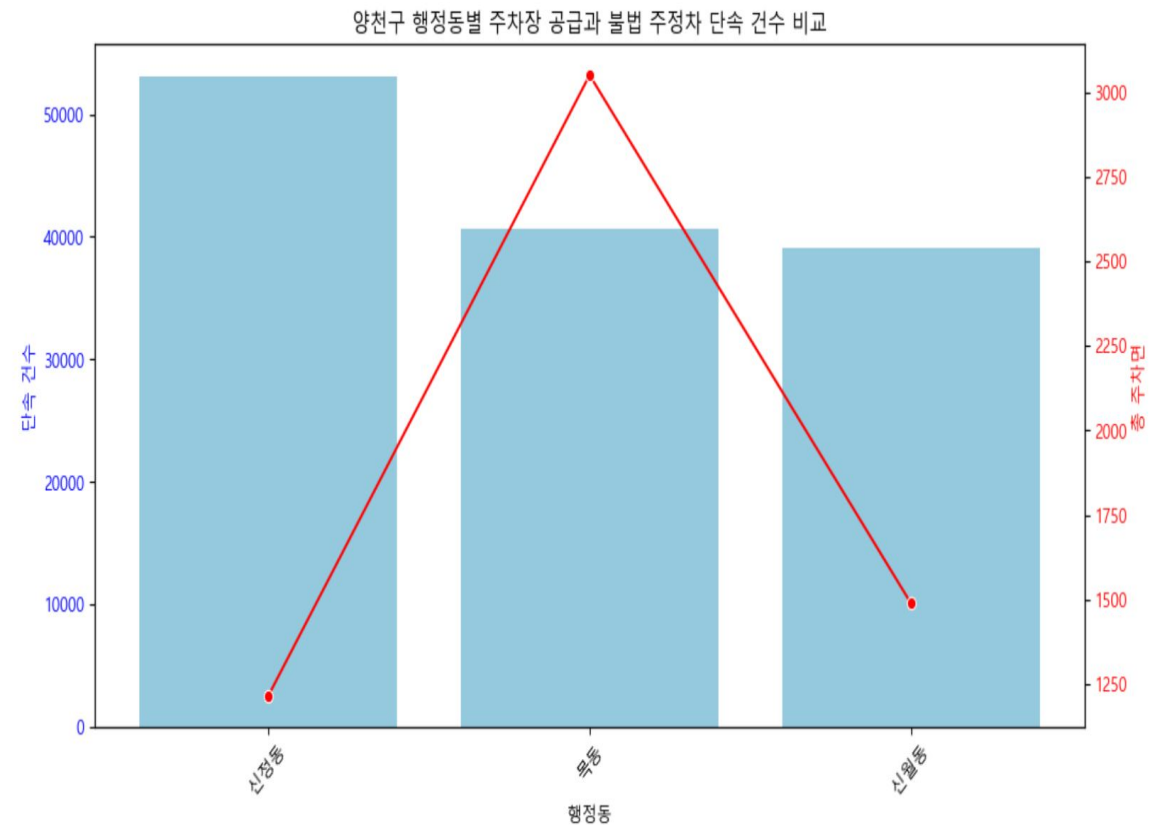
yangcheon_merged = pd.merge(yangcheon_crackdown_count, yangcheon_supply, on='주소_동', how='inner')

#그래프
fig, ax1 = plt.subplots(figsize=(12,6))

#막대그래프
sns.barplot(data=yangcheon_merged, x='주소_동', y='단속건수', ax=ax1, color='skyblue')
ax1.set_ylabel('단속 건수', color='blue')
ax1.set_xlabel('행정동')
ax1.tick_params(axis='y', labelcolor='blue')
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=45)

#꺾은선그래프
ax2 = ax1.twinx()
sns.lineplot(data=yangcheon_merged, x='주소_동', y='총 주차면', ax=ax2, color='red', marker='o')
ax2.set_ylabel('총 주차면', color='red')
ax2.tick_params(axis='y', labelcolor='red')

plt.title('양천구 행정동별 주차장 공급과 불법 주정차 단속 건수 비교')
plt.show()
```



# 1. 데이터 처리 및 분석

## 신정동 산업대분류별 2023년 사업체 수

```
# 신정동 관련 동 이름 리스트
sinjeong_dongs = ['신정1동', '신정2동', '신정3동', '신정4동', '신정6동', '신정7동']

# 신정동 데이터 필터링
sinjeong_biz = biz_info[biz_info['동별'].isin(sinjeong_dongs)]

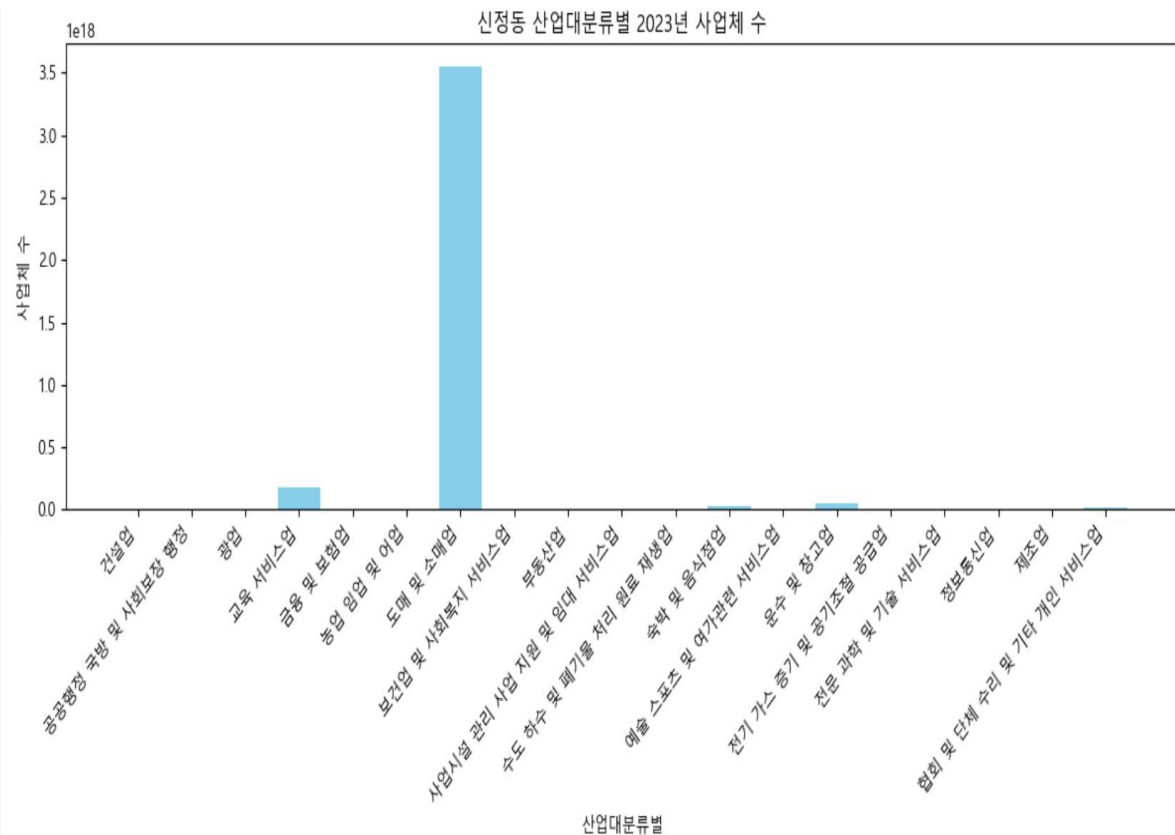
# '사업체수'만 필터링
sinjeong_biz_count = sinjeong_biz[sinjeong_biz['구분별'] == '사업체수']

# 2023년 기준으로 동별 합산
sinjeong_biz_sum = sinjeong_biz_count.groupby('산업대분류별')['2023년'].sum().reset_index()

# '합계' 항목 제외
sinjeong_biz_sum = sinjeong_biz_sum[sinjeong_biz_sum['산업대분류별'] != '합계']

# 사업체 수를 정수형으로 변환 (혹시 - 같은 문자 있을 경우 처리)
sinjeong_biz_sum['2023년'] = pd.to_numeric(sinjeong_biz_sum['2023년'], errors='coerce').fillna(0).astype(int)

# 막대그래프 그리기
plt.figure(figsize=(12,6))
plt.bar(sinjeong_biz_sum['산업대분류별'], sinjeong_biz_sum['2023년'], color='skyblue')
plt.xticks(rotation=45, ha='right')
plt.title('신정동 산업대분류별 2023년 사업체 수')
plt.xlabel('산업대분류별')
plt.ylabel('사업체 수')
plt.tight_layout()
plt.show()
```





# 1. 데이터 처리 및 분석

## 잠원동 산업대분류별 2023년 사업체 수

```
# 잠원동 관련 동 이름 리스트
jamwon_dongs = ['잠원동']

# 잠원동 데이터 필터링
jamwon_biz = biz_info[biz_info['동별'].isin(jamwon_dongs)]

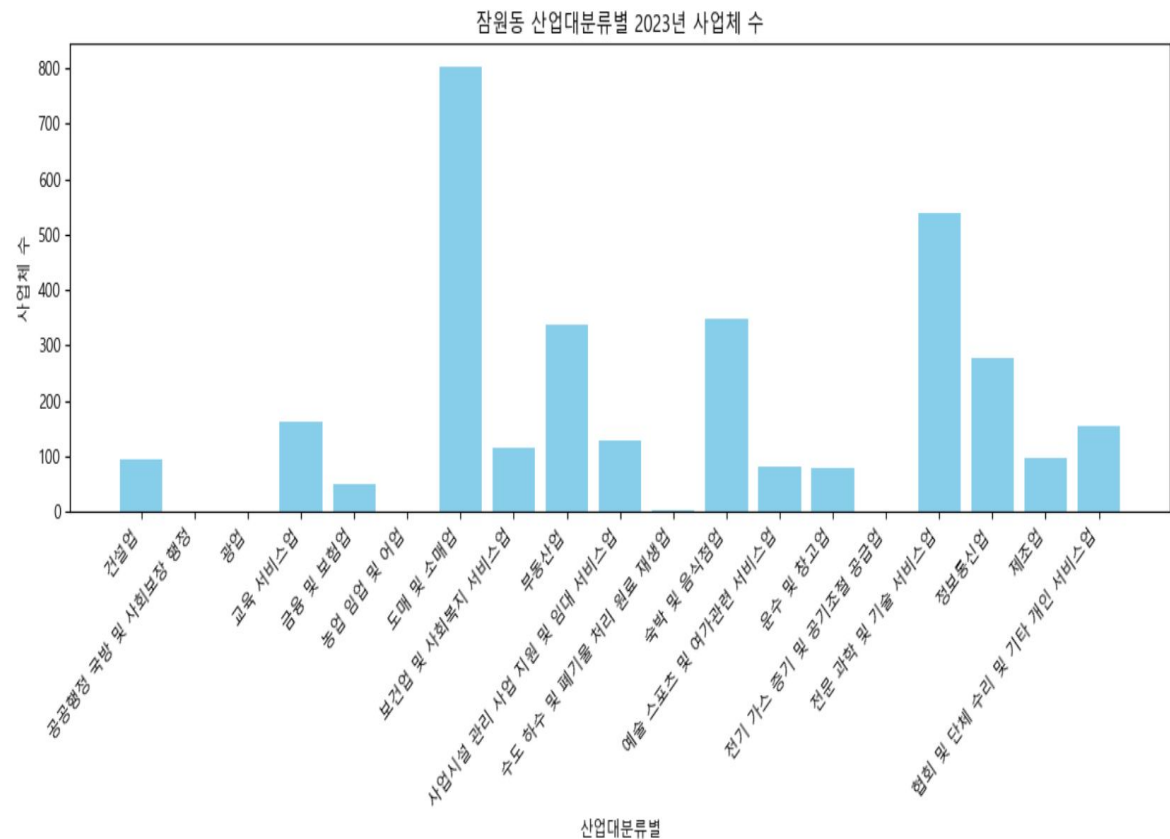
# '사업체수'만 필터링
jamwon_biz_count = jamwon_biz[jamwon_biz['구분별'] == '사업체수']

# 2023년 기준으로 동별 합산
jamwon_biz_sum = jamwon_biz_count.groupby('산업대분류별')['2023년'].sum().reset_index()

# '합계' 항목 제외
jamwon_biz_sum = jamwon_biz_sum[jamwon_biz_sum['산업대분류별'] != '합계']

# 사업체 수를 정수형으로 변환
jamwon_biz_sum['2023년'] = jamwon_biz_sum['2023년'].astype(int)

# 막대그래프 그리기
plt.figure(figsize=(12,6))
plt.bar(jamwon_biz_sum['산업대분류별'], jamwon_biz_sum['2023년'], color='skyblue')
plt.xticks(rotation=45, ha='right')
plt.title('잠원동 산업대분류별 2023년 사업체 수')
plt.xlabel('산업대분류별')
plt.ylabel('사업체 수')
plt.tight_layout()
plt.show()
```



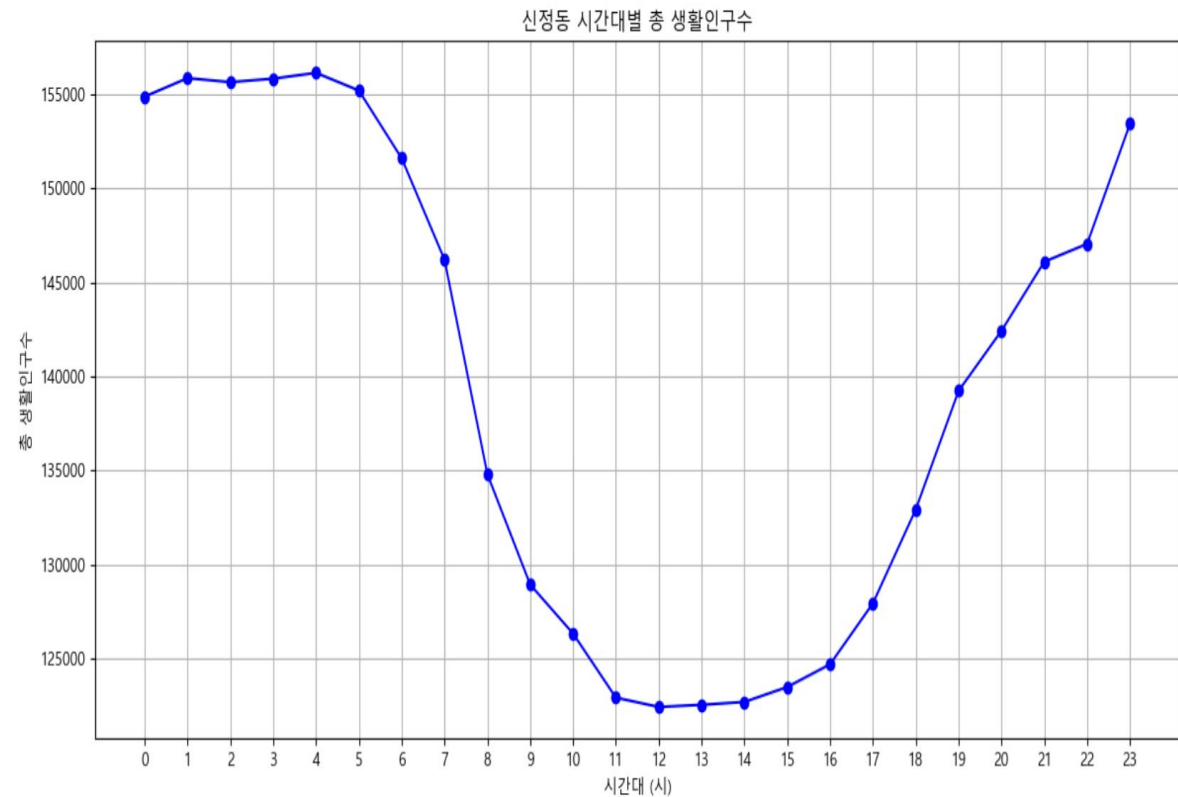
# 1. 데이터 처리 및 분석

## 신정동 시간대별 총 생활인구 수

```
# 신정동 동 이름 리스트
sinjeong_dongs = ['신정1동', '신정2동', '신정3동', '신정4동', '신정6동', '신정7동']

# 신정동 데이터 필터링 후, 시간대별 총생활인구수 합산
sinjeong_time_pop = (
    pop_seoul[pop_seoul['행정동명'].isin(sinjeong_dongs)]
    .groupby('시간대구분')['총생활인구수']
    .sum()
    .reset_index()
)

# 그래프 그리기
plt.figure(figsize=(12,6))
plt.plot(sinjeong_time_pop['시간대구분'], sinjeong_time_pop['총생활인구수'], marker='o', color='blue')
plt.title('신정동 시간대별 총 생활인구수')
plt.xlabel('시간대 (시)')
plt.ylabel('총 생활인구수')
plt.xticks(range(0,24))
plt.grid(True)
plt.tight_layout()
plt.show()
```



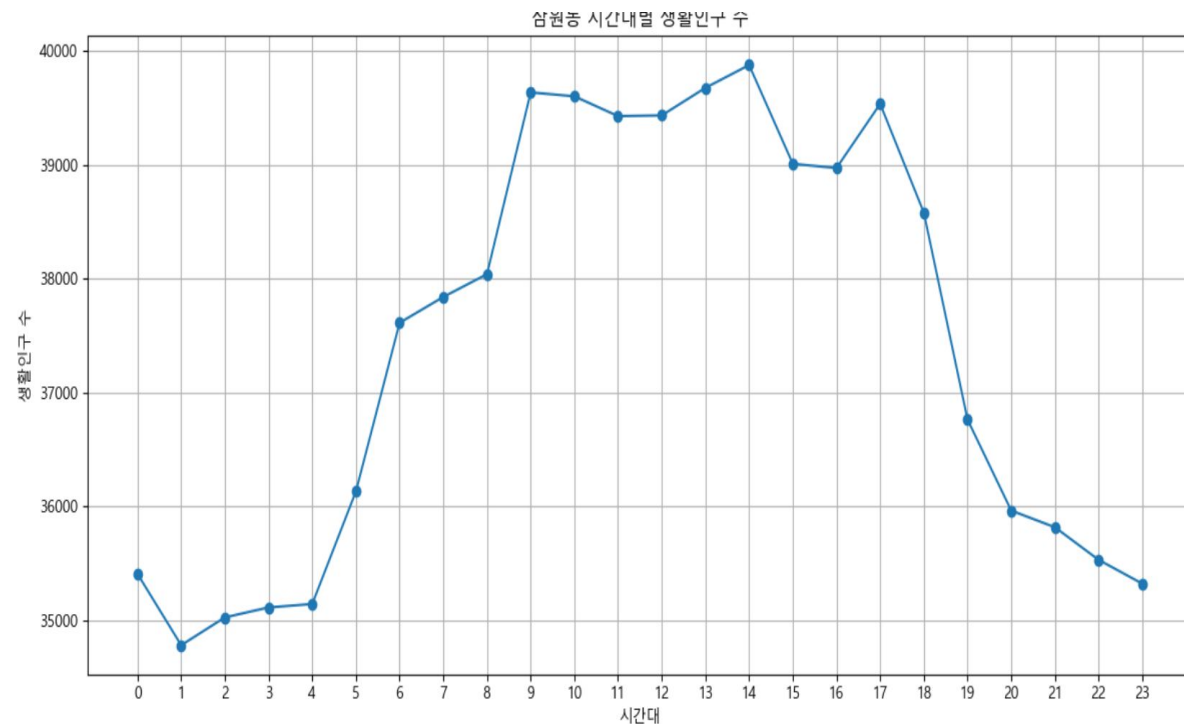
# 1. 데이터 처리 및 분석

## 잠원동 시간대별 생활인구 수

```
# 잠원동 데이터 필터링
jamwon_pop = pop_seoul[pop_seoul['행정동명'] == '잠원동']

# 시간대별 총생활인구수 합계 계산
jamwon_time_pop = jamwon_pop.groupby('시간대구분')['총생활인구수'].sum().reset_index()

# 그래프 그리기
plt.figure(figsize=(12,6))
plt.plot(jamwon_time_pop['시간대구분'], jamwon_time_pop['총생활인구수'], marker='o')
plt.title('잠원동 시간대별 생활인구 수')
plt.xlabel('시간대')
plt.ylabel('생활인구 수')
plt.grid(True)
plt.xticks(jamwon_time_pop['시간대구분'])
plt.tight_layout()
plt.show()
```



# 1. 데이터 처리 및 분석

## 연령대별 생활인구 수 비교 (신정동 vs

```
# 신정동 동 리스트
sinjeong_dongs = ['신정1동', '신정2동', '신정3동', '신정4동', '신정6동', '신정7동']

# 연령대별 컬럼 목록 (남자 + 여자 합산)
age_groups = {
    '0-9세': ['남자0세부터9세생활인구수', '여자0세부터9세생활인구수'],
    '10-14세': ['남자10세부터14세생활인구수', '여자10세부터14세생활인구수'],
    '15-19세': ['남자15세부터19세생활인구수', '여자15세부터19세생활인구수'],
    '20-24세': ['남자20세부터24세생활인구수', '여자20세부터24세생활인구수'],
    '25-29세': ['남자25세부터29세생활인구수', '여자25세부터29세생활인구수'],
    '30-34세': ['남자30세부터34세생활인구수', '여자30세부터34세생활인구수'],
    '35-39세': ['남자35세부터39세생활인구수', '여자35세부터39세생활인구수'],
    '40-44세': ['남자40세부터44세생활인구수', '여자40세부터44세생활인구수'],
    '45-49세': ['남자45세부터49세생활인구수', '여자45세부터49세생활인구수'],
    '50-54세': ['남자50세부터54세생활인구수', '여자50세부터54세생활인구수'],
    '55-59세': ['남자55세부터59세생활인구수', '여자55세부터59세생활인구수'],
    '60-64세': ['남자60세부터64세생활인구수', '여자60세부터64세생활인구수'],
    '65-69세': ['남자65세부터69세생활인구수', '여자65세부터69세생활인구수'],
    '70세이상': ['남자70세이상생활인구수', '여자70세이상생활인구수']
}

# 함수: 특정 동 리스트에서 연령대별 생활인구 합계 계산
def get_agegroup_population(df, dong_list):
    df_filtered = df[df['행정동명'].isin(dong_list)]
    result = {}
    for age_group, cols in age_groups.items():
        # 남/여 컬럼 합산 후 동별, 시간대별 합산
        pop_sum = df_filtered[cols].sum().sum()
        result[age_group] = pop_sum
    return pd.Series(result)

# 신정동 연령대별 생활인구수
sinjeong_agepop = get_agegroup_population(pop_seoul, sinjeong_dongs)

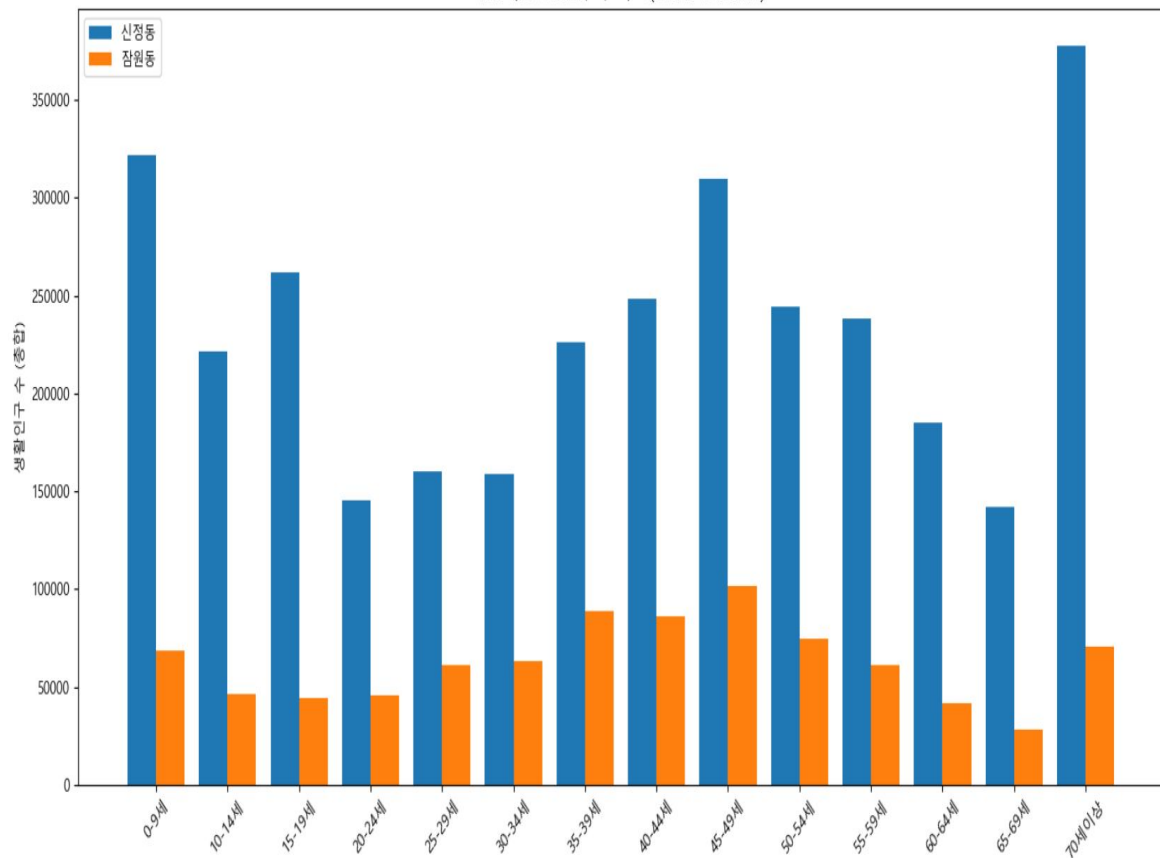
# 잠원동 연령대별 생활인구수
jamwon_agepop = get_agegroup_population(pop_seoul, ['잠원동'])

# 그래프 그리기
age_labels = list(age_groups.keys())
x = range(len(age_labels))

plt.figure(figsize=(14,7))
plt.bar(x, sinjeong_agepop[age_labels], width=0.4, label='신정동', align='center')
plt.bar([i + 0.4 for i in x], jamwon_agepop[age_labels], width=0.4, label='잠원동', align='center')

plt.xticks([i + 0.2 for i in x], age_labels, rotation=45)
plt.ylabel('생활인구 수 (총합)')
plt.title('연령대별 생활인구 수 비교 (신정동 vs 잠원동)')
plt.legend()
plt.tight_layout()
plt.show()
```

연령대별 생활인구 수 비교 (신정동 vs 잠원동)





## 2. 종합 결과

### 문제점 분석

#### 신정동

1. 시간대별 생활인구 그래프에서  
밤 20-24시 사이 인구가 높음 → 야간 주차 수요 존재
2. 주차시설 자료에서 총 주차면 수 낮음

#### 잠원동

1. 시간대별 생활인구 그래프에서  
8시-18시 인구 많음 (업무/상업 중심)
2. 사업체 수 데이터에서 잠원동의 전체 사업체 수  
특히 주간영업 업종이 많음
3. 불법 주정차 단속 데이터에서 출퇴근 시간대에  
집중되어 있다면, 단기 주차 수요 높음 → 회전율 필요

## 2. 종합 결과

### 해결방안

#### 신정동

1. 주거지 전용 야간주차장 운영  
학교 운동장, 공공기관 부설주차장을 야간에만 개방
2. 학원가 / 유흥가 주변 불법주차 단속  
야간집중
3. 인근주차장  
야간 정액제, 주간엔 낮은 요금으로 회전을 확보

#### 잠원동

1. 출퇴근 시간대에 단시간 회전 주차 유도
2. 오전~오후 주차 시간 제한 설정
3. 상가 밀집 구역엔 이중주차 금지구역 지정
4. 인근주차장 → 시간 요금제  
예를 들어 첫 1시간은 무료

