중간고사 대체 과제

가. 형식
다음 [과제안] 5가지 중에 택 1를 해서 분석하고 제출 결과물을 기한 내 제출
[과제안] 난이도가 최종 평가에서 고려됨

나. 주제
[과제안 1] 지역별 공공도서관 현황과 인구 대비 이용 가능성 분석
[과제안 2] 서울시 따릉이 대여/반납 데이터 분석
[과제안 3] 지역별 쓰레기 배출량 및 재활용률 분석
[과제안 4] 전국 전통시장 시설현황 + 시장별 리뷰 크롤링 분석
[과제안 5] 공공도서관 현황 + 도서관 블로그 리뷰 크롤링 분석

다. 제출 결과물
과제 제출 자료 목록 및 세부 항목 (HWP/WORD, 양식 자유)
- 분석보고서
  과제 목표
  데이터 수집 및 전처리 과정
  분석 방법
  분석 결과 및 인사이트
  정책 제안
- 데이터 파일 (CSV)
  최종 병합 결과 파일

라. 제출 기한
2025. 6. 9(월) 23:59 e-Campus 8주차 과제 제출란

마. 평가 지표

평가 항목	배점	평가 기준
데이터 수집 및 전처리 정확성	25점	컬럼 정리, 병합 오류 여부
분석 논리성과 창의성	35점	단순 분석을 넘어선 인사이트 도출 여부
결과물 완성도	20점	시각화, 가독성, 발표 명확성
제출물 형식 준수	20점	파일 제출 형식 및 체계성

[과제안 1] 지역별 공공도서관 현황과 인구 대비 이용 가능성 분석

1. 주제
지역별 공공도서관 수와 인구 대비 접근성 분석
인구수 대비 도서관 수 부족 지역을 파악하고 개선 필요성을 제시하는 것이 목표임

2. 관련 데이터 확보 방법
공공데이터포털(https://www.data.go.kr) → "전국 공공도서관 현황" 데이터 다운로드 (CSV 파일 제공)
행정안전부 주민등록 인구통계(https://jumin.mois.go.kr/) → "시군구별 인구수" 데이터 다운로드

3. 데이터 처리 프로세스(안)

필요한 라이브러리
```
import pandas as pd
import numpy as np
import sqlite3
```

처리 순서
1. 도서관 데이터와 인구 데이터 파일 불러오기
```
library_df = pd.read_csv('library_data.csv')
population_df = pd.read_csv('population_data.csv')
```

2. 시군구 단위로 정리 및 필요한 컬럼만 추출
```
library_grouped = library_df.groupby('시군구명').size().reset_index(name='도서관수')
population_grouped = population_df[['시군구명', '총인구수']]
```

3. 데이터 병합
```
merged_df = pd.merge(library_grouped, population_grouped, on='시군구명')
```

4. 인구 1만 명당 도서관 수 계산
```
merged_df['인구1만명당_도서관수'] = merged_df['도서관수'] / (merged_df['총인구수'] / 10000)
```

5. SQLite 데이터베이스 저장
```
conn = sqlite3.connect('library.db')
merged_df.to_sql('library_info', conn, if_exists='replace', index=False)
conn.close()
```

4. 데이터 처리 주의

시군구 명칭 통일성 확인 (공백, 오타 정리)
연도 기준 통일 (예: 2023년 인구 데이터 기준)
소규모 지역(인구 1만 명 미만)의 극단값(outlier) 처리 주의

5. 추가 고려 사항

단순 개수 비교 외에 고령 인구 비율 대비 도서관 수 분석 가능
면적 대비 도서관 분포 밀도 고려(특히 농촌/도시 차이)
도서관 운영시간 정보가 있다면 접근성(오픈 시간) 분석 추가 가능

6. 예상 결과물 예시

시군구별 인구 대비 도서관 수 상위/하위 10개 지역 표
인구 1만 명당 도서관 수를 지도(heatmap)로 시각화
고령화율 높은 지역 중 도서관 부족 지역 리스트

[과제안 2] 서울시 따릉이 대여/반납 데이터 분석

1. 주제
서울시 공공자전거 따릉이의 대여/반납 패턴 분석
이용이 많은 지역, 시간대, 대여-반납 불균형 지역을 분석하는 것이 목표임.

2. 관련 데이터 확보 방법
서울열린데이터광장(https://data.seoul.go.kr) → "공공자전거 대여소 이용정보" 다운로드 (월별 CSV 제공)

3. 데이터 처리 프로세스(안)

필요한 라이브러리
```
import pandas as pd
import numpy as np
import sqlite3
```

처리 순서
1. 따릉이 대여 데이터 파일 불러오기
```
rental_df = pd.read_csv('rental_data.csv')
```

2. 대여소별 대여건수, 반납건수 집계
```
rental_count = rental_df.groupby('대여소ID')['대여소ID'].count().reset_index(name='대여건수')
return_count = rental_df.groupby('반납대여소ID')['반납대여소ID'].count().reset_index(name='반납건수')
```

3. 대여소별 데이터 병합
```
station_df = pd.merge(rental_count, return_count, left_on='대여소ID', right_on='반납대여소ID', how='outer')
station_df['대여소ID'] = station_df['대여소ID'].fillna(station_df['반납대여소ID'])
station_df = station_df[['대여소ID', '대여건수', '반납건수']].fillna(0)
```

4. 대여소별 불균형 지수 계산 (대여건수 - 반납건수)
```
station_df['불균형지수'] = station_df['대여건수'] - station_df['반납건수']
```

5. SQLite 데이터베이스 저장
```
conn = sqlite3.connect('bike.db')
station_df.to_sql('bike_info', conn, if_exists='replace', index=False)
conn.close()
```

4. 데이터 처리 주의

대여소 ID와 반납소 ID 매칭 오류 주의
날짜 및 시간 포맷 정제(특히 시간대 분석 시 필요)
누락된 대여소 위치정보(Web 크롤링 추가 수집 가능)

5. 추가 고려 사항

시간대별(출근/퇴근/심야) 대여-반납 불균형 분석
대여소별 거리 기반 분석: 가까운 대여소 간 이동 패턴 파악
기상 데이터(비/눈)와 대여건수 비교(기본적인 날씨 영향 검토)

6. 예상 결과물 예시

대여소별 대여/반납 불균형 순위표
시간대별 대여량 변화 그래프 (출근 시간 vs 심야)
불균형 심한 대여소 지도 표시 (heatmap)

[과제안 3] 지역별 쓰레기 배출량 및 재활용률 분석

1. 주제
지역별 생활폐기물 배출량과 재활용률을 분석하여 재활용률이 낮은 지역을 도출하고, 자원순환 정책 개선방향을 제시하는 것이 목표임

2. 관련 데이터 확보 방법
공공데이터포털(https://www.data.go.kr) → "폐기물 배출 및 처리현황" 데이터 다운로드 (환경부 제공, CSV 형식)
행정안전부 주민등록 인구통계(https://jumin.mois.go.kr) → 시군구별 인구 데이터 확보

3. 데이터 처리 프로세스(안)

필요한 라이브러리
```
import pandas as pd
import numpy as np
import sqlite3
```

처리 순서
1. 데이터 불러오기
```
waste_df = pd.read_csv('waste_data.csv')
pop_df = pd.read_csv('population_data.csv')
```

2. 전처리 및 정리
```
waste_df = waste_df[['시군구명', '배출량', '재활용량']]
pop_df = pop_df[['시군구명', '총인구수']]
```

3. 인구 1인당 쓰레기 배출량 및 재활용률 계산
```
waste_df = pd.merge(waste_df, pop_df, on='시군구명')
waste_df['1인당_배출량'] = waste_df['배출량'] / waste_df['총인구수']
waste_df['재활용률'] = waste_df['재활용량'] / waste_df['배출량'] 100
```

4. 데이터베이스 저장
```
conn = sqlite3.connect('waste.db')
waste_df.to_sql('waste_info', conn, if_exists='replace', index=False)
conn.close()
```

4. 데이터 처리 주의
단위 통일(톤, kg 등)

배출량과 재활용량 모두 있는 지역만 분석 포함
기준 연도 일치 필수 (예: 2025년 기준)

5. 추가 고려 사항
재활용률 높은 지역의 특징 도출(정책, 인프라 등)
지역별 1인당 배출량 비교 (도시/농촌 차이)
쓰레기 소각 비율 등 추가 항목 고려 가능

6. 예상 결과물 예시
재활용률 상위/하위 10개 지역 표
1인당 배출량 vs 재활용률 산점도
자원순환 우수/취약 지역 지도 시각화

[과제안 4] 전국 전통시장 시설현황 + 시장별 리뷰 크롤링 분석

1. 주제
전통시장 시설현황과 온라인 리뷰 데이터를 결합하여 시장별 만족도와 시설 개선 필요 요인을 분석을 목표로 함

2. 관련 데이터 확보 방법
공공데이터포털(https://www.data.go.kr) → "전국 전통시장 현황" 데이터 다운로드
네이버 지도 또는 카카오맵 → 시장명 검색 후 리뷰 평점 및 리뷰 수 크롤링 (requests + BeautifulSoup 사용)

3. 데이터 처리 프로세스(안)

필요한 라이브러리
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
import sqlite3
```

처리 순서
1. 시장 정보 로딩
```
market_df = pd.read_csv('market_data.csv')
```

2. 크롤링 함수 작성
```
def get_review_data(market_name):
    URL 수정 필요 (예시)
    url = f"https://map.naver.com/v5/search/{market_name}"
    response = requests.get(url, headers={'User-Agent': 'Mozilla/5.0'})
    soup = BeautifulSoup(response.text, 'html.parser')
    rating, reviews = None, None
    예시 선택자
    try:
        rating = soup.select_one('.rating_class').text
        reviews = soup.select_one('.review_count_class').text
    except:
        pass
    return rating, reviews
```

3. 리뷰 데이터 병합
```
market_df[['평점', '리뷰수']] = market_df['시장명'].apply(lambda x:
```

pd.Series(get_review_data(x)))

4. 저장
conn = sqlite3.connect('market_review.db')
market_df.to_sql('market_info', conn, if_exists='replace', index=False)
conn.close()

4. 데이터 처리 주의
시장명 표준화 (서울남대문시장 vs 남대문시장)
크롤링 실패 처리 (리뷰 없는 시장 예외 처리)
요청 차단 회피 (time.sleep 사용 등)

5. 추가 고려 사항
시장 규모(점포 수)와 리뷰 평점 비교
키워드 분석: 청결, 가격, 친절 등 단어 빈도
수도권/지방 시장별 만족도 차이

6. 예상 결과물 예시
평점 상위 시장 순위표
시장 규모 대비 리뷰 평점 비교
키워드 분석 워드클라우드

[과제안 5] 공공도서관 현황 + 도서관 블로그 리뷰 크롤링 분석

1. 주제
전국 공공도서관 정보와 블로그 리뷰를 결합하여 도서관별 서비스 만족도와 개선 요소를 분석을 목표로 함

2. 관련 데이터 확보 방법
공공데이터포털(https://www.data.go.kr) → "전국 공공도서관 운영현황" 다운로드
네이버 블로그 검색 → 시장명 기반 블로그 제목/요약 크롤링 (검색 결과 파싱)

3. 데이터 처리 프로세스(안)

필요한 라이브러리
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

처리 순서
1. 도서관 데이터 불러오기
```
library_df = pd.read_csv('library_info.csv')
```

2. 블로그 리뷰 수집 함수 (네이버 검색 기반)
```
def get_blog_reviews(library_name):
    url = f"https://search.naver.com/search.naver?query={library_name}+도서관"
    headers = {'User-Agent': 'Mozilla/5.0'}
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, 'html.parser')
    제목 추출 예시
    titles = [t.text for t in soup.select('.blog .title_class')][:5]
    return titles
```

3. 병합
```
library_df['리뷰제목'] = library_df['도서관명'].apply(lambda x: get_blog_reviews(x))
```

4. 데이터 처리 주의
도서관명 정확도(검색 실패 시 예외 처리)
블로그 리뷰 신뢰성(상업글 필터링 불가능성)
크롤링 지연 조치(time.sleep)

5. 추가 고려 사항

운영시간 길이 vs 리뷰 만족도 상관분석
리뷰 텍스트 내 키워드 빈도(예: "조용함", "열람실")
최근 신축 도서관 vs 노후 도서관 비교

6. 예상 결과물 예시

블로그 키워드 기반 워드클라우드
도서관 평판이 좋은 지역 분포 지도
시설면적/운영시간 등과의 비교표