**코드 및 알고리즘 상세 설명 보고서**

팀명: SNU GIS

리더: 석사과정2

팀원: 석사과정1, 석사1

1. **사용 환경 및 라이브러리 버전**

* Python: 3.6
* 라이브러리
  + dash==1.19.0
  + esda==2.3.5
  + Fiona==1.8.17
  + folium==0.11.0
  + GDAL==3.1.4
  + geojson==2.5.0
  + geopandas==0.8.1
  + libpysal==4.3.5
  + mapclassify==2.4.2
  + matplotlib==3.3.2
  + osmnx==1.0.1
  + plotly==4.14.1
  + pysal==2.3.0
  + Rtree==0.9.4
  + scikit-learn==0.23.2
  + seaborn==0.11.1
  + Shapely==1.7.1
  + statsmodels==0.12.1
  + sympy==1.7.1
  + tqdm==4.51.0

1. **데이터 전처리**
   1. 그리드 통합, 주소 부여

* 분석 설명: 제공받은 데이터 프레임들을 불러오고 자료들 간 사이즈 불일치, 주소 부여 등 데이터 통합을 위한 블록, 전처리 된 데이터를 로컬 파일에 저장
* 코드 설명:
  + 사용 데이터: 3, 11, 12, 16, 17번 자료
  + 아웃풋: 데이터 프레임 딕셔너리(1~17번, 통합 그리드)
  + 함수 별 설명
  + write\_data, load\_data: pickle 라이브러리를 이용한 데이터 저장, 불러오기 함수
  + split\_18: 코드 정의서인 18번자료의 각 테이블을 개별 dataframe으로 잘라주는 함수
  + get\_data: 1~17번 자료를 한꺼번에 불러와 f\_dict라는 딕셔너리에 저장
  + grid\_trim: 16번자료와 3~6번자료를 통합하는 함수, 사이즈가 다른 두 그리드를 500x500 사이즈로 변환함
    - 가나 : 한글 인덱스, 123: 숫자 인덱스, ab: 알파벳 인덱스
    - df1: 격자 개체의 크기가 100 x 100인 데이터 프레임, gid는 가나123456의 구조로 돼있으며 가는 '다'로 고정, 나는 '사' or '바'를 가짐 가, 나는 df2와 동일
    - 12는 가로축 고유 번호로 서쪽으로 갈 수록 값이 오르면 45는 세로축 번호로 남쪽으로 갈 수록 값이 오름. 3과 6은 0~9의 값을 가지며 서쪽, 남쪽으로 갈수록 큰 값을 가짐
    - df2: 격자 개체의 크기가 250 x 250인 데이터 프레임, gid는 가나12ab45cd의 구조로 돼있으며 가나, 12, 45는 df1과 같고 ab와 cd는 이진(a와 b로만 이루어짐) 인덱스로
    - ab는 가로 4개의 인덱스, cd는 세로 4개의 인덱스를 표현함. 즉 16개의 격자가 한 쌍으로 구성됨 (2진법의 네자리 경우의 수).
    - df1은 3~6번 데이터에 해당되며 df2는 16번 데이터임 df1의 격자 25개와 df2의 격자 16개가 겹치는 패턴을 보이기에 두 데이터 프레임을 합칠 수 있음
  + assign\_address: 11, 12번 자료를 사용해 주소 부여
  1. 분석에 필요한 데이터 프레임 생성
* 분석 설명: 기본 데이터 프레임들에서 차후 분석에 필요한 데이터 프레임 생성
* 코드 설명:
  + 사용 데이터: 1, 2번 자료, 그리드 자료
  + 아웃풋: 상점 개수, 업종 대분류, 중분류의 각 코드가 맵핑된 데이터 프레임
  + 함수 별 설명: 함수 미 사용

1. **그리드 데이터를 활용한 분석 시행**
   1. 정규성 검사

* 분석 설명: D'Agostino and Pearson's Test, Shapiro-Wilk Test, Anderson-Darling Test 3가지 정규성 test 실행
* 코드 설명:
  + 사용 데이터: 연속형 데이터
  + 아웃풋: 정규성 여부 판단
  + 함수 별 설명:
    - normality\_test: 데이터 프레임 내 관심 컬럼(col)의 정규성 여부를 판단
  1. 단순 분석
* 분석 설명: 그리드 자료에 포함된 여러 데이터를 사용하여 단순한 분석 시행
* 코드 설명:
  + 사용 데이터: 그리드 자료
  + 아웃풋: 다양한 그리드 지도 (주요 창업 연령대, 청년 창업비율, 젊은 법정동, 사업체 평균 나이, 총 인구수, 총 사업체 수 타일별/법정동 별 사업체 당 인구수, 채무 매출 비율, 재무 안정도 (LSR/평균 사업체 나이))
  + 함수 별 설명
    - simple\_grid\_calc: 다양한 mode에 따라 간단한 column-wise 연산을 시행하고 반환하는 함수
    - plot\_grid\_map: 데이터 프레임 내 관심 컬럼 (COI)를 기준으로 분류하여 지도 생성, 저장
  1. 핫스팟 분석
* 분석 설명: local moran’s I 통계량을 이용한 hot spot 식별
* 코드 설명:
  + 사용 데이터: 그리드 데이터, 그리드 별 분석하고자 하는 데이터
  + 아웃풋: 그리드 별 hot spot, cold spot 여부
  + 함수 별 설명:
    - lps.weights.Queen.from\_dataframe(df): 공간행렬을 Queen 방식으로 구축
    - li = esda.Moran\_Local(c, wq)

sig = 1 \* (li.p\_sim < 0.05) : local moran 통계량이 유의한 지역 판단

* 1. 최근린 분석
* 분석 설명:
  + 그리드 별 상점들의 최근접 이웃과의 거리의 평균 분석을 통해 상점간의 물리적 경쟁을 정량화해서 확인
  + 최근린 거리가 작을수록 거리 기반의 경쟁이 치열함을 의미
  + 최근린 거리의 역수를 등급화하여 그리드 별 거리기반의 경쟁 파악을 용이하게함
* 코드 설명:
  + 사용 데이터: 1번 자료, 그리드 자료
  + 아웃풋: 그리드 별 거리기반의 경쟁을 나타내는 class
  + 함수 별 설명:
    - PointPattern([(j.x,j.y) for j in shop['geometry']]).nnd : 상점 별 최근린 거리 계산
  1. 유동인구 분석
* 분석 설명:
  + 유동인구의 효율적인 분석을 위해 시간을 집계해 0-6시 사이를 새벽, 7-12시 사이를 아침, 13시-18시를 오후, 19시-24시를 저녁시간으로 나눔
  + grid 별 새벽, 아침, 오후, 저녁시간의 유동인구를 단계구분도로 시각화하고 local moran's I로 핫스팟 탐지
* 코드 설명:
  + 사용 데이터: 그리드 자료, 유동인구 자료
  + 아웃풋: 단계구분도, 핫스팟 탐지
  + 함수 별 설명:
    - ydpop\_plot: 새벽, 아침, 점심, 저녁 별 평균 유동인구의 단계 구분도, hot spot plot
  1. 연령대별 카드 사용 비율
* 분석 설명:
  + 통계청 추가 데이터를 활용해 가구주 연령별 가구당 월평균 가계지출 항목 비율 분석
  + 용인시 내 각 그리드의 카드 사용 비율이 가장 높은 연령대 산출 및 비율 조사
* 코드 설명:
  + 사용 데이터: 그리드 데이터, 통계청 추가 데이터
  + 아웃풋: 파이차트, 그리드 단계 구분도
  + 함수 별 설명:
    - make\_pie: 통계청 자료를 가구주 연령대 별로 분할한 데이터프레임을 입력으로 받고, 항목 별 지출 비율에 대한 파이차트 생성
    - age\_choropleth: 용인시 각 그리드의 카드 사용 비율이 가장 높은 연령대와 그 비율(수) { 그리드id: 연령대, 비율 }로 구성된 데이터프레임을 입력으로 받은 후, 연령대를 대상으로 단계 구분도 생성

1. **정량 분석** 
   1. 소상공인, 20대 창업, 업종별 상관관계 분석

* 분석 설명:
* 코드 설명:
  + 사용 데이터: 전처리 단계에서 구축한 업종 코드 별 맵핑 데이터
  + 아웃풋:
    - 소상공인 수와 상관관계가 높은 대분류 업종 코드 scatter plot
    - 소상공인 수, 20대 창업 수와 상관관계가 높은 대분류 업종 코드 scatter plot
    - 소상공인 수와 상관관계가 높은 중분류 업종 코드 scatter plot
    - 소상공인 수, 20대 창업 수와 상관관계가 높은 중분류 업종 코드 scatter plot
  + 함수 별 설명:
    - plot\_2D\_scatters: 데이터 프레임 내 두 column의 강한 상관관계 (R^2 > 0.7)에 대한 점산도와 방향성 출력
    - plot\_3D\_scatters: 데이터 프레임 내 세 column의 강한 상관관계 (R^2 > 0.7)에 대한 점산도와 평면 출력
    - get\_sales\_trend: 데이터 프레임의 매출 데이터를 사용해 이동 평균 값을 구하고 전체 평균과 비교해 추세값을 구함
  1. 행정구역 기반 분기별 매출 추이
* 분석 설명: 용인시 지역구 (3개)에 대한 법정동 별 분기 매출 추이 분석
  + 17년 1분기~20년 3분기의 그리드 별 매출액을 지역구 법정동 별로 요약하여 라인 그래프 생성
* 코드 설명:
  + 사용 데이터: 그리드 데이터, 16번 데이터, 17번 데이터
  + 아웃풋: 법정동 별 분기 매출 추이를 나타내는 라인 그래프 (interactive dash plotly 그래프)
  + 함수 별 설명: 함수 미사용
  1. 정책 제안을 위한 지수 도출
* 분석 설명: 그리드 별 세부 정책제안을 하기 위해 클래스를 도출하는 데 사용되는 3가지 지수 도출 코드
  + 1. 금융 지수: 평균 매출액 / 가중 평균 대출액 \* 매출 트랜드 값
  + 2. 경쟁 지수: 총 상점 수 \* 평균 최근린 거리 \* Hotspot 여부
  + 3. 입지 지수: (총 인구 수 \* 총 인구 Hot spot) + (유동인구 Hot spot \* 유동인구)
* 코드 설명:
  + 사용 데이터
    - 금융지수: 16번 데이터, 그리드 데이터
  + 아웃풋: 3가지 지수에 따른 그리드 지도, 3차원 scatter plot, 최종 그리드 클래스별 지도
  + 함수 별 설명: simple\_grid\_calc, get\_sales\_trend
    - 금융지수에서 쓰인 함수:
      * simple\_grid\_calc: 다양한 mode에 따라 간단한 column-wise 연산을 시행하고 반환하는 함수
      * get\_sales\_trend: 데이터 프레임의 매출 데이터를 사용해 이동 평균 값을 구하고 전체 평균과 비교해 추세값을 구함
  1. 지수를 사용한 그리드 별 정책 제안
* 분석 설명: 상기 분석을 바탕으로 도출된 지수들을 통해 클래스를 나누고 시각화하는 파트
* 코드 설명:
  + 사용 데이터: 3대 지수와 그리드 데이터
  + 아웃풋: 금융, 경쟁, 입지지수 Scatter plot, 그리드 별 최종 분류
  + 함수 별 설명: 함수 미사용

1. **관심 지역 집중 분석**
   1. 타 지역 SK하이닉스 반도체산업단지와 유사 국가/기업주도 산업단지 인근의 상권 업종 중분류명 별 소상공인 사업체 개수 및 비율

* 분석 설명:
* 코드 설명:
  + 사용 데이터: 소상공인시장진흥공단 출처의 경기, 충남 상권정보
  + 아웃풋:
    - 개발 대상 지역의 상권 현황 비율 누적 막대 그래프
    - 유사 산업 단지의 사업체 개수 막대 그래프 (3개)
    - 각 산업 단지의 업종별 사업체 비율 누적 막대 그래프
  + 함수 별 설명:
    - make\_overtime\_bargraph: 소상공인진흥공단의 추가 데이터를 활용하여 유사 산업 단지(현대 아산, 이천 SK하이닉스, 포승공단)내 업종 별 사업체 개수를 요약한 데이터프레임을 입력으로 받고 각 지역에 대한 사업체 개수 막대그래프 생성
    - horizontal\_stacked\_bar1: 상기 함수에 입력으로 주었던 데이터프레임의 사업체 개수를 비율로 변환한 데이터프레임을 입력으로 받고 누적 막대 그래프를 생성
  1. 용인시 SK하이닉스 반도체산업단지와 기흥 플랫폼시티 인근의 상권 업종 중분류명 별 소상공인 사업체 비율
* 분석 설명:
* 코드 설명:
  + 사용 데이터: 12번 데이터, 1번 데이터, 2번 데이터
  + 아웃풋: 플랫폼시티와 SK하이닉스 반도체 클러스터 개발 예정 지역에 대한 업종별 사업체 비율 누적 막대 그래프
  + 함수 별 설명:
    - horizontal\_stacked\_bar2: 1번 데이터를 이용해 개발 예정 지역에 속하는 사업체들만 추출하여 만든 데이터프레임을 입력으로 받고, 총 4개의 지역에 대한 사업체의 비율을 누적막대그래프로 생성