3조 Presentation

INDEX

Part 1,

개요

Part 1,

주제 선정

설사:
150만 명 사망
(연간)
88% 는 어린이

영양실조:
86만 명 사망
(연간)
대부분 5세 미만 어린이

장 기생 선충 감염:
20억 명 감염
세계 인구의 1/3

주혈 흡충증:
2억 명
예방할 수 있는 감염

A

WHO자료에 따르면 매년 300만명 이상의 사람들이 수질과 관련되어 사망한다.

B

대한민국에서는 수질이 건강에 어떠한 영향을 미칠까?

C

수질과 다양한 질병들의 지역별 데이터를 통해 조사한다.

Part 1,

진행 과정

주제 설정

상수도 수질이
지역사회 건강에
미치는 영향

데이터 전처리

수질 데이터 전처리
건강 데이터 전처리

데이터 분석

지역별 건강 지표 분석

지역별 수질 지표 분석

건강 결정 요인과
수질 지표의
상관관계 분석

시각화 및 웹 서버

산점도, 막대그래프
를 통한 시각화

웹 서버 구현

웹 시스템 구축

Part 1,

역할

이재웅

팀장

- 전체 과정 총괄
- 수질 데이터 전처리
- 회귀분석 및 시각화
- 웹 서버 구현
- 기획안, PPT 작성

김병현

팀원

- 건강결정요인 데이터 전처리
- 건강결정요인 분석 및 시각화
- 웹 서버 구현
- PPT 작성

유준웅

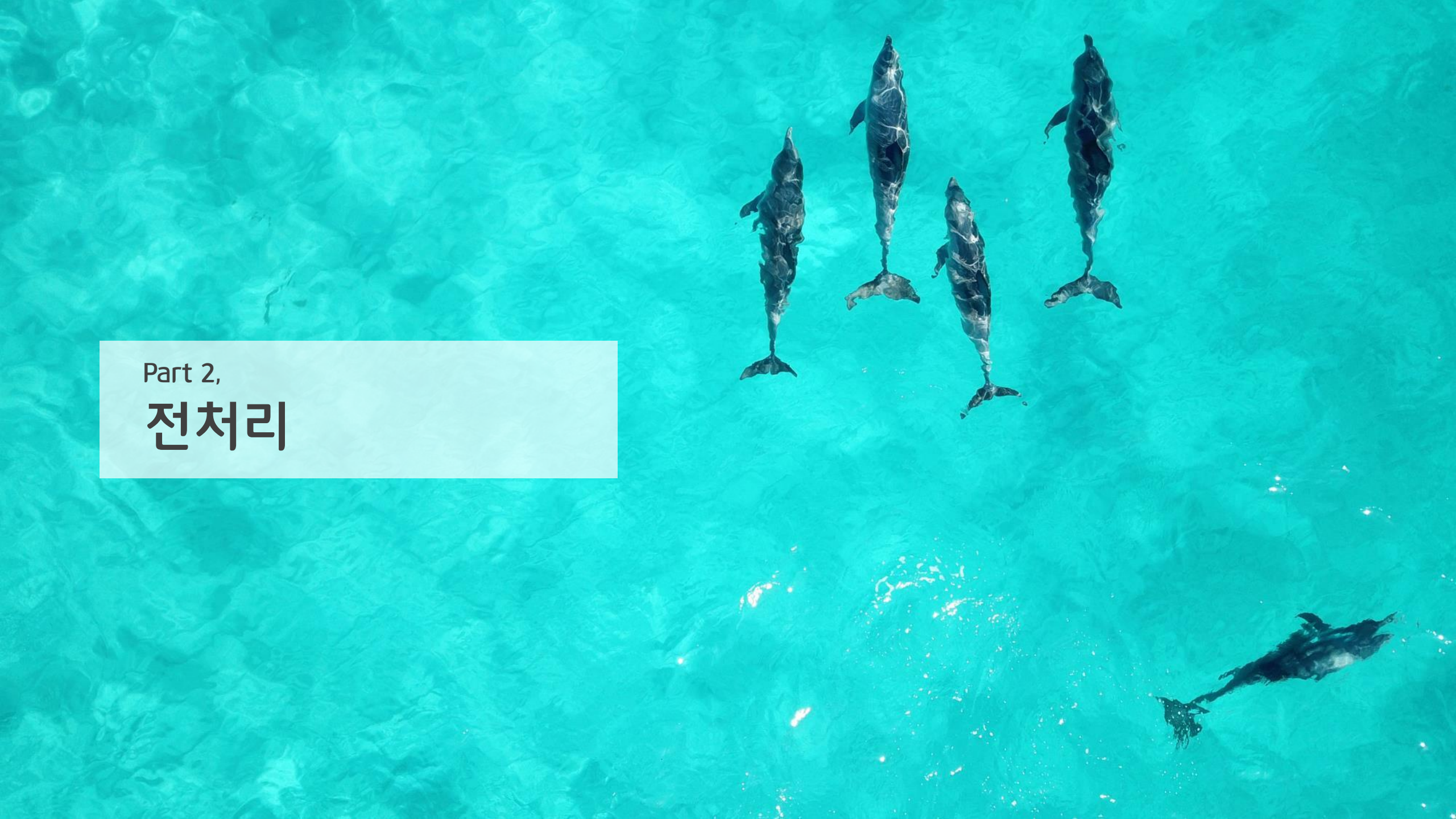
팀원

- 수질 데이터 전처리
- 수질 데이터 분석 및 시각화
- 웹 서버 구현
- 웹 호스팅

개발 환경 및 도구

개발 환경 및 도구

언어	Python
통합 개발 환경	Pycharm, Jupyter Notebook
분석 라이브러리	Pandas, Numpy, Sikit-Learn
시각화 라이브러리	Highcharts, Matplotlib, Seaborn
웹 서버	Django
웹 클라이언트	HTML5, JS, Ajax, CSS
시스템 운영 환경	Python Anywhere
협업 도구	Github, Zoom, Google Docs



Part 2,

전처리

Part 2,

전처리

```
class VS:
    def healthPreprocessing(self):
        # 2009 ~ 2018년 건강 데이터
        dfh_all = pd.read_excel(DATA_DIRS[0] + '//health_2009_2018_preprocessed.xlsx', sheet_name = None, engine = 'openpyxl');
        dfh = pd.concat(dfh_all, ignore_index=True);

        # 분석 대상 지역 데이터 추출
        area = dfh['지역'].isin(['서울특별시','부산광역시','대구광역시','인천광역시','광주광역시','대전광역시','울산광역시','경기도',
                                 '강원도','충청북도','충청남도','전라북도','전라남도','경상북도','경상남도','제주특별자치도'])
        dfh = dfh[area]

        # 분석 대상 건강지표 컬럼 추출
        health = dfh[['시도', '스트레스인지율_표준화율', '우울감경험률_표준화율', '주관적구강건강이나쁜인구의분율_표준화율',
                      '스트레스로인한정신상담률_표준화율']]
        health = health.set_index('시도')

        # 건강 데이터 전처리 데이터프레임: health
        return health
```

건강 데이터 전처리

Part 2,

전처리

```
for x in range(11):
    a = alist[x][0]
    b = alist[x][1]
    c = alist[x][2]
    d = alist[x][3]
    e = alist[x][4]
    f = alist[x][5]
    g = alist[x][6]
    h = alist[x][7]
    i = alist[x][8]
    j = alist[x][9]
    k = alist[x][10]
    l = alist[x][11]
    m = alist[x][12]
    n = alist[x][13]
    o = alist[x][14]
    p = alist[x][15]
    q = alist[x][16]
    dic1 = {'서울특별시':a, '부산광역시':b, '대구광역시':c, '인천광역시':d, '광주광역시':e,'대전광역시':f,'울산광역시':g, '세종특별자치
    }
    dic = {}

    #for k, v in dic1.items():
        #dic[v] = k

    s_dic= sorted(dic1.items(), key=operator.itemgetter(1), reverse = True)
    a = list(s_dic)
    print(s_dic, '₩n')
    for i in range(17):
        print('"'+s_dic[i][0]+'",', end=' ')
    print('₩n')
    for i in range(17):
        print(s_dic[i][1],',', end=' ')
    print('₩n')
```

건강 데이터 전처리 - 데이터를 내림차순으로 변경한다.

Part 2,

전처리

```
def waterPreprocessing(self):
    # 2009 ~ 2018년 수질 데이터
    dfw_all = pd.read_excel(DATA_DIRS[0] + '//water_2009_2018_preprocessed.xlsx', sheet_name = None, engine='openpyxl');
    dfw = pd.concat(dfw_all, ignore_index=True);

    # 필요없는 컬럼 drop
    dfw.drop(['시설명', '소재지', '수원', '채수년월일'], axis=1, inplace=True)

    # '납(기준:0.05/ 단위:(mg/L))', '비소(기준:0.05/ 단위:(mg/L))', '망간(기준:0.3/ 단위:(mg/L))'은 컬럼명이 달라 엑셀에서 컬럼명을 통일
    #'1,4-다이옥산(기준:0.05/ 단위:(mg/L))'은 2010년 부터, '브롬산염(기준:0.01/ 단위:(mg/L))'는 2017년 부터, '포름알데히드(기준:0.5/ 단위:(mg/L))'는 2014년 부터 측정하여 컬럼 제거
    dfw = dfw.drop(['1,4-다이옥산(기준:0.05/ 단위:(mg/L))', '브롬산염(기준:0.01/ 단위:(mg/L))', '포름알데히드(기준:0.5/ 단위:(mg/L))'], axis=1)

    # '총대장균군(기준:0/ 단위:MPN)', '대장균/분원성대장균군(기준:0/ 단위:MPN)' 컬럼에 '불검출' 개수가 많아 컬럼 삭제
    dfw = dfw.drop(['총대장균군(기준:0/ 단위:MPN)', '대장균/분원성대장균군(기준:0/ 단위:MPN)'], axis=1)

    # '냄새(기준:0/ 단위:(mg/L))', '맛(기준:0/ 단위:(mg/L))' 컬럼은 모든 값이 '적합'이므로 컬럼 삭제
    dfw = dfw.drop(['냄새(기준:0/ 단위:(mg/L))', '맛(기준:0/ 단위:(mg/L))'], axis=1)

    # 탁도 단위(NTU)의 호환성 때문에 컬럼 drop
    dfw.drop('탁도(기준:0.5/ 단위:(NTU))', axis=1, inplace=True)
```

수질 데이터 전처리

Part 2,

전처리

```
In [66]: df.shape
Out[66]: (52031, 54)
```

```
In [67]: # 0을 NaN로 바꾸고 결측치 개수 = 행 개수(52031)인 경우만 drop하기
         df.replace(0, np.nan, inplace=True)

         # 모든 컬럼에 대하여 결측치 개수 확인
         nan_num = df.isnull().sum()                          # 결측치 수
         drop_list = list(nan_num[nan_num == len(df)].index)   # drop할 컬럼명 list
         water = df.drop(drop_list, axis=1)                   # 수질 전처리 파일명: water
```

결측치 0으로 대체

```
df = df.dropna(axis=0)
df.isnull().sum()
```

결측치 행 삭제

```
In [11]: df.groupby('수도사업자').mean()
```

```
In [12]: fill = lambda g: g.fillna(g.mean())
         df.groupby('수도사업자').apply(fill)
```

결측치 평균으로 대체

수질 데이터 전처리 - 결측치 처리 방법 결정

Part 2,

전처리

```
# 결측치를 가진 행 삭제
dfw = dfw.dropna(axis=0)

# 모든 값이 0인 컬럼 삭제
dfw.replace(0, np.nan, inplace=True)                  # 0 -> NaN 변경
nan_num = dfw.isnull().sum()                          # 컬럼별 결측치 수
drop_list = list(nan_num[nan_num == len(dfw)].index)  # 모든 값이 0인 컬럼의 컬럼명 list
water = dfw.drop(drop_list, axis=1)                   # 모든 값이 0인 컬럼 제거
water.fillna(0, inplace=True)                         # 남은 NaN -> 0 되돌리기

# 컬럼 명 변경
water.rename({'수도사업자':'지역'}, axis=1, inplace=True)

# ['검사월']을 연도만 남기고 ['year']로 컬럼명 변경
water['검사월'] = water['검사월'].str[0:4]
water.rename({'검사월':'year'}, axis=1, inplace=True)

# 수질 데이터 전처리 데이터프레임: water
return water
```

수질 데이터 전처리

Part 2,

전처리

```
def waterQualMean(self):            # 각 지역별, 연도별 1년 평균 물질 농도를 계산하여 dataframe을 반환하는 함수

    lst = []   # Dataframe 만들기 위해서 준비
    years = ['2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016','2017', '2018']
    cities = ['서울특별시', '부산광역시', '대구광역시', '인천광역시', '광주광역시', '대전광역시', '울산광역시', '세종특별자치시',
              '경기도', '강원도', '충청북도', '충청남도', '전라북도', '전라남도', '경상북도', '경상남도', '제주특별자치도']
    concs = water.columns[3:]

    for y in years:
        year = water['year'] == y
        water_y = water[year]
        for city in cities:
            ct_conc = [city + y]        # 하나의 도시에 대해, 이름과 모든 물질의 농도를 모은 리스트
            for conc in concs:
                ct_water = water_y[water_y['지역'].str.contains(city)]
                if conc == '일반세균(기준:100/ 단위:(CFU/mL))':
                    ct_water['월별물질농도(CFU/일)'] = ct_water['시설용량(㎥/일)'] * ct_water[conc] * (10**6)
                    t_conc = ct_water['월별물질농도(CFU/일)'].sum() / ct_water['시설용량(㎥/일)'].sum() * (10**-6)  # 2018년, xx지역, 일반세균 평균 농도(CFU/mL)

                elif conc == '수소이온농도(기준:5.8 ~ 8.5/ 단위:-)':
                    ct_water[conc] = ct_water[conc].apply(lambda x: 1/ 10**x)   # pH를 [H+](단위:mol/L)로 바꾸는 과정

                    ct_water['월별물질농도(mol/일)'] = ct_water['시설용량(㎥/일)'] * ct_water[conc] * 1000
                    t_conc = ct_water['월별물질농도(mol/일)'].sum() / ct_water['시설용량(㎥/일)'].sum() * 0.001   # 2018년, xx지역, 수소이온 평균 농도(mol/L)
                    t_conc = -np.log10(t_conc)  # 원래대로 pH로 변환

                else:    # '색도'포함
                    ct_water['월별물질농도(mg/일)'] = ct_water['시설용량(㎥/일)'] * ct_water[conc] * 1000
                    t_conc = ct_water['월별물질농도(mg/일)'].sum() / ct_water['시설용량(㎥/일)'].sum() * 0.001   # 2018년, xx지역, xx 물질 평균 농도(mg/L)
                ct_conc.append(t_conc)
            lst.append(ct_conc)


    result = pd.DataFrame(lst, columns=['지역'] + list(concs))
    result = result.set_index('지역').drop(['세종특별자치시2009','세종특별자치시2010','세종특별자치시2011','세종특별자치시2012','세종특별자치시2013',
                                        '세종특별자치시2014','세종특별자치시2015','세종특별자치시2016','세종특별자치시2017','세종특별자치시2018'], axis=0)
    return result
```

수질 데이터 전처리 – 지역별 연평균 수질 산출

Part 2,

전처리

```
def standardization(self, x):
    # Standardization
    scaler = StandardScaler()
    x_scaled = scaler.fit_transform(x)

    return x_scaled
```

데이터 표준화

Part 3,

분석

Part 3,

분석

```
def stepwiseSelection(self, x_scaled, y):          #변수선택법(단계별 선택법)

    colName = water.columns[3:]
    df = pd.DataFrame(x_scaled, columns=colName)
    df_y = y

    ## 전진 단계별 선택법
    variables = df.columns.tolist() ## 설명 변수 리스트

    ## 반응 변수 = 'y의 컬럼들'
    selected_variables = [] ## 선택된 변수들
    sl_enter = 0.05
    sl_remove = 0.05

    sv_per_step = [] ## 각 스텝별로 선택된 변수들
    adjusted_r_squared = [] ## 각 스텝별 수정된 결정계수
    steps = [] ## 스텝
    step = 0
    for i in range(0,6):
        y = df_y.iloc[:,i]
        y = list(y)
        while len(variables) > 0:
            remainder = list(set(variables) - set(selected_variables))
            pval = pd.Series(index=remainder) ## 변수의 p-value
            ## 기존에 포함된 변수와 새로운 변수 하나씩 돌아가면서
            ## 선형 모형을 적합한다.
            for col in remainder:
                X = df[selected_variables+[col]]
                X = sm.add_constant(X)
                model = sm.OLS(y,X).fit()
                pval[col] = model.pvalues[col]
```

변수선택법 – 단계별 선택법

Part 3,

분석

```
            min_pval = pval.min()
            if min_pval < sl_enter: ## 최소 p-value 값이 기준 값보다 작으면 포함
                selected_variables.append(pval.idxmin())
                ## 선택된 변수들에대해서
                ## 어떤 변수를 제거할지 고른다.
                while len(selected_variables) > 0:
                    selected_X = df[selected_variables]
                    selected_X = sm.add_constant(selected_X)
                    selected_pval = sm.OLS(y,selected_X).fit().pvalues[1:] ## 절편항의 p-value는 뺀다
                    max_pval = selected_pval.max()
                    if max_pval >= sl_remove: ## 최대 p-value값이 기준값보다 크거나 같으면 제외
                        remove_variable = selected_pval.idxmax()
                        selected_variables.remove(remove_variable)
                    else:
                        break

                step += 1
                steps.append(step)
                adj_r_squared = sm.OLS(y,sm.add_constant(df[selected_variables])).fit().rsquared_adj
                adjusted_r_squared.append(adj_r_squared)
                sv_per_step.append(selected_variables.copy())
            else:
                break
        print(selected_variables)
    return
```

변수선택법 – 단계별 선택법

Part 3,

분석

```
if __name__ == '__main__':
    VS = VS()

    water = VS.waterPreprocessing()
    x = VS.waterQualMean()
    y = VS.healthPreprocessing()

    x_scaled = VS.standardization(x)
    VS.stepwiseSelection(x_scaled,y)
```

전처리 ~ 변수선택법 실행

변수선택 결과

```
# 독립변수, 종속변수 리스트 생성
x_lst1 = ['증발잔류물(기준:500/ 단위:(mg/L))', '알루미늄(기준:0.2/ 단위:(mg/L))', '색도(기준:5/ 단위:(도))', '디브로모아세토니트릴(기준:0.1/ 단위:(mg/L))',
          '과망간산칼륨소비량(기준:10/ 단위:(mg/L))', '염소이온(기준:250/ 단위:(mg/L))', '불소(기준:1.5/ 단위:(mg/L))', '디브로모클로로메탄(기준:0.1/ 단위:(mg/L))']
y_lst1 = ['스트레스인지율_표준화율']
x_lst2 = ['염소이온(기준:250/ 단위:(mg/L))', '불소(기준:1.5/ 단위:(mg/L))', '증발잔류물(기준:500/ 단위:(mg/L))', '클로로포름(기준:0.08/ 단위:(mg/L))', '황산이온(기준:200/ 단위:(mg/L))',
          '세제(기준:0.5/ 단위:(mg/L))', '브로모디클로로메탄(기준:0.03/ 단위:(mg/L))', '클로랄하이드레이트(기준:0.03/ 단위:(mg/L))', '크롬(기준:0.05/ 단위:(mg/L))']
y_lst2 = ['스트레스로인한정신상담률_표준화율']
x_lst3 = ['염소이온(기준:250/ 단위:(mg/L))', '불소(기준:1.5/ 단위:(mg/L))', '붕소(기준:1/ 단위:(mg/L))', '잔류염소(기준:4/ 단위:(mg/L))', '증발잔류물(기준:500/ 단위:(mg/L))',
          '클로로포름(기준:0.08/ 단위:(mg/L))', '사염화탄소(기준:0.002/ 단위:(mg/L))']
y_lst3 = ['주관적구강건강이나쁜인구의분율_표준화율']
x_lst4 = ['알루미늄(기준:0.2/ 단위:(mg/L))', '과망간산칼륨소비량(기준:10/ 단위:(mg/L))', '염소이온(기준:250/ 단위:(mg/L))', '불소(기준:1.5/ 단위:(mg/L))', '디브로모클로로메탄(기준:0.1/ 단위:(mg/L))',
          '질산성질소(기준:10/ 단위:(mg/L))', '수소이온농도(기준:5.8 ~ 8.5/ 단위:-)', '1,2-디브로모-3-클로로프로판(기준:0.003/ 단위:(mg/L))', '톨루엔(기준:0.7/ 단위:(mg/L))']
y_lst4 = ['우울감경험률_표준화율']
x_lst5 = ['불소(기준:1.5/ 단위:(mg/L))', '증발잔류물(기준:500/ 단위:(mg/L))', '황산이온(기준:200/ 단위:(mg/L))', '알루미늄(기준:0.2/ 단위:(mg/L))', '망간(기준:0.05/ 단위:(mg/L))']
y_lst5 = ['우울증상으로인한정신상담률_표준화율']
```

분석

```
# 모든 값이 0인 컬럼 삭제
dfw.replace(0, np.nan, inplace=True)                    # 0 -> NaN 변경
nan_num = dfw.isnull().sum()                            # 컬럼별 결측치 수
drop_list = list(nan_num[nan_num == len(dfw)].index)    # 모든 값이 0인 컬럼의 컬럼명 list
water = dfw.drop(drop_list, axis=1)                     # 모든 값이 0인 컬럼 제거
water.fillna(0, inplace=True)                           # 남은 NaN -> 0 되돌리기

# 컬럼 명 변경
water.rename({'수도사업자':'지역'}, axis=1, inplace=True)

# ['검사월']을 연도만 남기고 ['year']로 컬럼명 변경
water['검사월'] = water['검사월'].str[0:4]
water.rename({'검사월':'year'}, axis=1, inplace=True)

water = water[['지역','year','시설용량(m³/일)'] + x]

# 수질 데이터 전처리 데이터프레임: water
return water
```

데이터 분석 – 독립변수 추출

분석

```
class MLR:
    def healthPreprocessing(self, y):
        # 2009 ~ 2018년 건강 데이터
        dfh_all = pd.read_excel(DATA_DIRS[0] + '//health_2009_2018_preprocessed.xlsx',
        dfh = pd.concat(dfh_all, ignore_index=True);

        # 분석 대상 지역 데이터 추출
        area = dfh['지역'].isin(['서울특별시','부산광역시','대구광역시','인천광역시','광주광역
        dfh = dfh[area]

        # 분석 대상 건강지표 컬럼 추출
        health = dfh[['시도'] + y]
        health = health.set_index('시도')

        # 건강 데이터 전처리 데이터프레임: health
        return health
```

데이터 분석 – 종속변수 추출

분석

```python
def mlr(self, x, y):
    # Standardization
    scaler = StandardScaler()
    x_scaled = scaler.fit_transform(x)

    # train, test data 나누기
    x_train, x_test, y_train, y_test = train_test_split(x_scaled, y, test_size=0.2)

    # Linear Regression
    lr = LinearRegression()
    lr.fit(x_train, y_train)

    # lr_coef = 기울기, lr_intercept = 절편
    lr_coef = lr.coef_
    lr_intercept = lr.intercept_

    # Test
    pd.DataFrame(lr.predict(x_test))
    score = lr.score(x_test, y_test)

    return score, lr, lr_coef, lr_intercept, x_test, y_test
```

데이터 분석 - 다중회귀분석

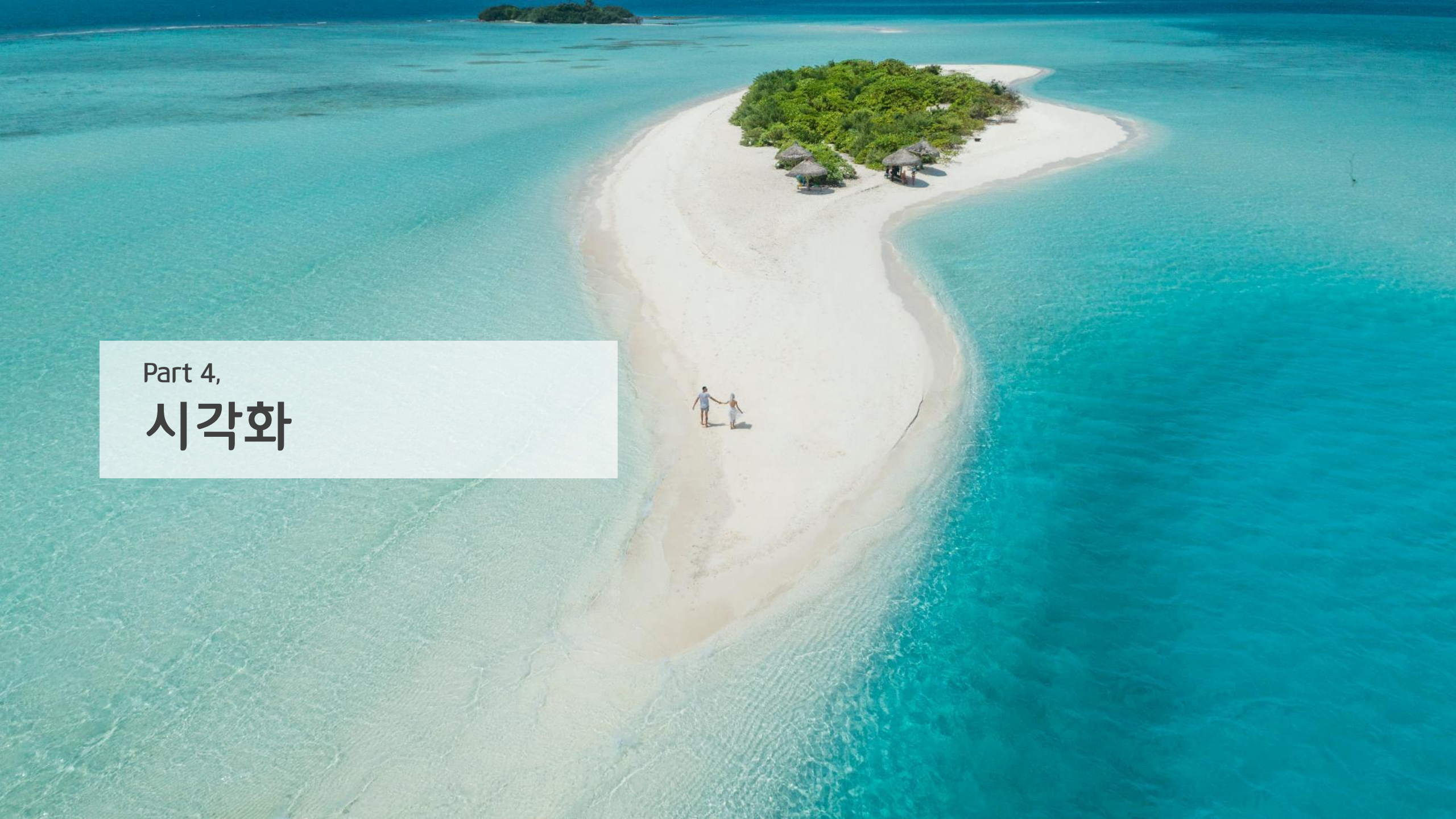
Part 3,

분석

```
# Multi Linear Regression 실행
water = MLR.waterPreprocessing(x_lst5)
x1 = MLR.waterQualMean()
y1 = MLR.healthPreprocessing(y_lst5)
score, lr, lr_coef, lr_intercept, x_test, y_test = MLR.mlr(x1,y1)
print(score)
```

다중회귀분석 실행

독립변수	종속변수	상관관계
20여 개의 수질 지표	1. 스트레스 인지율_표준화율	R-squared : 0.6088644122003146
	2. 스트레스로 인한 정신상담률_표준화율	R-squared : 0.5029349543538737
	3. 주관적 구강건강이 나쁜 인구_표준화율	R-squared : 0.5658958905007379
	4. 우울감 경험률_표준화율	R-squared : 0.5964814609962665
	5. 우울증상으로 인한 정신상담률_표준화율	R-squared : 0.1917737492610384



Part 4,

시각화

Part 4,

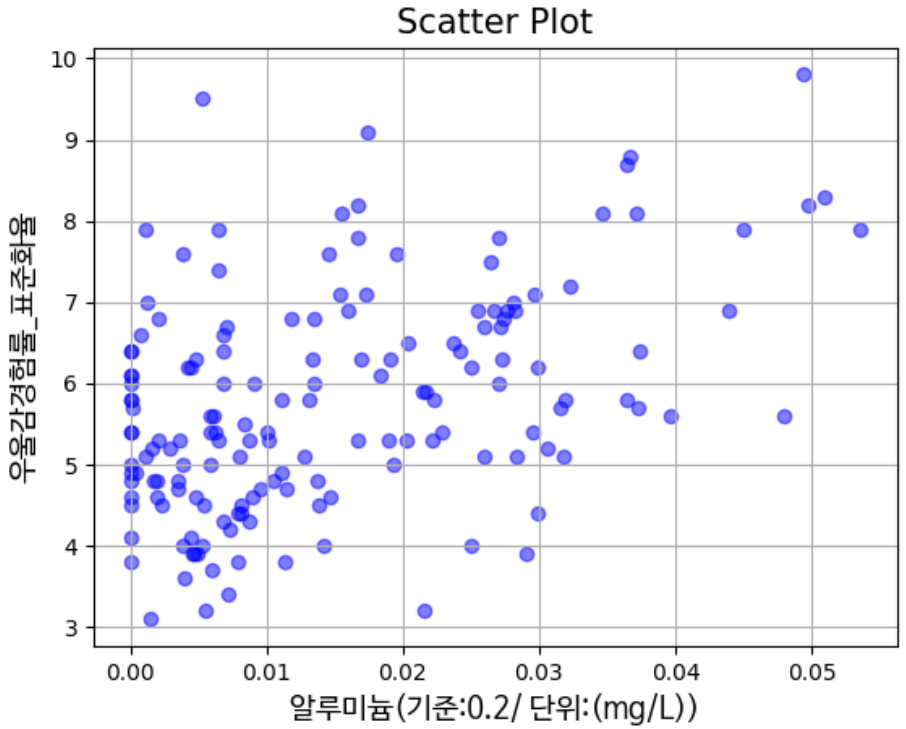
시각화

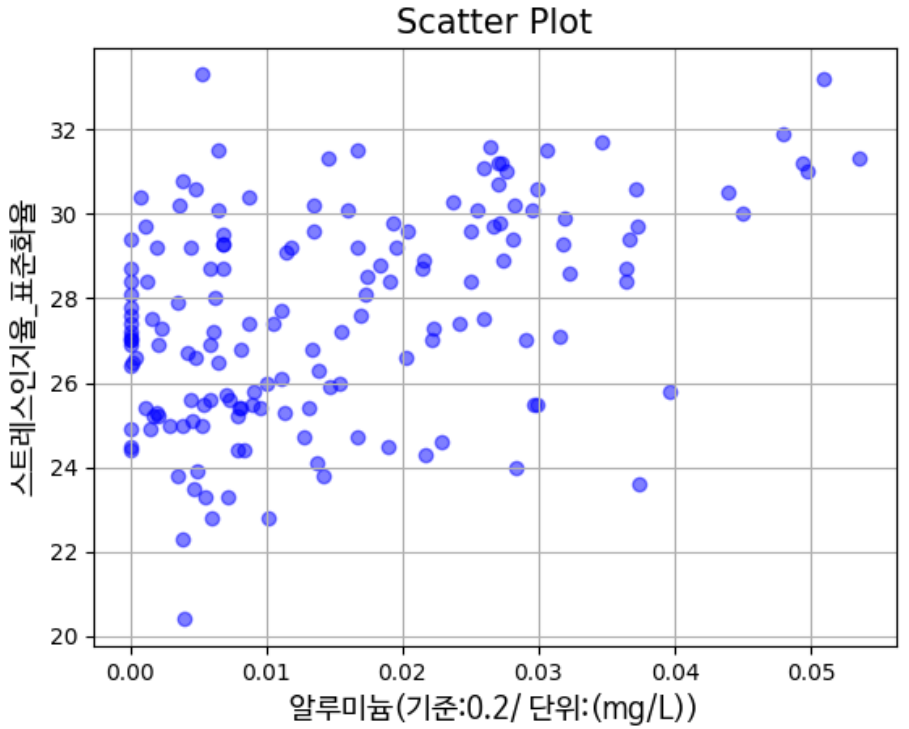
```
def visual1(self, x, x_lst, y, y_lst):
    for i in range(0,len(x_lst)):
        plt.title("Scatter Plot", fontsize=15)
        plt.scatter(x[x_lst[i]], y[y_lst], c='b', alpha=0.5)
        plt.xlabel(x_lst[i], fontproperties=fontprop, fontsize=13)
        plt.ylabel(y_lst[0], fontproperties=fontprop, fontsize=13)
        plt.grid()
        plt.show()
```

시각화 – 상관관계 산점도

Part 4,

시각화





시각화 – 상관관계 산점도

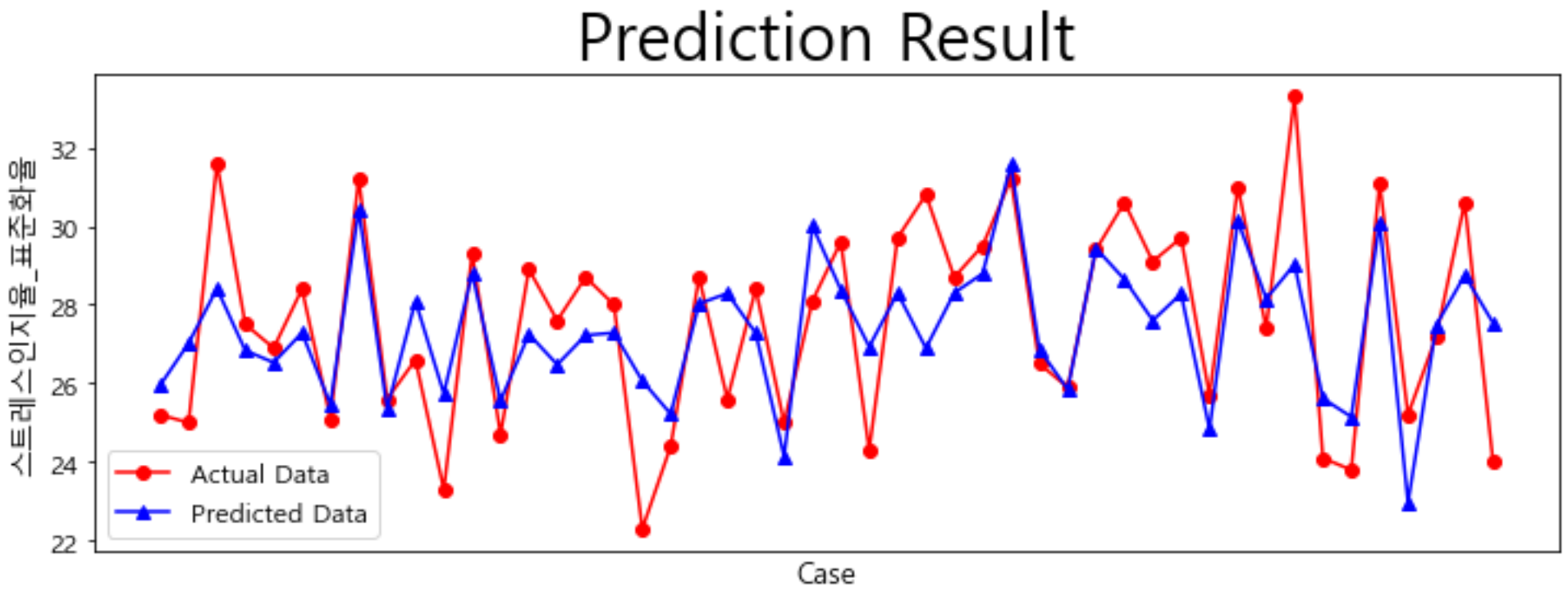
시각화

```python
def visual2(self, y_name):
    fig = plt.figure(figsize=(20, 5))
    graph = fig.add_subplot(1,1,1)
    graph.plot(y_test, marker='o', color='r', label='Actual Data')
    graph.plot(lr.predict(x_test), marker='^', color='b', label='Predicted Data')
    graph.set_title('다중회귀분석 예측 결과', size=30)
    plt.xlabel("Case", fontsize=13)
    plt.ylabel(y_name[0], fontproperties=fontprop, fontsize=13)
    plt.legend(loc='best')
    plt.show()
```

시각화 – 다중회귀분석 예측 결과

Part 4,

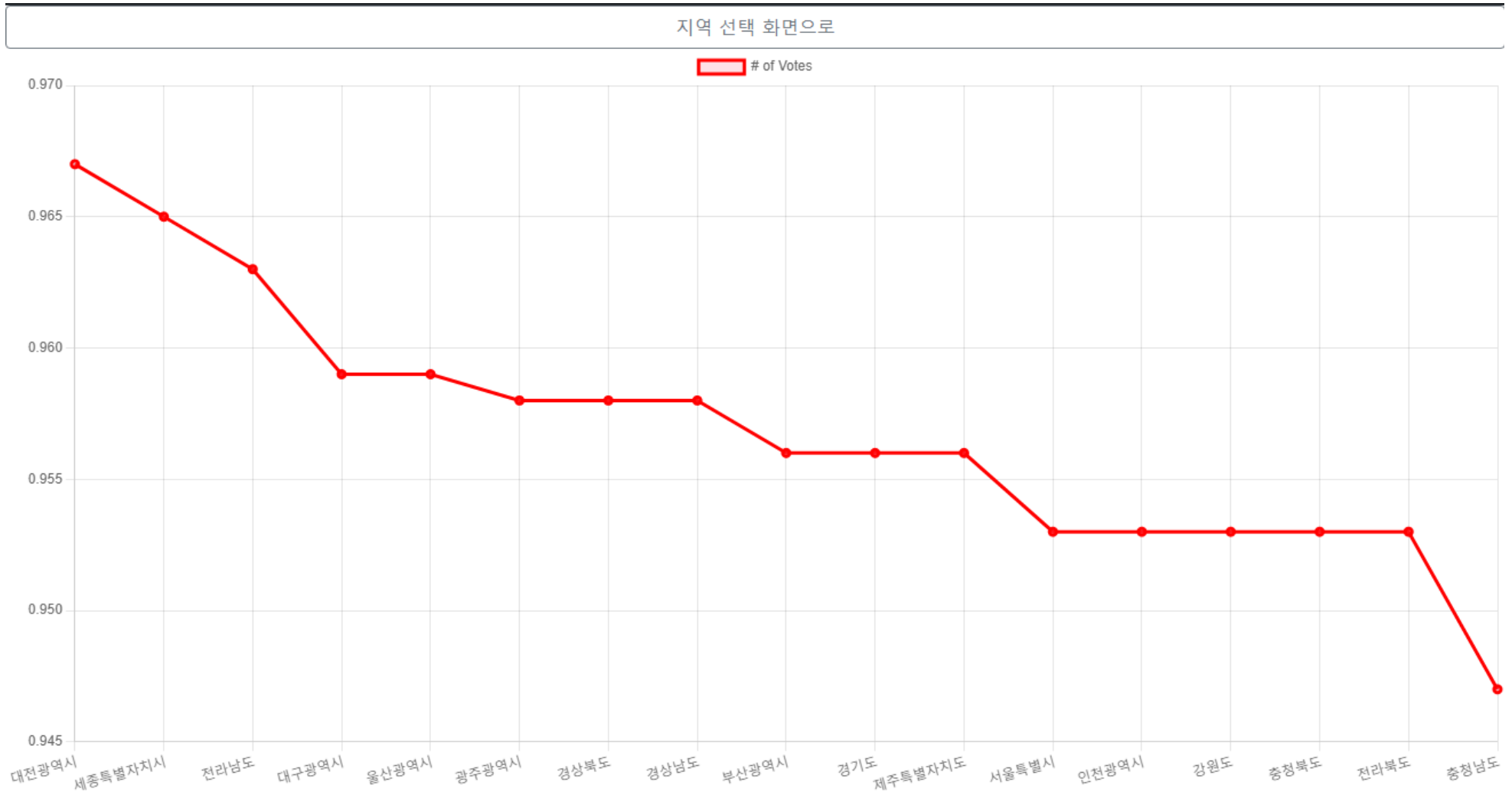
시각화



시각화 – 다중회귀분석 예측 결과

Part 4,

시각화

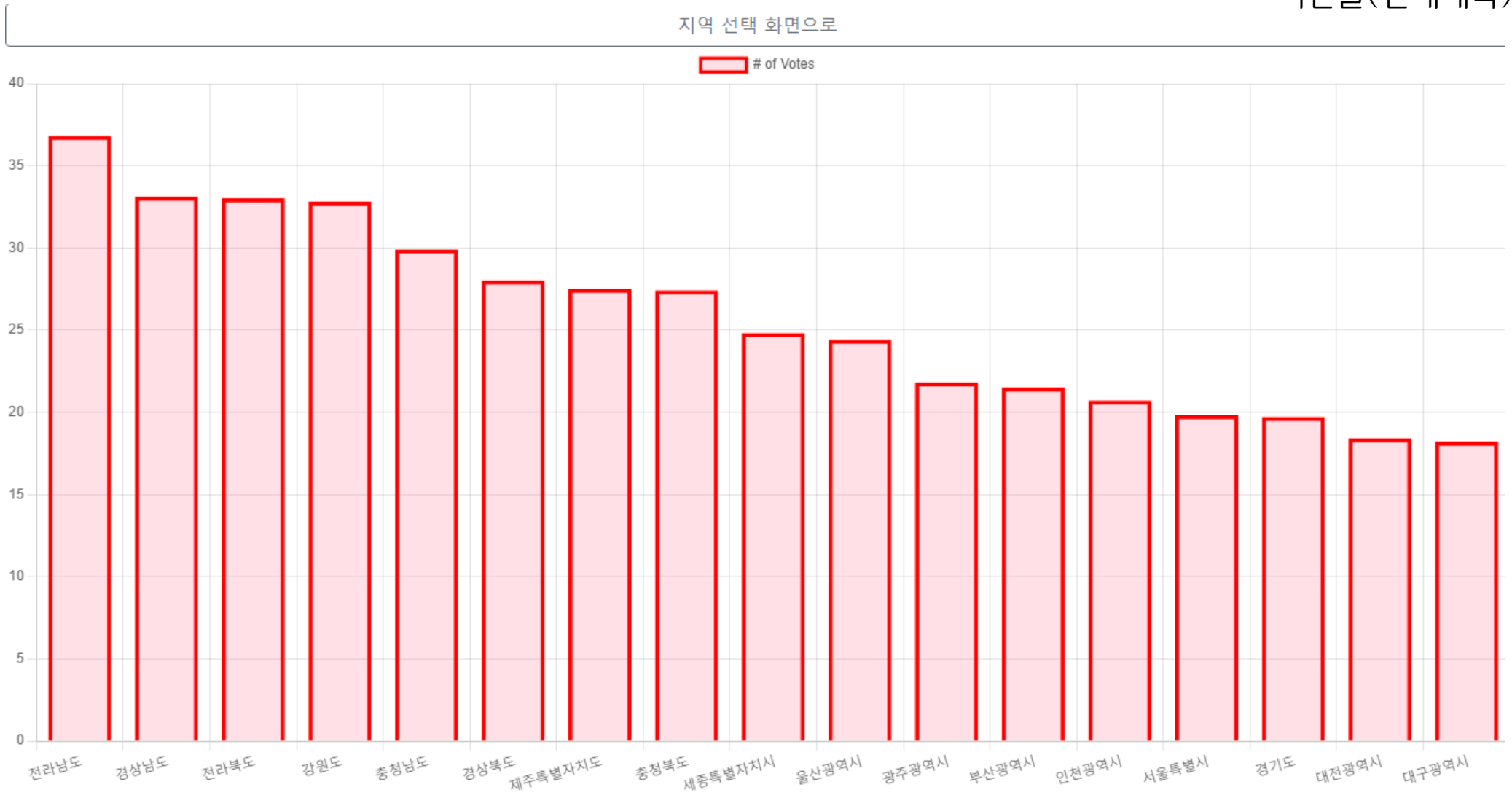


삶의 질 지수
(EQ-5D)_표준화율

Part 4,

시각화

비만율(신체계측)_표준화율



Part 5,

결론

결론

분석 목표

상수도 수질이 지역사회 건강에 미치는 영향

지역별 상수도 수질 데이터와 지역사회 건강결과 및 건강결정요인 데이터를 활용하여
상수도 수질이 지역사회 건강에 끼치는 영향을 파악

Part 5,

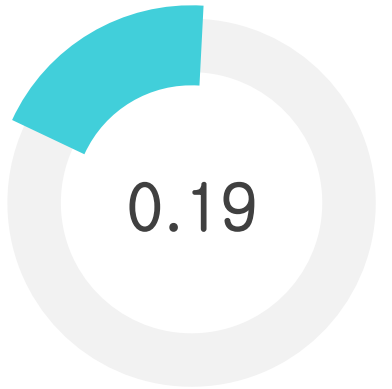
결론

스트레스 인지율

스트레스로 인한 정신상담률

스트레스 인지율_표준화율

우울감 경험률



우울증상으로 인한 정신상담률

결정계수(R^2, R-Square) : 회귀모형 내에서 설명변수 x로 설명할 수 있는 반응변수 y의 변동 비율

Part 5,

결론

분석 발전 방향 : 다양한 환경 지표를 결합하여 환경이 건강에 미치는 영향

자료 출처

1)지역별 상수도 수질 데이터
https://www.waternow.go.kr/web/lawData?pMENUID=4

2)지역사회 건강 결과 및 건강결정요인
https://chs.kdca.go.kr/chs/recsRoom/dataBaseMain.do

감사합니다