

광우	어력	대 90,000 중 70,000 소 60,000
감성동	시가	

시가의 공포...

스끼다시조

팀장: 장재훈

팀원: 김용현, 윤성준, 이경희

목차

1. 데이터 설명
 - a. 활용 데이터 출처 및 학계
 - b. 데이터 전처리 과정
2. 데이터 시각화 (1년 전체 평균가 또는 최저 평균가)
 - a. 네이버 openAPI를 활용한 선호도 조사 워드클라우드로 구현
 - b. 설문조사 결과를 바탕으로 3개 어종 선정(지역구분 없이 전체자료)
3. 감성동
 - a. 3년 가격 변동 시각화
 - b. 계절별 생산량 및 생산지 시각화
 - c. 수온 - 생산량 상관관계 분석
 - d. 생산량 - 평균가 상관관계 분석
 - e. 감성동 제철 시기 및 추천 지역 제공
4. 방어
 - a. 3년 가격 변동 시각화
 - b. 계절별 생산량 및 생산지 시각화
 - c. 수온 - 생산량 상관관계 분석
 - d. 생산량 - 평균가 상관관계 분석
 - e. 감성동 제철 시기 및 추천 지역 제공
5. 광어
 - a. 3년 가격 변동 시각화
 - b. 계절별 생산량 및 생산지 시각화
 - c. 수온 - 생산량 상관관계 분석
 - d. 생산량 - 평균가 상관관계 분석
 - e. 감성동 제철 시기 및 추천 지역 제공

1. 데이터 설명

a. 해양수산부 위판 자료

i. 출처 : [해양수산부_수산물_위판정보](#)

ii. 기간 : 19년 1월 ~ 22년 1월

iii. 용량 : 총 약 300MB

b. 해양수산부 어획량 자료

c. 바다누리 수온 자료

i. 분단위 측정 데이터를 월로 환산하는 작업

ii. open API 호출량 초과 등 제한 사항

d. 해면어업 생산량 평균가 데이터

i. 시각화 및 상관관계 분석을 위해 생성

ii. 수산물 위판정보 자료 사용

iii. 해면어업 생산량(광어, 방어, 김성듬) 자료 사용

데이터 전처리 과정 설명

위판 정보 데이터

이상값오기, null 값 등의 제거

- 필요한 컬럼 필터링 및 컬럼명 변경
- 사용 어종 선정 및 산지조합 선정

19개의 어종 선정 - ['봉장어', '민어', '넙치', '조피볼락', '낙지', '문어', '전복', '갑오징어', '고등어', '송어', '전어', '감성듬', '돌돔', '방어', '참돔', '참다랑어', '꼼뱅이', '송어', '밀복', '참복']

```
fish = ['봉장어', '민어', '넙치', '조피볼락', '낙지', '문어', '전복', '갑오징어', '고등어', '송어', '전어', '감성듬', '돌돔', '방어', '참돔', '참다랑어', '꼼뱅이', '송어', '밀복', '참복']

def fish_choice(df):
    for row in range(len(df)):
        if (df.loc[row, '수산물표준코드명'] not in fish) or (df.loc[row, '어종상태명'] != '(활)' :
            df.drop([row], axis = 0, inplace = True)
```

20개 산지조합명 선정 - '인천', '서산', '안면도', '대천서부', '서천서부', '군산시', '목포', '전남서부어류양식', '고흥군', '여수', '남해군', '통영', '거제', '부산시', '울산', '포항', '삼척', '강릉시', '속초시', '동해시'

(원본 데이터 자료의 개수와 크기가 작은 편은 아니다보니 필터링하는 과정에서 상당한 시간이 소요됨)

각 일자별 어종별 산지조합별 평균가 계산 후 평균가 0인 행 제거

```
def value_avg(df) :
    for row in range(len(df)-1):
        if ( df.loc[row, '위판일자'] == df.loc[row+1, '위판일자'] )
            and ( df.loc[row, '산지조합명'] == df.loc[row+1, '산지조합명'] )
            and ( df.loc[row, '수산물표준코드명'] == df.loc[row+1, '수산물표준코드명'] ) :
            df.loc[row+1, '평균가'] = ( (df.loc[row, '평균가']) + df.loc[row+1, '평균가'] ) / 2
    df.drop( [row], axis=0, inplace = True)
```

이상치를 평균값으로 대체

19개 어종 중 워드클라우드와 맵캡 수강생들의 투표를 통해 산출된 선호도 지표를 통해 3개의 어종 선정 후 이에 대한 이상치(사분위절삭)를 평균값으로 대체

```
for row in range(len(df)):
    if df.loc[row,'수산물표준코드명'] == '넙치' and df.loc[row,'평균가'] > 132878.0 :
        df.loc[row,'평균가'] = 42524.0
    elif df.loc[row,'수산물표준코드명'] == '방어' and df.loc[row,'평균가'] > 95719.0 :
        df.loc[row,'평균가'] = 30875.0
    elif df.loc[row,'수산물표준코드명'] == '감성돔' and df.loc[row,'평균가'] > 130985.0 :
        df.loc[row,'평균가'] = 45054.0
    else : pass
```

수온 및 생산량 데이터

수온데이터(바다누리 API)

- 데이터 출처 : 국립해양조사원에서 제공하는 해양정보 데이터를 openAPI로 제공
- 제공 사이트에서 openAPI access token을 신청하여 발급 받은 token으로 API 사용

데이터 전처리(관측소)

`requests`라는 HTTP 요청 라이브러리를 사용하여 openAPI에서 제공하는 요청 URL을 요청

```
# 관측소별 데이터 제공
observe = 'http://www.khoa.go.kr/api/oceangrid/ObsServiceObj/search.do?ServiceKey=AYCy7rXZP9Q1qVTilyuWg==&ResultType=json'
print(observe)
```

요청 변수에 제공하는 `Parameter`를 대입하여 원하는 정보를 `json` 타입으로 추출

```
data
{'result': {'data': [{"data_type': '조위관측소',
  'obs_lat': '35.024178',
  'obs_lon': '128.810933',
  'obs_object': '조위,수온,염분,기온,기압,풍속,풍향',
  'obs_post_id': 'DT_0063',
  'obs_post_name': '가덕도'},
 {"data_type': '조위관측소',
  'obs_lat': '37.731944',
  'obs_lon': '126.522222',
  'obs_object': '조위,기압',
  'obs_post_id': 'DT_0032',
  'obs_post_name': '강화대교'},
 {"data_type': '조위관측소',
  'obs_lat': '34.028333',
```

`json` 타입의 데이터를 전처리하여 `DataFrame` 타입으로 변경

	obs_post_id	obs_lat	data_type	obs_post_name	obs_lon	obs_object
0	DT_0063	35.024178	조위관측소	가덕도	128.810933	조위,수온,염분,기온,기압,풍속,풍향
1	DT_0031	34.028333	조위관측소	거문도	127.308889	조위,수온,염분,기온,기압,풍속,풍향
2	DT_0029	34.801389	조위관측소	거제도	128.699167	조위,수온,염분,기온,기압,풍속,풍향
3	DT_0026	34.481111	조위관측소	고흥발포	127.342778	조위,수온,염분,기온,기압,풍속,풍향
4	DT_0018	35.975556	조위관측소	군산	126.563056	조위,수온,염분,기온,기압,풍속,풍향

데이터 전처리(수온 데이터)

마찬가지로 수온데이터를 요청할 수 있는 URL을 통해 Parameter로 원하는 날짜와 관측소를 대입하여 요청

```
# 지정 관측소에서 하루 시간별 수온 측정 데이터
endpoint = 'http://www.khoa.go.kr/api/oceangrid/tideObsTemp/search.do?ServiceKey=AVCy7rXZP9QlqvTilyuWg==&ObsCode=DT_0017&Date=20200902&ResultType=json'
print(endpoint)
```

대산 관측소의 2020년 09월 02일 수온 데이터를 json 타입으로 요청

```
data
```

```
{'result': {'data': [{'record_time': '2020-09-02 00:00:00',
    'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:01:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:02:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:03:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:04:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:05:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:06:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:07:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:08:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:09:00', 'water_temp': '24.4'},
    {'record_time': '2020-09-02 00:10:00', 'water_temp': '24.4'}],
```

위의 요청 데이터를 보면 1회 요청시마다 1일 00:00 ~ 23:59분 까지 분당 수온 데이터를 수집

1일 요청 초과 이유

- 본 프로젝트에서는 3개년(2019, 2020, 2021)년 관측소별 월별 수온 데이터가 필요한 상황.
- 해당 사이트의 하루 요청 건수는 20,000건으로 제한 되어있고, 프로젝트에 필요한 데이터는 총 32,850건을 요청하는 도중 하루 요청 허용치를 넘었다는 오류가 발생.

총 요청 관측소 * 총 일수(2019 ~ 2021년) = 총 요청 건수

$$30개소 * 1,095일 = 32,850건$$

- 다행히 요청을 초과했음에도 불구하고 차단 당하지는 않아 이틀이 걸쳐 다시 요청 진행

전처리 코드 구성

```
for 관측소: # 30개 관측소
    for Month data: # 2019.01 ~ 2021.12
        for daily data: # 해당 월의 1일부터 말일까지
            endpoint='http://www.~/요일/관측소/ResultType=json'
            daily.mean() # 일평균 온도
            관측소[월] = month.mean() # 월평균 온도
            관측소별.append(관측소 월평균 데이터)
```

3중 for문으로 관측소마다 매일 수온데이터를 요청받아 일평균 온도의 평균값을 월별로 수집하여 해당 관측소에 추가하고, 월별 수온 데이터가 입력된 관측소 데이터를 모아 DataFrame 생성

어획량 데이터

데이터 출처 : 통계청

데이터 전처리

- 통계청에서 수집한 csv파일을 어종별, 지역별로 나눠 전처리 수행

- csv 파일구성

어업별	품종별	어법별	행정구역별	판매형태별	항목	단위	월별 (2019.01~2021.12)
일반해 면어업	넙치류	계	부산광역시	계	활어	NaN	546.27914
...
천해양 식어업	방어류	계	제주도	계	활어	NaN	90880.0

- 품종별 전처리

어업별로 크게 일반해면어업, 천해양식어업으로 나누고, 품종별(19종)로 정렬하여 각 품종별 지역의 생산량 값을 모두 합치는 방식으로 전처리

품종별	2019.01 월	2019.02 월	2019.03 월	2019.04 월	2019.05 월	2019.06 월	2019.07 월	2019.08 월
0 감성들	78.913	57.263	63.026	151.241	142.56	36.46	19.732	29.908
1 갑오징어류	15.162	0.586	37.339	525.969	1065.125	559.597	58.161	3.608

- 행정구역별 전처리

역시 어업별로 일반해면어업, 천해양식어업으로 나누고, 행정구역별로 정렬하여 본 프로젝트에서 분석할 3개의 어종(넙치(광어), 방어, 감성들)만 전처리를 수행

어업별	품종별	행정구역별	2019.01 월	2019.02 월	2019.03 월	2019.04 월	2019.05 월	2019.06 월
19 일반해 면어업	넙치류	부산광역시	1.60624	0.52372	0.44332	1.04709	4.47746	6.4978
20 일반해 면어업	넙치류	인천광역시	2.41037	0.036	7.76703	35.69026	65.6077	22.0918

해면어업_생산량 평균가 데이터

광어 (→ 방어, 감성돔도 동일한 과정으로 생성)

수산물 위판정보에서 광어와 관련된 행만 필터링

해면어업 생산량 DF의 형식을 보기 좋게 정제한 후, 행정구역별 일자별 평균가를 넣을 DF하고 모든 값 0으로 초기화

(DF 형식을 정제하고 맞추는 과정에서 생각보다 작업해야하는 부분들이 많았음)

수산물 위판정보의 지역을 행정구역별로 그룹화하여 일자별 행정구역별로 해당 어종 평균가 계산 후 값 대체

```
# 일자별로 각 행정구역에 해당하는 지역별로 평균내기
for y in range(3):
    for m in range(1,13):
        Busan, Incheon, Ulsan, Gangwon, Chungnam, Jeonbuk, Jeonnam, Gyeongbuk, Gyeongnam = [], [], [], [], [], [], [], []
        for d in range(1,32):
            for row in range(len(ff_df)):
                if ff_df.loc[row,'산지조합명'] == '부산시' and ff_df.loc[row,'위판일자'] == f'{2019+y}-{m}-{d}' : Busan.append(ff_df.loc[row,'평균가'])
                elif ff_df.loc[row,'산지조합명'] == '인천' and ff_df.loc[row,'위판일자'] == f'{2019+y}-{m}-{d}' : Incheon.append(ff_df.loc[row,'평균가'])
                elif ff_df.loc[row,'산지조합명'] == '울산' and ff_df.loc[row,'위판일자'] == f'{2019+y}-{m}-{d}' : Ulsan.append(ff_df.loc[row,'평균가'])
                elif (ff_df.loc[row,'산지조합명'] == '삼척' or ff_df.loc[row,'산지조합명'] == '강릉시') or ff_df.loc[row,'산지조합명'] == '속초시' or ff_df.loc[row,'산지조합명'] == '대전서부' or ff_df.loc[row,'산지조합명'] == '안면도' or ff_df.loc[row,'산지조합명'] == '군산시' and ff_df.loc[row,'위판일자'] == f'{2019+y}-{m}-{d}' : Jeonbuk.append(ff_df.loc[row,'평균가'])
                elif ff_df.loc[row,'산지조합명'] == '삼척' or ff_df.loc[row,'산지조합명'] == '목포' or ff_df.loc[row,'산지조합명'] == '전남서부이류양식' or ff_df.loc[row,'산지조합명'] == '포항' and ff_df.loc[row,'위판일자'] == f'{2019+y}-{m}-{d}' : Gyeongbuk.append(ff_df.loc[row,'평균가'])
                elif (ff_df.loc[row,'산지조합명'] == '남해군' or ff_df.loc[row,'산지조합명'] == '통영' or ff_df.loc[row,'산지조합명'] == '거제') and ff_df.loc[row,'산지조합명'] == '경상북도', f'{2019+y}-{m}' ] = round(sum(Busan)/len(Busan), 1)
            try : ff_price.loc['부산광역시',f'{2019+y}-{m}'] = round(sum(Busan)/len(Busan), 1)
            except : pass
            try : ff_price.loc['인천광역시',f'{2019+y}-{m}'] = round(sum(Incheon)/len(Incheon), 1)
            except : pass
            try : ff_price.loc['울산광역시',f'{2019+y}-{m}'] = round(sum(Ulsan)/len(Ulsan), 1)
            except : pass
            try : ff_price.loc['강원도',f'{2019+y}-{m}'] = round(sum(Gangwon)/len(Gangwon), 1)
            except : pass
            try : ff_price.loc['충청남도',f'{2019+y}-{m}'] = round(sum(Chungnam)/len(Chungnam), 1)
            except : pass
            try : ff_price.loc['전라북도',f'{2019+y}-{m}'] = round(sum(Jeonbuk)/len(Jeonbuk), 1)
            except : pass
            try : ff_price.loc['전라남도',f'{2019+y}-{m}'] = round(sum(Jeonnam)/len(Jeonnam), 1)
            except : pass
            try : ff_price.loc['경상북도',f'{2019+y}-{m}'] = round(sum(Gyeongbuk)/len(Gyeongbuk), 1)
            except : pass
            try : ff_price.loc['경상남도',f'{2019+y}-{m}'] = round(sum(Gyeongnam)/len(Gyeongnam), 1)
            except : pass
```

(각 행정구역별 일자별 값을 대체하는 과정에서 예외처리 부분에 오류가 생겼는데 원인을 찾는데 많은 시간이 소요됨
)

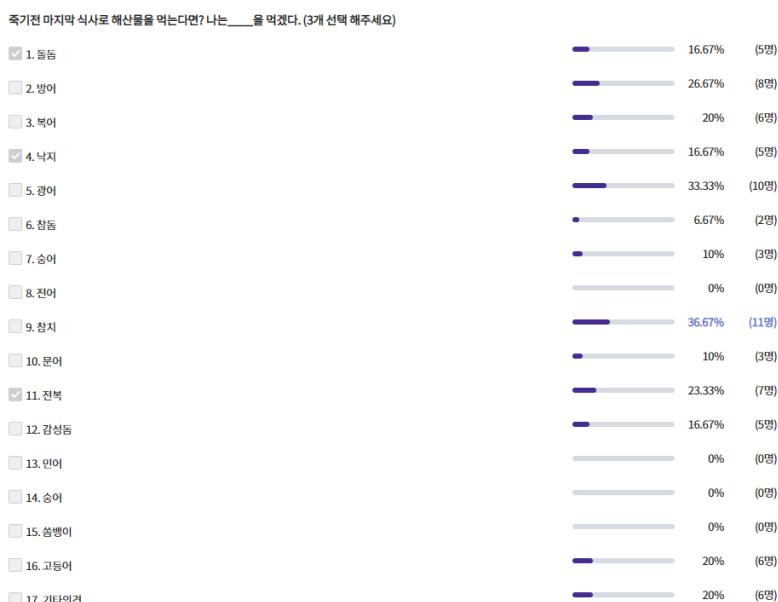
2. 횟감 선호도 조사(3년치 네이버 검색량 활용)



돌돔

- 검색량이 많다는 것을 선호 및 관심도가 높다는 것으로 가정
- 쉽게 접하기 어려운 고급 어종 등의 정보 획득을 위해 검색량이 높을 것이라고 가정
- 고가이며 생소한 어종인 돌돔과 감성돔, 참돔 등은 상대적으로 검색량이 저조한 반면, 대중적 어종인 문어, 방어, 고등어, 참치 전복 등의 검색량이 높았음.

이에 선호도 좀 더 구체적으로 확인하기 위해 설문조사를 실시.



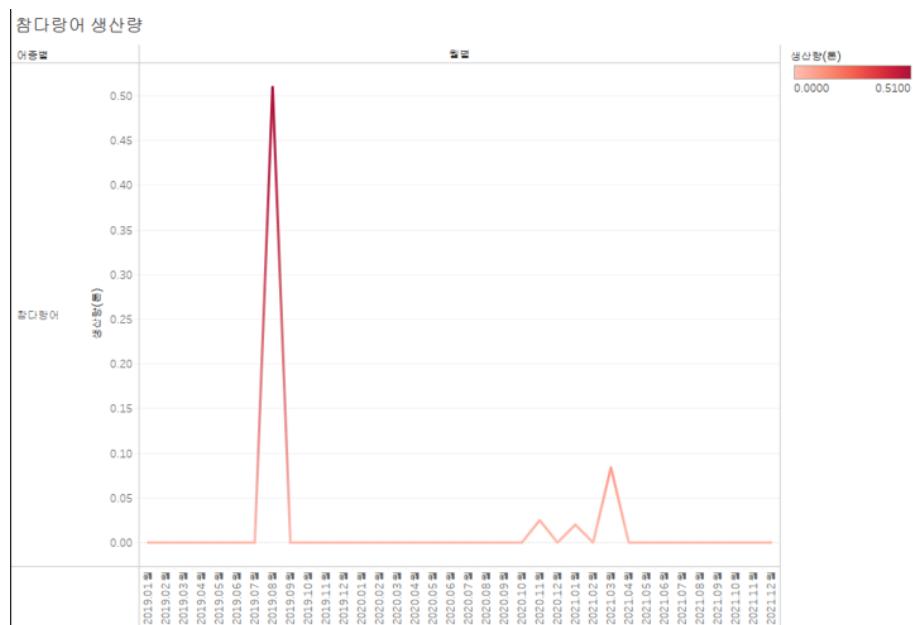
총 56명을 대상으로 3개의 복수 답변을 허용하여 설문을 실시한 결과 28명 응답하였습니다.

설문조사 결과

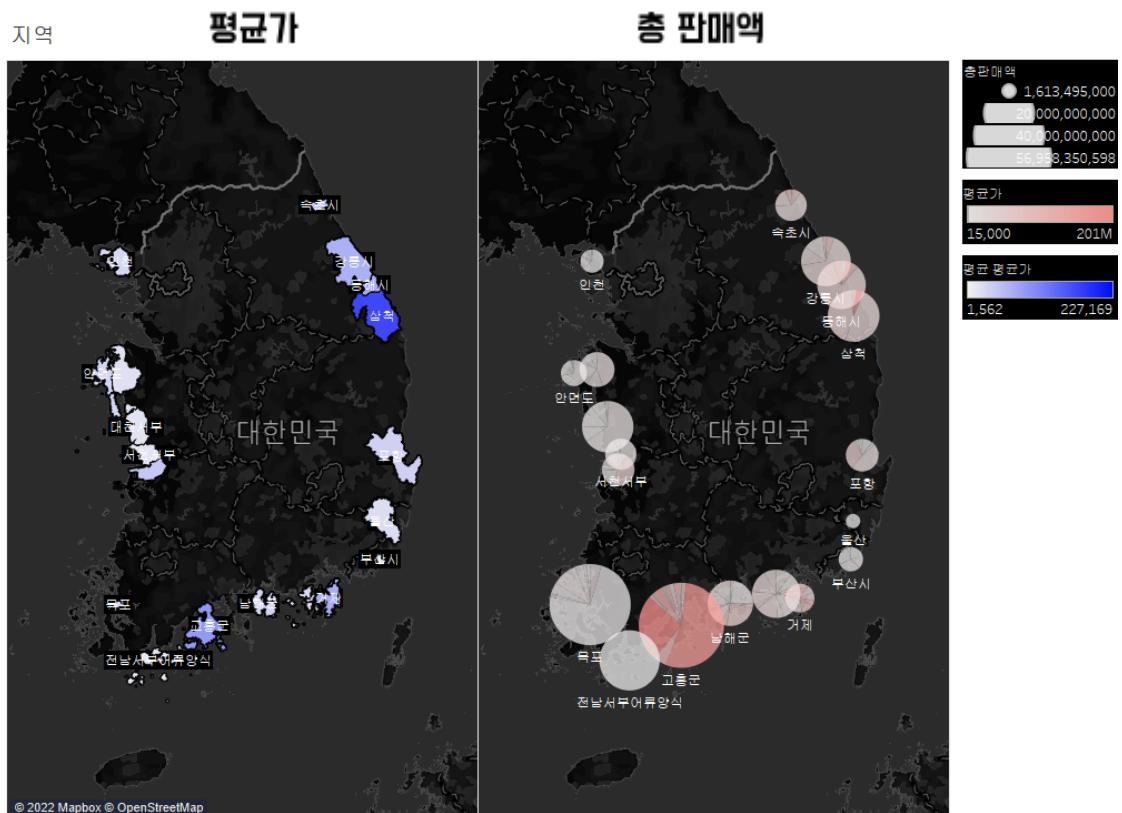
	20% 이상	15 % 이상
1순위	참치	감성돔
2순위	광어	돌돔
3순위	방어	낙지
4순위	전복	

가격과 선호도를 고려하여 광어, 방어, 감성돔을 선정

참치가 1등인데 분석에 포함되지 않은 이유

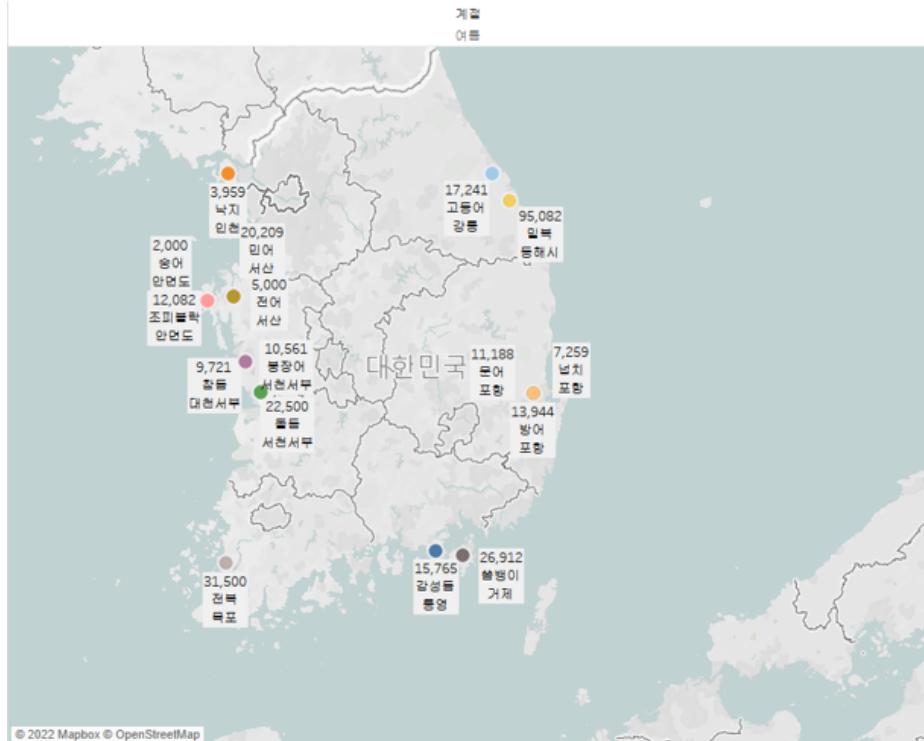


19개 어종 분석 및 시각화

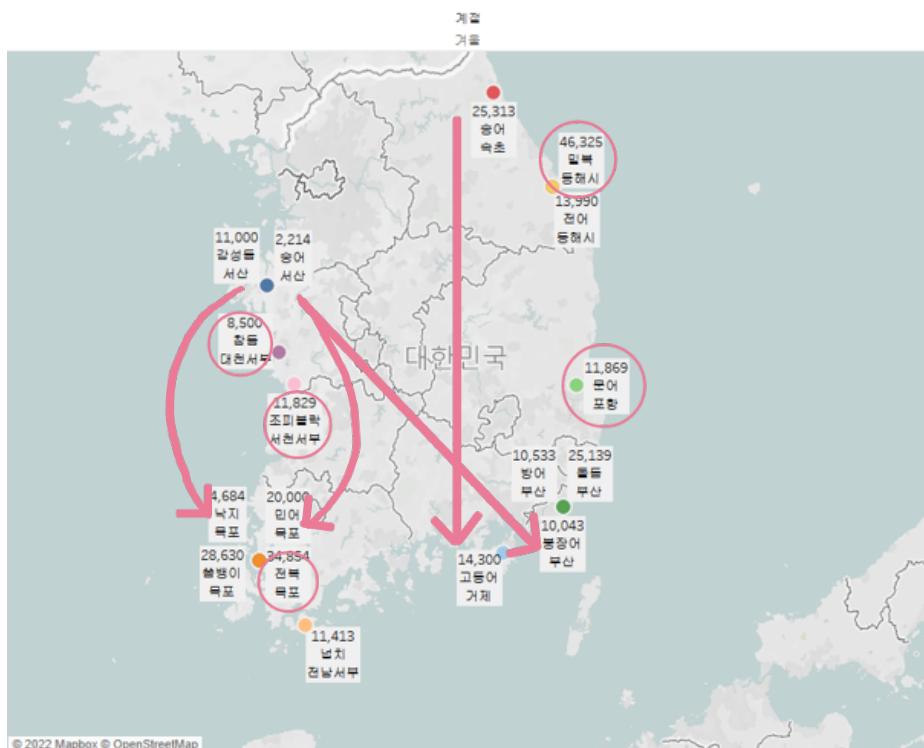


구체적인 어종들의 계절별 최저가를 표시해보면.

여름 어종별 최저가 지역



겨울 어종별 최저가 지역

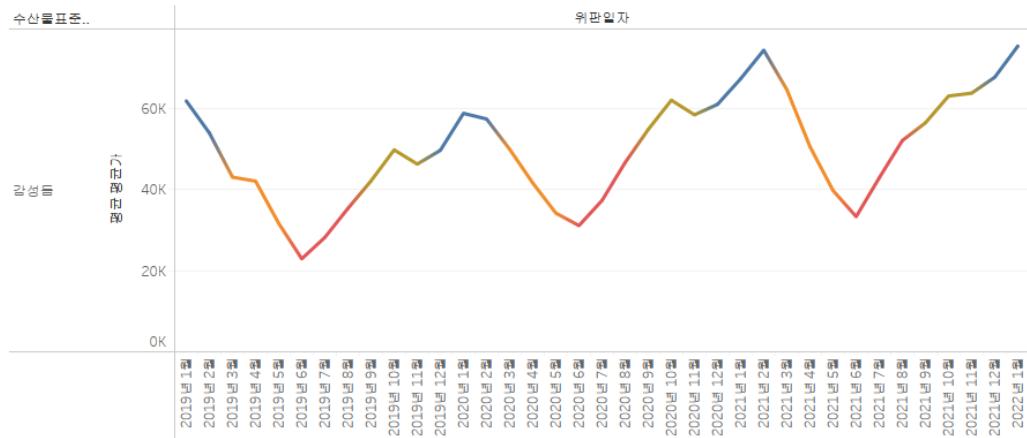


- 확실히 겨울이 되면 여름철보다 따뜻한 남해 쪽으로 어종들이 몰리는 것 같습니다.

감광방 분석

3. 감성동

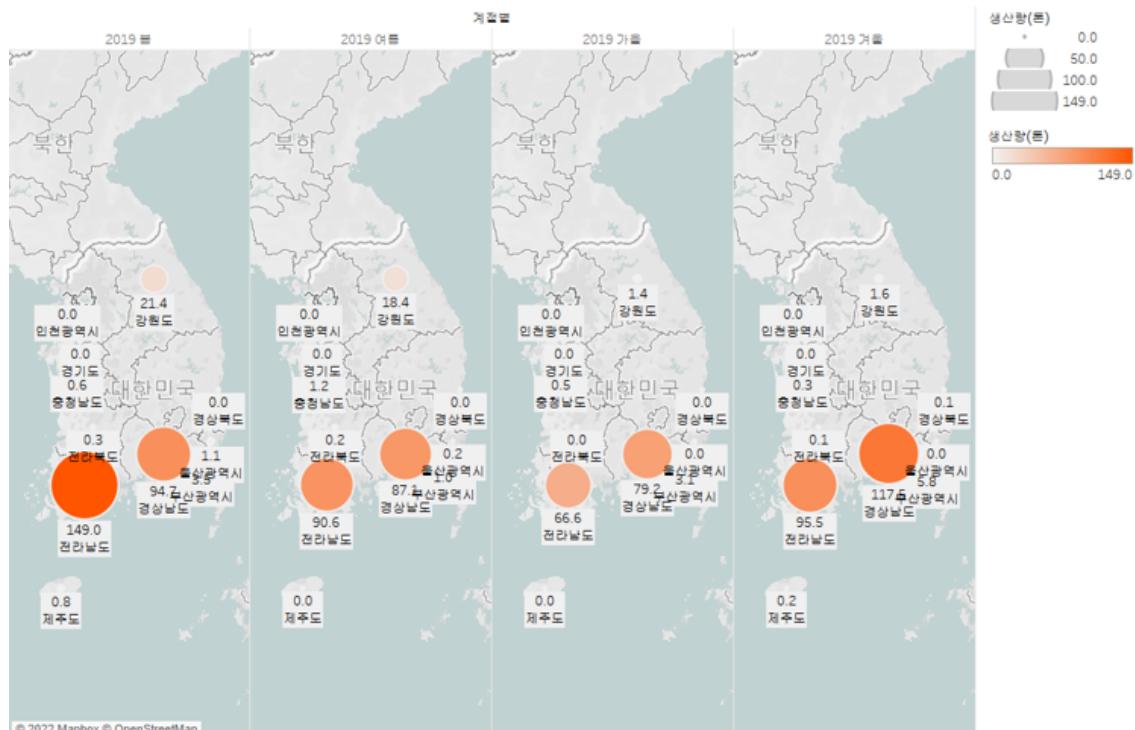
감성동 가격 변동



1월부터 5월까지 하락, 6월부터 12월까지 상승
가격이 생산량 및 수온의 영향을 받고 있는지 분석해보겠습니다.

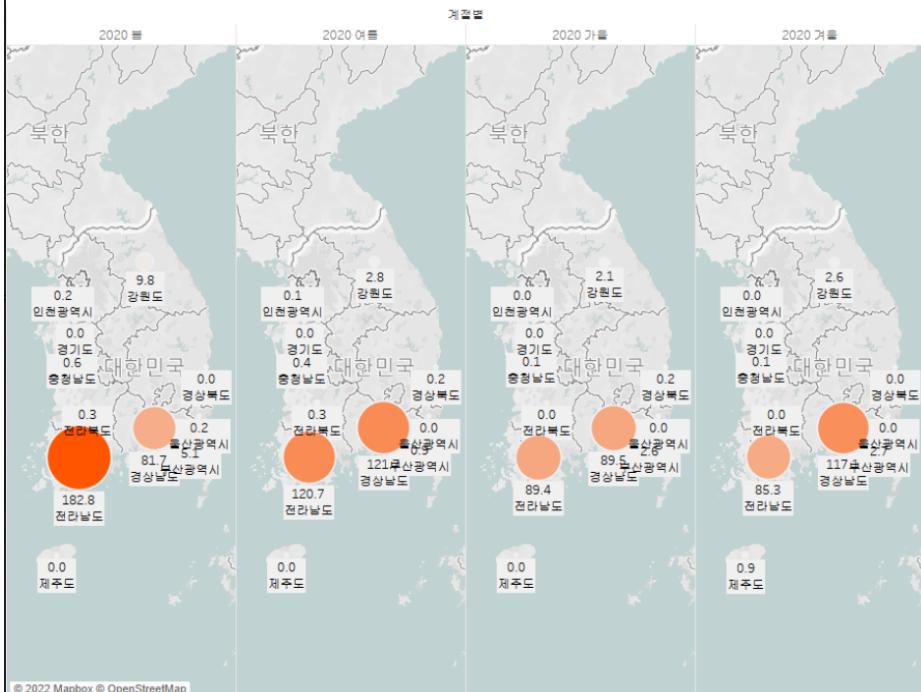
감성동 생산량 분포입니다.

2019년 감성동 생산량(지도)



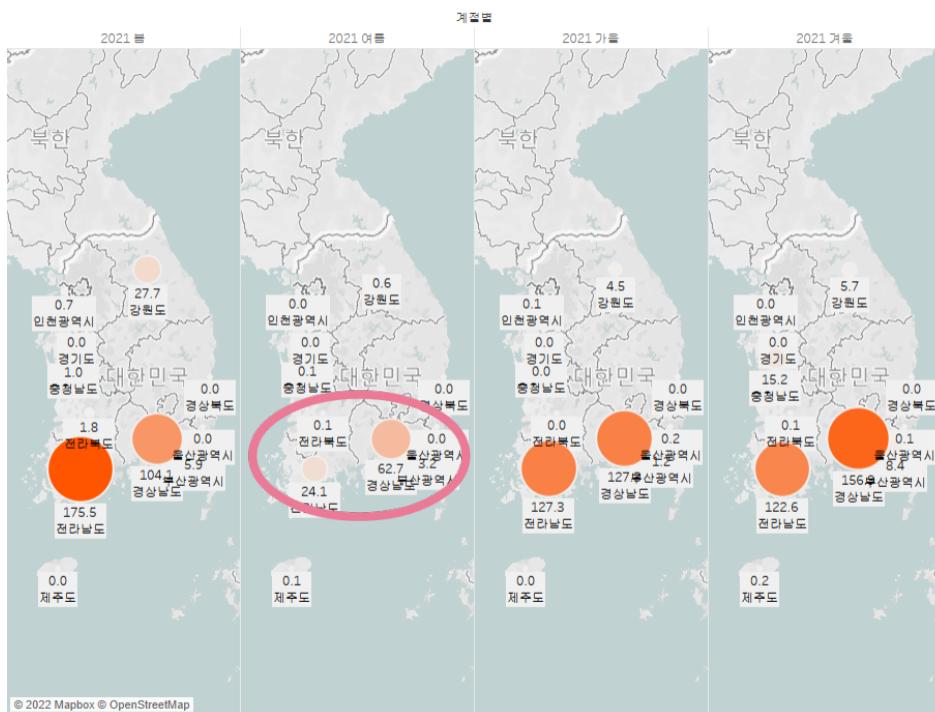
2019년 계절별 감성동 어획량을 확인해 대체적으로 비슷한 어획량을 보여주고 있습니다. 자세히 보면 겨울철과 봄철에 생산량이 더 많은 경향을 나타내고 있습니다. 20년도와 21년도 자료도 함께 보겠습니다.

2020년 감성돔 생산량(지도)



2020년 감성돔 계절별 어획량 추이 4계절 비슷한 양상을 보이지만 봄에 더 많이 잡히는 것을 확인할 수 있습니다.

2021년 감성돔 생산량(지도)

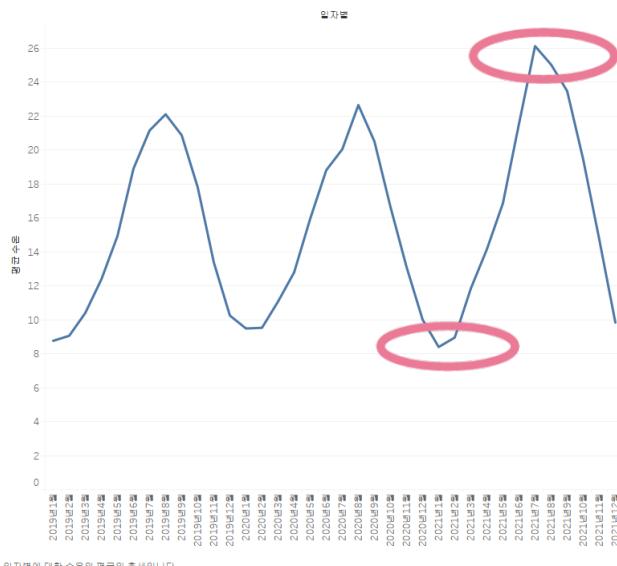


2021년 감성돔 계절별 어획량 추이는 여름의 생산량을 제외하고 비슷한 양상을 유지하고 있습니다.
마찬가지로 봄과 겨울에 많은 어획량을 보여주고 있습니다.

수온의 영향을 비교적 적게 받음을 추론해 볼 수 있습니다.

구체적인 수온과의 관계를 분석해 보겠습니다.

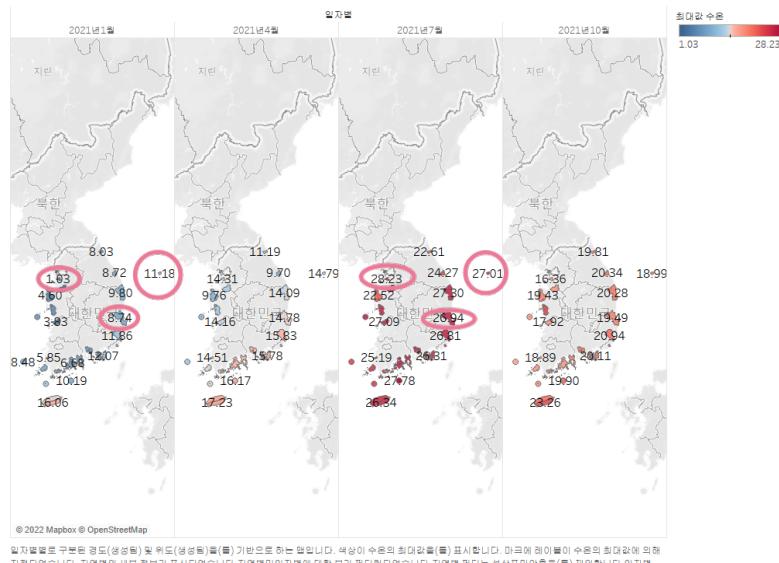
연도별 전체 수온



전국 바다의 평균 온도의 변화입니다.

- 21년도 여름과 겨울의 수온 차이가 20도 가까이 벌어지고 있음.

2021년 계절별 수온(지도)



구체적인 동해, 서해, 남해 수온의 분포입니다.

겨울철 바다 분포를 보면

서해는 1도~4도 정도로 가장 낮고, 동해는 8도~9도, 남해는 10도~16도를 기록하고 있습니다.

생산량과 수온 사이의 상관관계를 분석해보겠습니다.

1. 전처리 완료한 날해 서해 동해의 수온을 월별 평균 값을 추출해 리스트에 넣어줍니다.

```

wt, namhae_wt_list, donghae_wt_list, seohae_wt_list = [], [], [], [] # 그냥 wt는 통, 서, 남해 전체의 수온
for row in range(len(namhae_wt)):
    namhae_wt_list.append(round(namhae_wt.iloc[row,1:].values.mean(), 2))
    donghae_wt_list.append(round(donghae_wt.iloc[row,1:].values.mean(), 2))
    seohae_wt_list.append(round(seohae_wt.iloc[row,1:].values.mean(), 2))
    wt.append( round( (namhae_wt_list[row] + donghae_wt_list[row] + seohae_wt_list[row])/3 , 2 ) )
# namhae_wt

ff_corr['남해_수온'] = namhae_wt_list; yt_corr['남해_수온'] = namhae_wt_list; bp_corr['남해_수온'] = namhae_wt_list
ff_corr['동해_수온'] = donghae_wt_list; yt_corr['동해_수온'] = donghae_wt_list; bp_corr['동해_수온'] = donghae_wt_list
ff_corr['서해_수온'] = seohae_wt_list; yt_corr['서해_수온'] = seohae_wt_list; bp_corr['서해_수온'] = seohae_wt_list
ff_corr['전체_수온'] = wt; yt_corr['전체_수온'] = wt; bp_corr['전체_수온'] = wt

```

2. 월별 감성동 어획량과 수온간의 상관관계를 구해줍니다.

```

plt.figure( figsize=(14,12) )

plt.subplot( 2, 2, 1 )
# plt.figure( figsize= (6,5) )
plt.title(' 광어 생산량_평균가 상관관계 분석 ', fontsize = 20)
sns.heatmap( data = ff_corr.corr(method='pearson'), annot=True, fmt='.2f', linewidths=.5, cmap='Blues', annot_kws = {'size' : 14} )

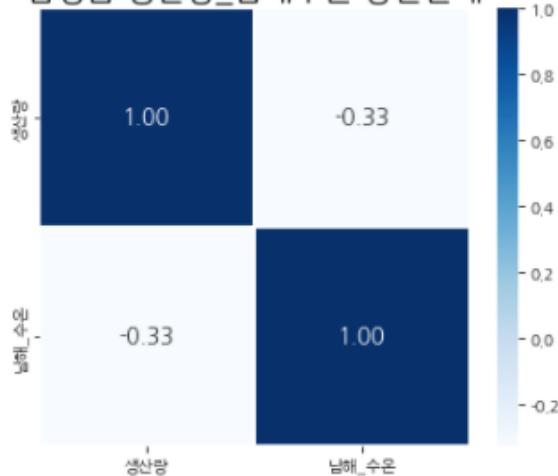
plt.subplot( 2, 2, 2 )
# plt.figure( figsize= (6,5) )
plt.title(' 방어 생산량_평균가 상관관계 분석 ', fontsize = 20)
sns.heatmap( data = yt_corr.corr(method='pearson'), annot=True, fmt='.2f', linewidths=.5, cmap='Blues', annot_kws = {'size' : 14} )

plt.subplot( 2, 2, 3 )
# plt.figure( figsize= (6,5) )
plt.title(' 감성동 생산량_평균가 상관관계 분석 ', fontsize = 20)
sns.heatmap( data = bp_corr.corr(method='pearson'), annot=True, fmt='.2f', linewidths=.5, cmap='Blues', annot_kws = {'size' : 14} )

```

감성동 생산량- 날해수온 사이의 상관계수입니다.

감성동 생산량_남해수온 상관관계

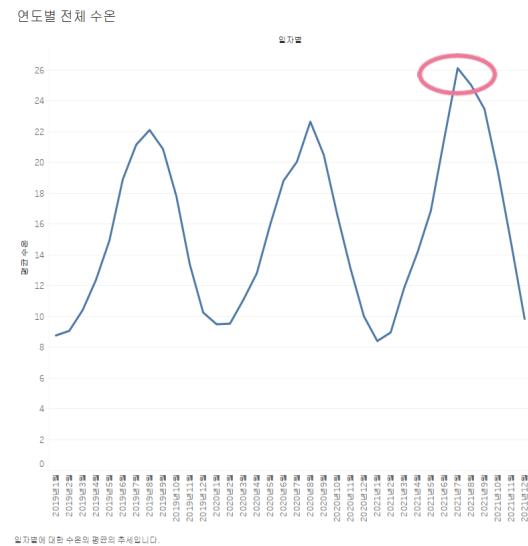
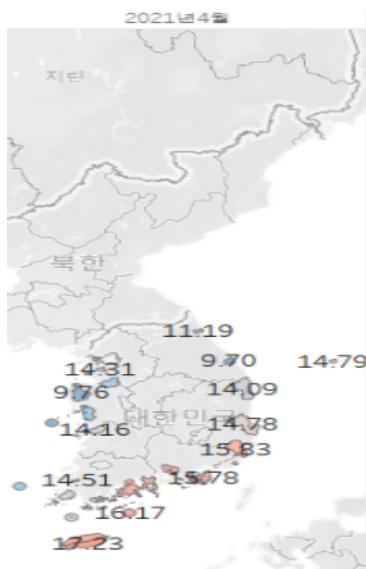
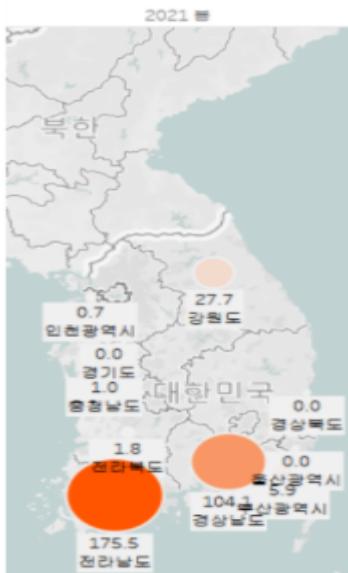


감성동 생산량 - 수온 사이의 상관계수 -0.33 에 대해 분석해 보면

수온이 줄어들 때 생산량이 약한 정도로 상승함

대체적으로 4개월 비슷한 생산량을 보이지만, 봄철과 겨울철에 더욱 많은 생산량을 보이는 현상을 잘 설명합니다.

- 감성동의 경우 13~16도의 수온에서 활동한다는 사실에 비추어 볼 때 봄철에 14도를 기록하는 전라남도에서 가장 많이 잡히는 것을 잘 설명합니다. 뿐만 아니라 21년도 여름 지난치게 올랐던 수온의 영향으로 어획량이 줄어든 것 또한 잘 설명하는 것으로 보입니다



생산량과 가격 사이의 관계가 있는지 분석해 보면

1. 먼저 위판정보 데이터에 포함된 감성동 가격 데이터를 추출하여 월별 평균가를 구한 뒤 리스트화 합니다.

```
# `수산물_위판정보_전처리완료`에서 감성동만 뽑아옴  
bp_price = price[price['수산물표준코드명'] == '감성동']  
bp_price.reset_index(drop=True, inplace=True)  
  
for row in range(len(bp_price)):  
    bp_price.loc[row, '위판일자'] = bp_price.loc[row, '위판일자'].replace('-%01', '-').replace("-", ".")  
  
# 19년도부터 21년도까지 총 36개월 평균가 구해서 list로  
bp_price_list = []  
for y in range(3):  
    for m in range(1,13):  
        month_price = []  
        for d in range(1,32):  
            if bp_price[bp_price['위판일자'] == f'{2019+y} .{m} .{d}']['평균가'].values.mean() >= 0 :  
                month_price.append(bp_price[bp_price['위판일자'] == f'{2019+y} .{m} .{d}']['평균가'].values.mean())  
        bp_price_list.append( round( sum(month_price) / len(month_price), 2 ) )
```

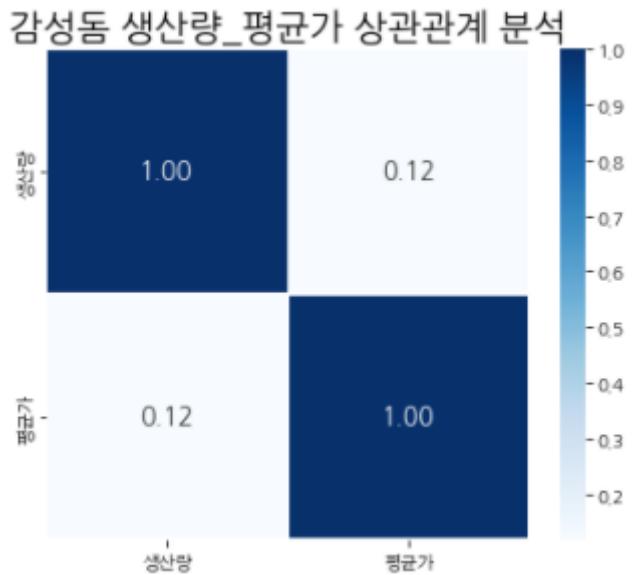
2. 다음으로 생산량 데이터에서 감성동 데이터만 추출후 컬럼명을 변경해줍니다.

```
❶ # 해면어업 품종별 생산량에서 감성동만 뽑음  
bp_production = production[production['품종별'] == '감성동'].reset_index(drop=True)  
  
# 계산하기 편하게하기 위해 컬럼명 바꿔줌  
for col in range(1, len(bp_production.columns)) :  
    old_col = bp_production.columns[col]  
    new_col = bp_production.columns[col].replace(".0","",).replace("월","",).replace(" ", "")  
    bp_production.rename(columns={ old_col : new_col }, inplace=True)
```

3. 19년부터 21년까지의 월별 평균가격과 생산량 사이의 상관관계를 구합니다.

```
# 감성동  
bp_corr = {'일자': list(bp_production.columns[1:]), '생산량': list(bp_production.iloc[0,1:]), '평균가': bp_price_list}  
bp_corr = pd.DataFrame(bp_corr)  
# bp_corr  
  
plt.figure( figsize=(5,4) )  
plt.title(' 감성동 생산량_평균가 상관관계 분석 ', fontsize=17)  
sns.heatmap( data = bp_corr.corr(method='pearson'), annot=True, fmt='.2f', linewidths=.5, cmap='Blues', annot_kws = {'size' : 14})
```

감성동 생산량과 평균가 사이의 상관관계입니다.

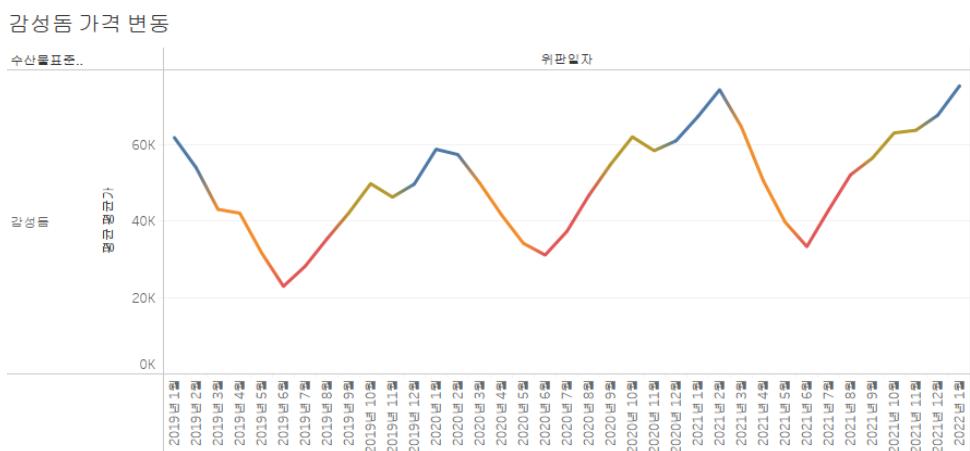


생산량과 가격 사이의 상관계수 0.12에 대해 분석해보면

생산량과 가격 사이에 **관련이 없음**을 의미합니다.

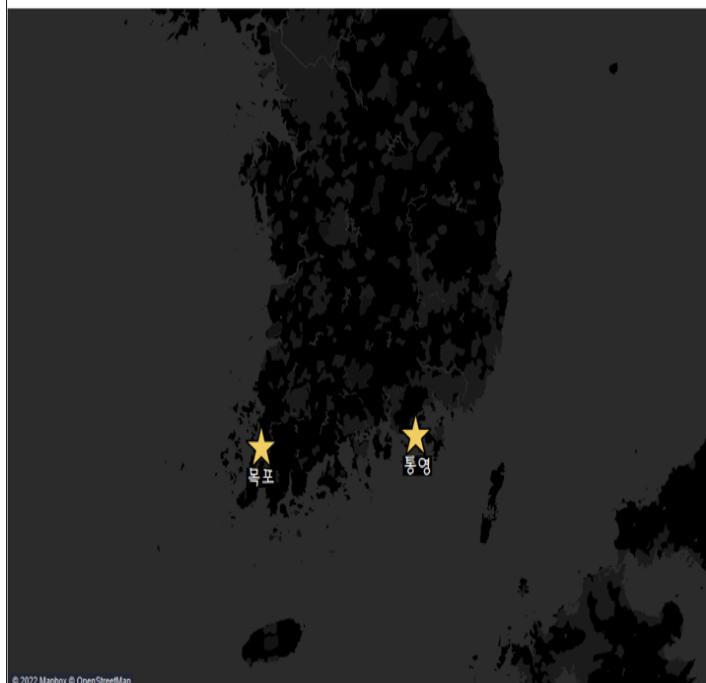
생산량 증가로 인해 가격 하방 압력이 발생하나, 감성동의 제철 또한 겨울이므로 높은 수요가 오히려 가격을 상승시키는 것 같습니다.

결국 생산량 보다는, **소비자들의 수요에 영향을 받는 것으로 보여지며 그 외에도 3월부터 5월 사이에 있는 산란기 및 금어기 등의 요소가 영향을 주는 것으로 추측됩니다.**



결론적으로 감성돔을 즐기고 싶으신 분은 거울철, 전라도 목포 또는 경상남도를 통영을 방문하신다면 다른 지역보다 저렴하고 맛있게 즐길 수 있을것입니다.

감성돔 추천 지역



© 2022 Mapbox © OpenStreetMap

감성돔(12월)

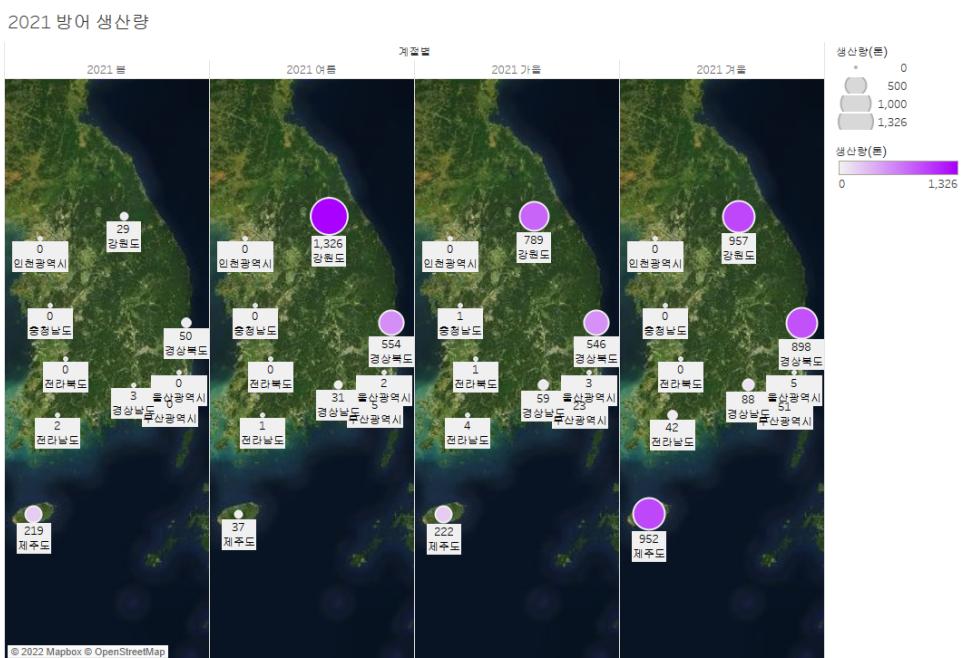


4. 방어

3개년 방어 가격 변동 그래프입니다.



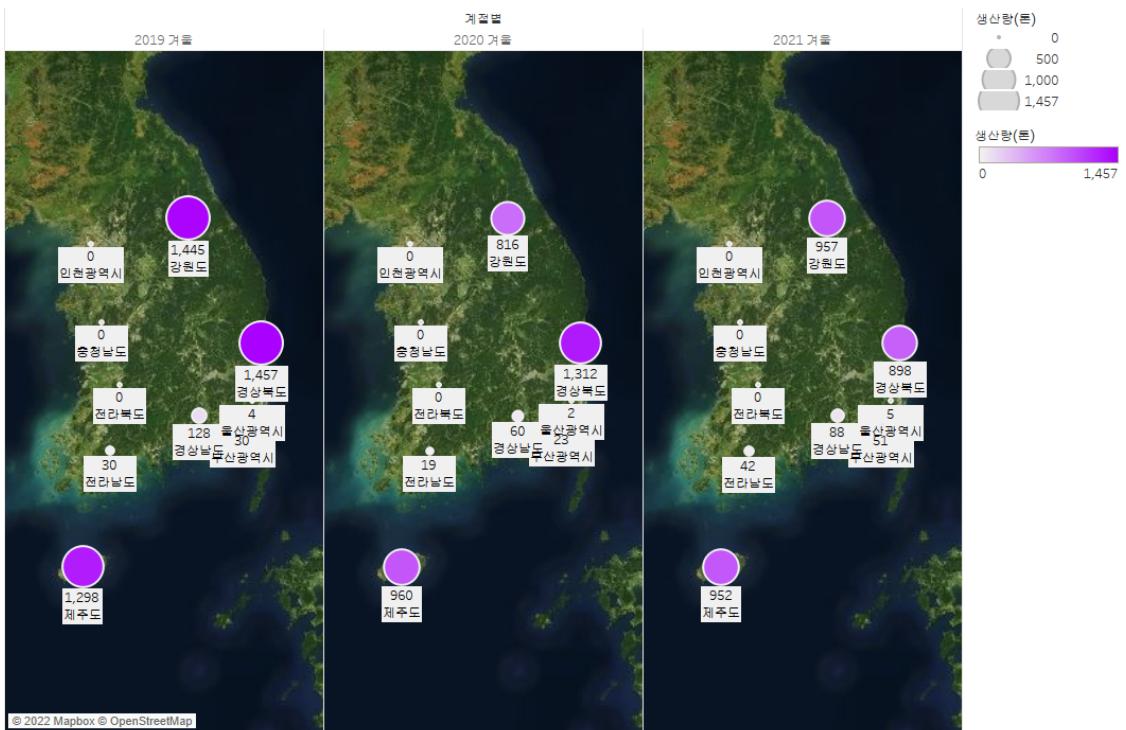
21년 계절별 방어 생산량입니다.



- 방어의 경우 거울철의 어획량이 높게 나타나고 있음을 확인했습니다. 지도상에 표시된 경상북도는 포항을 의미하고 . 강원도는 삼척. 동해시 등을 포함하고 있습니다.
 - 거울철의 경우 제주. 포항. 강원도 까지 전제적으로 생산량이 증가하는 반면 . 여름철에는 상대적으로 고위도인 강원도의 어획량이 많은 것으로 보여집니다.

3개년 겨울철 방어 생산량만 주목해보겠습니다.

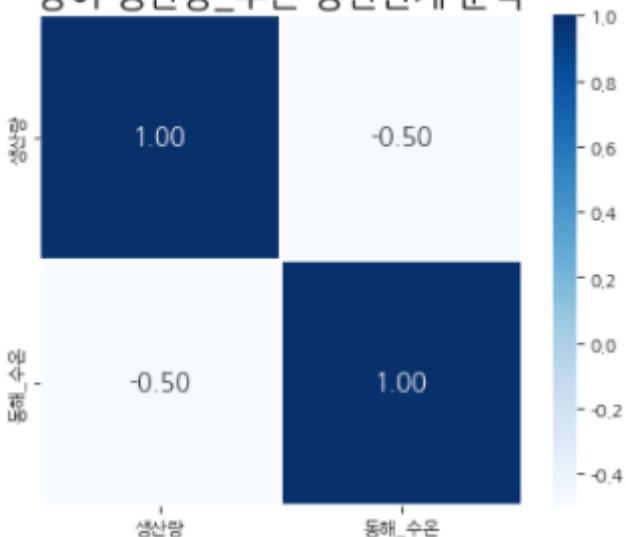
겨울 방어 생산량



©2022 Mapbox ©OpenStreetMap
계절별로 구분된 경도(성능률) 및 위도(성능률)를 기반으로 하는 맵입니다. 색상이 생산량(톤)의 합계를 표시합니다. 크기에서 생산량(톤)의 합계를 표시합니다. 마크에 레이블이 생산량(톤)의 합계 및 행정구역별로 지정되었습니다. 2019 겨울, 2020 겨울 및 2021 겨울을 유지하는 계절별로 대한 뷰가 팔려왔습니다.

- 최근 3개년 겨울철 방어 어획량을 보면 즐어드는 추세가 나타나고 있습니다.
- 이러한 원인을 수온으로 가정하고 수온과 생산량 간의 상관관계를 분석해보겠습니다.

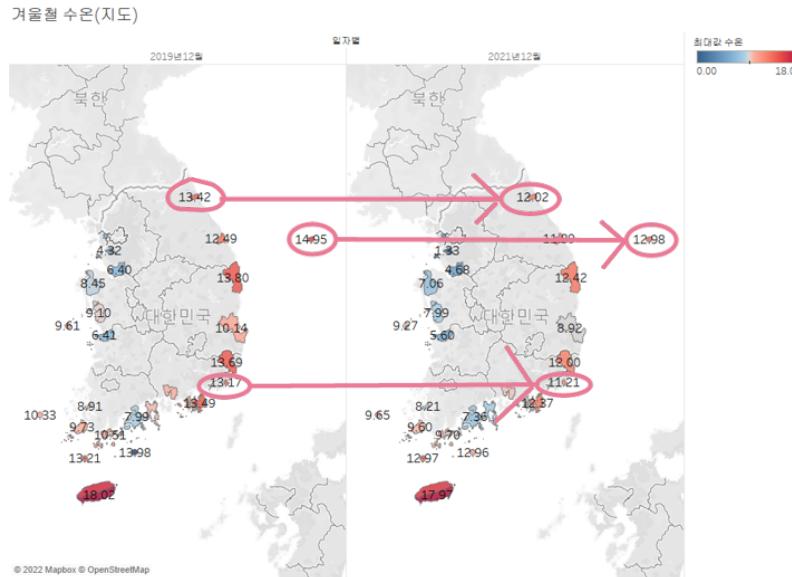
방어 생산량_수온 상관관계 분석



방어 생산량과 동해 수온 상관관계 분석 결과입니다.

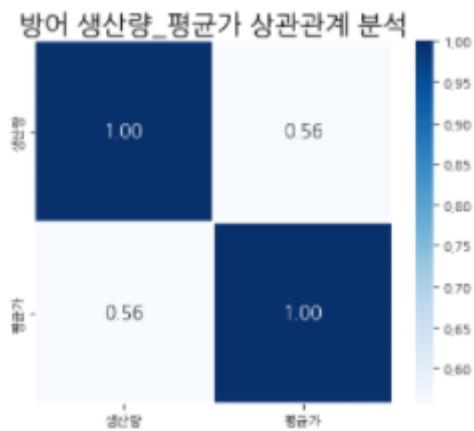
- 상관계수는 **-0.5**로, 수온이 감소하면 생산량이 증가함을 의미합니다.
- 19년과 21년의 온도 차이를 통해 생산량 변화를 확인해 보겠습니다.

19년도와 21년도 겨울철 수온의 분포입니다.



방어의 어획량이 많은 동해의 경우 작게는 1도에서 크게는 2도까지 감소한 경향을 보였습니다.
여름부터 겨울로 넘어가면서 수온의 온도가 감소하는 경우 어획량은 늘어나지만, 방어의 평균 활동 온도인 16도보다
지나치게 낮아지는 경우 전년대비 생산량이 더욱 감소하는 것으로 보여집니다.

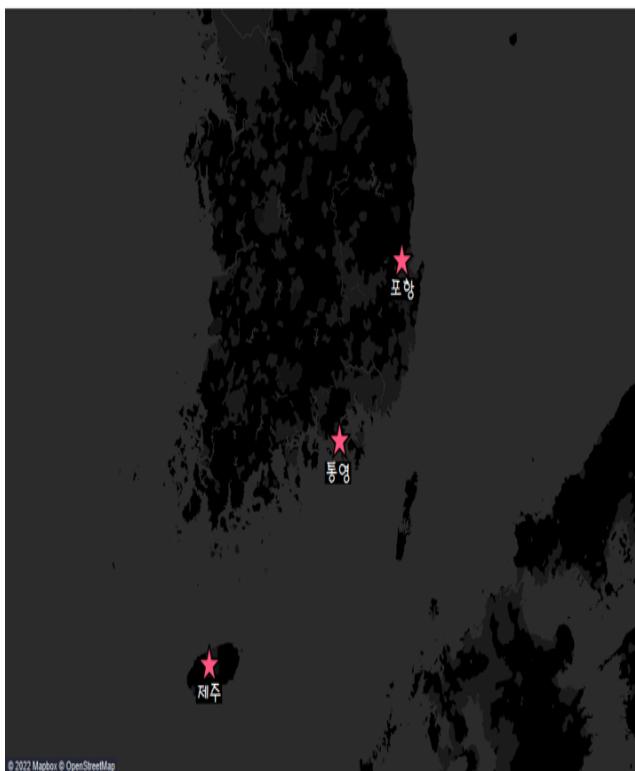
방어의 생산량과 평균가 사이의 상관관계입니다.



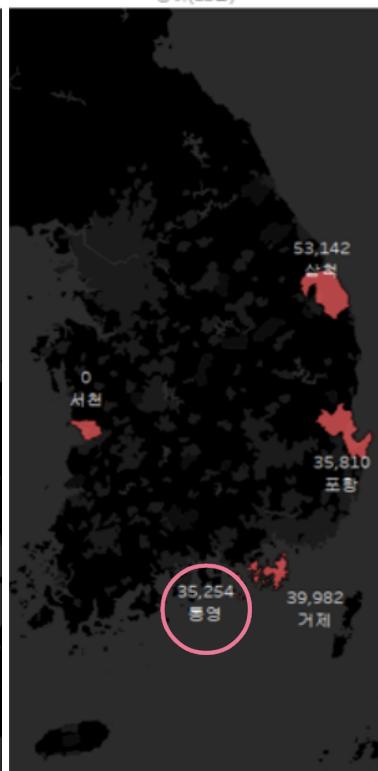
0.56에 대해 분석해보면 생산량이 증가할 때 가격도 증가
여름철 반동 및 겨울철 급등을 잘 설명

**결론적으로 방어의 경우 겨울철에도 16도의 수온을 일정하게 유지하고 있는 제주도와
경상도의 통영 또는 포항을 추천드립니다.
강원도는 비싼 경향이 있습니다.**

방어 추천 지역



방어(12월)



© 2022 Mapbox © OpenStreetMap

5. 광어

광어 가격 변동



2021 광어 생산량



게릴라법은 구분된 경도(성장률) 및 위도(성장률) 기반으로 하는 법입니다. 석상이 생산량(톤)의 합계율(률) 표시합니다. 크기에서 생산량(톤)의 합계율(률) 표시합니다. 미크로 웨이블이 생산량(톤)의 합계율(률)에 의해 지정되었습니다. 청중구역별의 세부 정보가 표시되었습니다. 2021 가을, 2021 겨울, 2021 봄 및 2021 겨울(률) 유지하는 게릴라에 대한 유가 할당되었습니다.

- 2021년도 광어 계절별 어획량을 보면 봄과 여름에는 충청도에서 가을과 겨울에는 강원도의 생산량이 증가함을 볼 수 있습니다. 뚜렷한 차별 요인이 보이지 않아 추가적으로 19년도 생산량을 확인해 보았습니다.

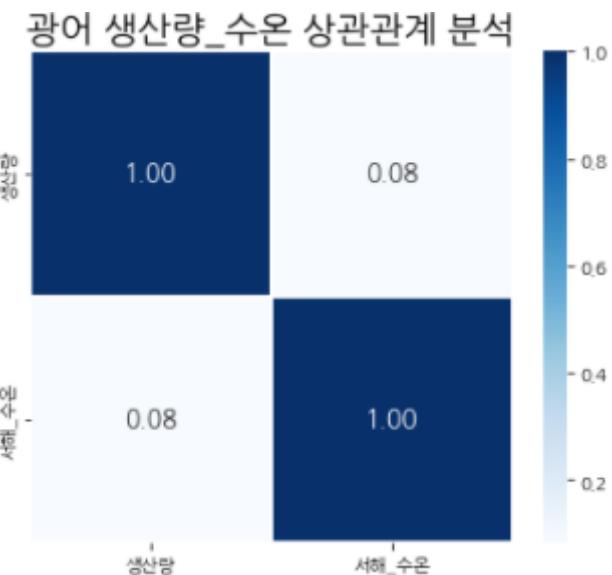
2019 광어 생산량



게릴라법은 구분된 경도(성장률) 및 위도(성장률) 기반으로 하는 법입니다. 석상이 생산량(톤)의 합계율(률) 표시합니다. 크기에서 생산량(톤)의 합계율(률) 표시합니다. 미크로 웨이블이 생산량(톤)의 합계율(률)에 의해 지정되었습니다. 청중구역별의 세부 정보가 표시되었습니다. 2019 가을, 2019 겨울, 2019 봄 및 2019 겨울(률) 유지하는 게릴라에 대한 유가 할당되었습니다.

- 19년도 생산량 또한 마찬가지로 광어의 어획은 전국적으로 고르게 분포되어 있습니다. 동일하게 여름에 충청남도 어획량이 급증하는 현상을 보이고 있습니다.

마찬가지로 수온과의 상관관계 분석을 수행해 보았습니다.



- 광어의 경우 수온과의 상관관계는 **0.08** 수준으로 관련이 없었습니다.
- 추가적으로 확인한 결과 **광어의 서식지는 수심 200m 깊이**으로 최저 기온과 최대 기온의 차이가 2도 밖에 나지 않기에 **수온의 영향을 받지 않는 것으로 판단됩니다.**
 - 방어의 경우 6~20m 깊이의 경우 50m의 수심에서 활동합니다. 방어의 경우 수온과의 상관계수는 -0.5였고, 깊이의 경우 -0.3 이었습니다. 따라서 방어 깊이와 방어 순으로 수온의 영향을 크게 받는 것으로 보여집니다.
- 충청도의 급격한 어획량 증가는 **다른 요인**이 있다고 **판단**하여 광어의 특성에 대해 조사했습니다.
 - 그 결과 광어는 **산란**을 할 시기인 3월~5월 사이, 수심 20~50m의 조수 흐름이 좋은 모래바닥 및 암초지대에서 수행한다는 사실을 알게되었습니다.
 - 충청남도 서천의 지형과 여름철 이라는 시기를 고려할 때 현상을 잘 설명한다고 보여집니다.
 - 충청남도 서천의 경우 여름철 자연산 대광어의 가격이 19000원으로 형성 되어 있었습니다.

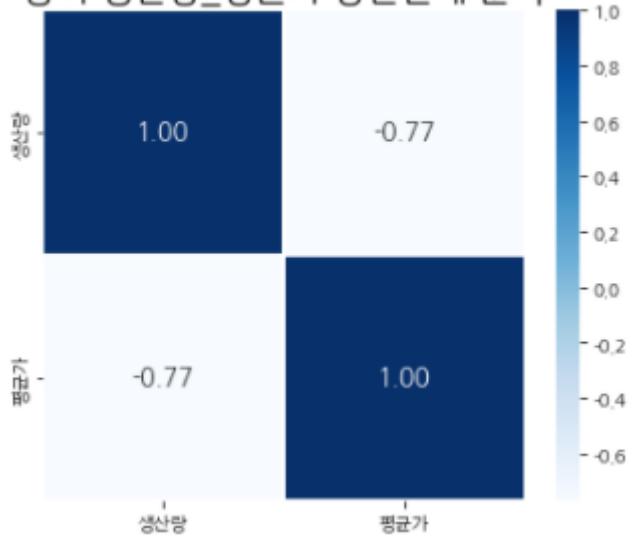


2018년 5월 서천자연산광어 도미축제 행사장에서 광어와 도미의 표장 판매 가격

- 이에 생산량과 가격이 상관관계가 정밀로 있는지 확인해 보았습니다.

광어 생산량과 평균가 사이의 상관관계입니다.

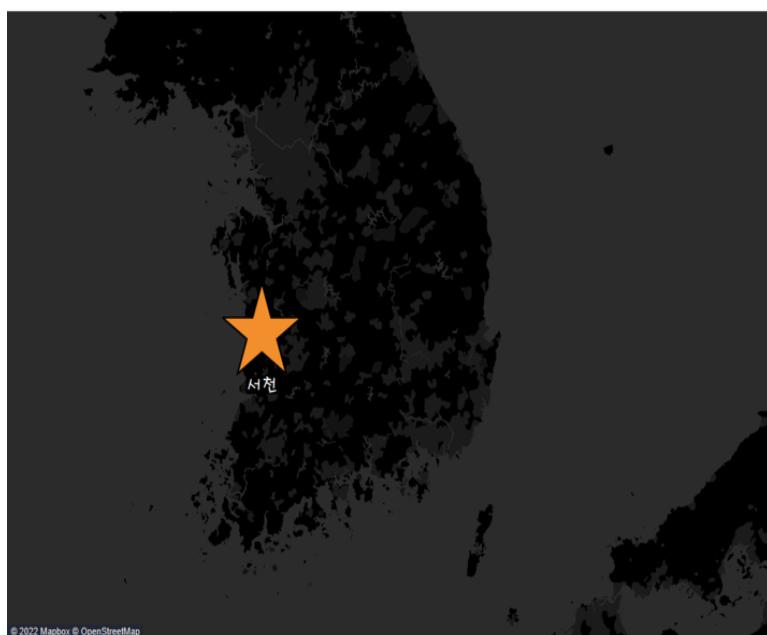
광어 생산량_평균가 상관관계 분석



생산량과 평균가격의 상관관계는 **-0.77**로 나타나고 있습니다. 이는 생산량이 증가하면 가격이 떨어짐을 의미합니다.
시계절 전국적으로 꾸준히 잡하고 있는 광어이기에 급격한 생산량 증가로 가격이 떨어지는 것으로 보여집니다.

따라서 자연산 대광어를 노리신다면 어틀철 충남 서천을 찾으시길 권해드립니다.
추가적으로 충남 서천에서 열리는 광어 축제도 참여해 맛과 즐거움 모두 찾으시길 바랍니다.

광어 추천 지역





(20년도 이후 코로나로 취소 되었지만 올해는 개최되길 바랍니다)

이상입니다.