**# 패키지 import 해오기**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import requests

import json

import time

from pandas.io.json import json\_normalize

from bs4 import BeautifulSoup

from scipy import stats

**# 데이터 파일 불러오기**

mobile\_df = pd.read\_csv('./mobile\_data.csv')

internet\_df = pd.read\_csv('./internet\_data.csv')

corona\_df = pd.read\_csv('./covid\_month.csv')

total\_internet\_mobile\_df = pd.read\_csv(‘./total\_data.csv’)

corona\_age\_df = pd.read\_csv('./covid\_age.csv')

click\_data\_total = pd.read\_csv('./click\_data\_pc.csv')

**# 변수 지정**

##모바일쇼핑 데이터와 코로나 월별 확진자 데이터 merge

mobile\_corona = mobile\_df.merge(corona\_df, on='date')

##mobile\_corona 데이터와 코로나 연령대별 확진자 데이터 merge

age\_mobile\_corona = mobile\_corona.merge(covid\_age, on='date')

##인터넷쇼핑 데이터와 코로나 월별 확진자 데이터 merge

internet\_corona = internet\_df.merge(corona\_df, on='date')

##internet\_corona 데이터와 코로나 연령대별 확진자 데이터 merge

age\_internet\_corona = internet\_corona.merge(covid\_age, on='date')

##인터넷쇼핑 상위 3개 카테고리와 하위 3개 카테고리(electronic, grocery, kids, travel, culture\_ls, office) 조회수 데이터와 연령대별 확진자 데이터 merge

click\_data\_total\_age = corona\_age\_df.merge(click\_data\_total, on='date')

*#이따 코멘트 달기*

total\_shopping\_corona = total\_internet\_mobile\_df.merge(corona\_age\_df, on='date')

**# 데이터 전처리**

##모바일쇼핑 데이터 날짜 인덱스 중 ‘p)’ 포함된 인덱스 수정하기

mobile\_df['date'] = mobile\_df['date'].replace(['2021. 07 p)'],'2021. 07')

mobile\_df['date'] = mobile\_df['date'].replace(['2021. 06 p)'],'2021. 06')

##인터넷쇼핑 데이터 날짜 인덱스 중 ‘p)’ 포함된 인덱스 수정하기

internet\_df['date'] = internet\_df['date'].replace(['2021. 07 p)'],'2021. 07')

internet\_df['date'] = internet\_df['date'].replace(['2021. 06 p)'],'2021. 06')

##모바일쇼핑 데이터와 코로나 월별 확진자 데이터를 merge한 데이터에서 일부 컬럼명 변경하기

mobile\_corona.rename(columns = {'total\_x' : 'trans\_amnt\_total'}, inplace = True)

mobile\_corona.rename(columns = {'total\_y' : 'positive\_total'}, inplace = True)

mobile\_corona.rename(columns = {'beverage' : 'food\_service'}, inplace = True)

age\_mobile\_corona.rename(columns = {'total' : 'age\_total'}, inplace = True)

##인터넷쇼핑 데이터와 코로나 월별 확진자 데이터를 merge한 데이터에서 일부 컬럼명 변경하기

internet\_corona.rename(columns = {'total\_x' : 'trans\_amnt\_total'}, inplace = True)

internet\_corona.rename(columns = {'total\_y' : 'positive\_total'}, inplace = True)

internet\_corona.rename(columns = {'beverage' : 'food\_service'}, inplace = True)

age\_internet\_corona.rename(columns = {'total' : 'age\_total'}, inplace = True)

#모바일쇼핑+인터넷쇼핑과 코로나 연령별 확진자 데이터를 merge한 데이터에서 일부 컬럼명 변경하기

total\_shopping\_corona.rename(columns = {'total\_x' : 'trans\_amnt\_total'}, inplace = True)

total\_shopping\_corona.rename(columns = {'total\_y' : 'positive\_total'}, inplace = True)

**# 데이터 시각화**

**## 월별 '합계'카테고리에 대한 인터넷과 모바일 간의 거래액 비교**

plt.figure(figsize=(12,6))

plt.title('Mobile vs. Internet shopping')

plt.xlabel('Date')

plt.ylabel('Total Transaction amount')

plt.bar(mobile\_df['date'], mobile\_df['total'], label='mobile')

plt.bar(internet\_df['date'], internet\_df['total'], label='internet')

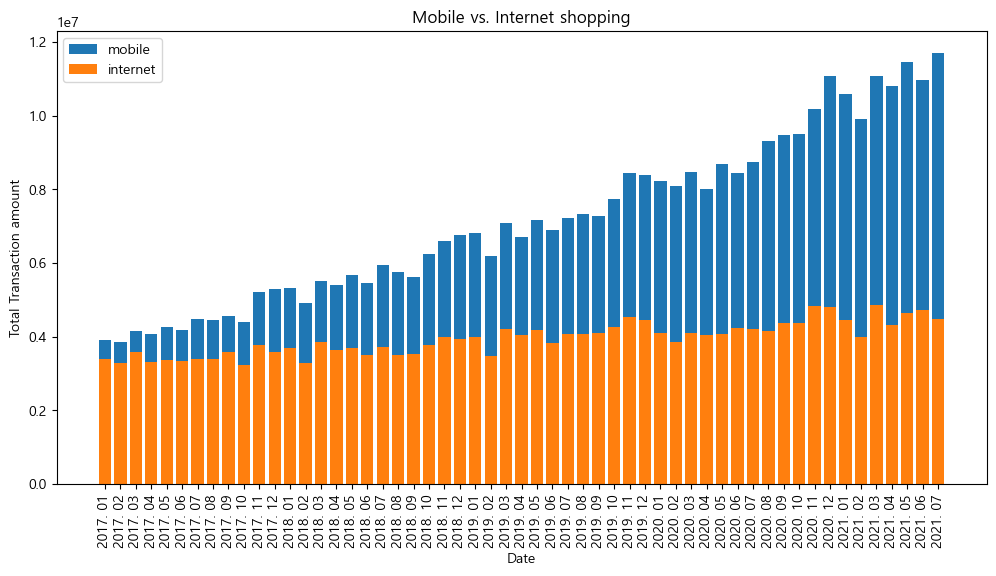
plt.xticks(rotation=90)

plt.legend()

plt.show()

> 발표 intro 중요 그래프

* 기본전제설명 추가 :

****

**## 모바일 쇼핑 증감액, 증감률 가로막대 그래프 그리기**

# 모바일쇼핑 거래액 데이터의 '시점'을 인덱스화

mobile\_df\_01 = mobile\_df.set\_index('date')

# 2020년 3월 항목의 row 추출

mobile\_march\_2020 = mobile\_df\_01[mobile\_df\_01.index=='2020. 03']

# 2019년 3월 항목의 row 추출

mobile\_march\_2019 = mobile\_df\_01[mobile\_df\_01.index=='2019. 03']

# 증감액 구하기 (2020, 2019년 3월)

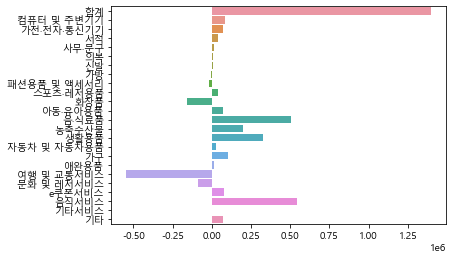
mobile\_march\_subtract = np.subtract(mobile\_march\_2020, mobile\_march\_2019)

# 증감률 구하기 (2020, 2019년 3월)

mobile\_march\_rate = np.divide(mobile\_march\_subtract, mobile\_march\_2019) \* 100

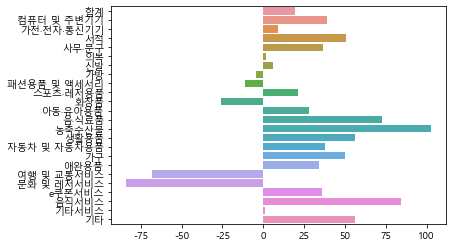
# 증감액 그래프 (모바일쇼핑 가로막대그래프)

sns.barplot(data=mobile\_march\_subtract, orient='h')



# 증감률 그래프 (모바일쇼핑 가로막대그래프)

sns.barplot(data=mobile\_march\_rate, orient='h')



**## 인터넷 쇼핑 증감액, 증감률 가로막대 그래프 그리기**

# 인터넷쇼핑 거래액 데이터의 'date'를 인덱스화

internet\_df\_01 = internet\_df.set\_index('date')

# 2020년 3월 항목의 row 추출 (인터넷)

internet\_march\_2020 = internet\_df\_01[internet\_df\_01.index=='2020. 03']

# 2019년 3월 항목의 row 추출 (인터넷)

internet\_march\_2019 = internet\_df\_01[internet\_df\_01.index=='2019. 03']

# 증감액 구하기 (2020, 2019년 3월) (인터넷)

internet\_march\_subtract = np.subtract(internet\_march\_2020, internet\_march\_2019)

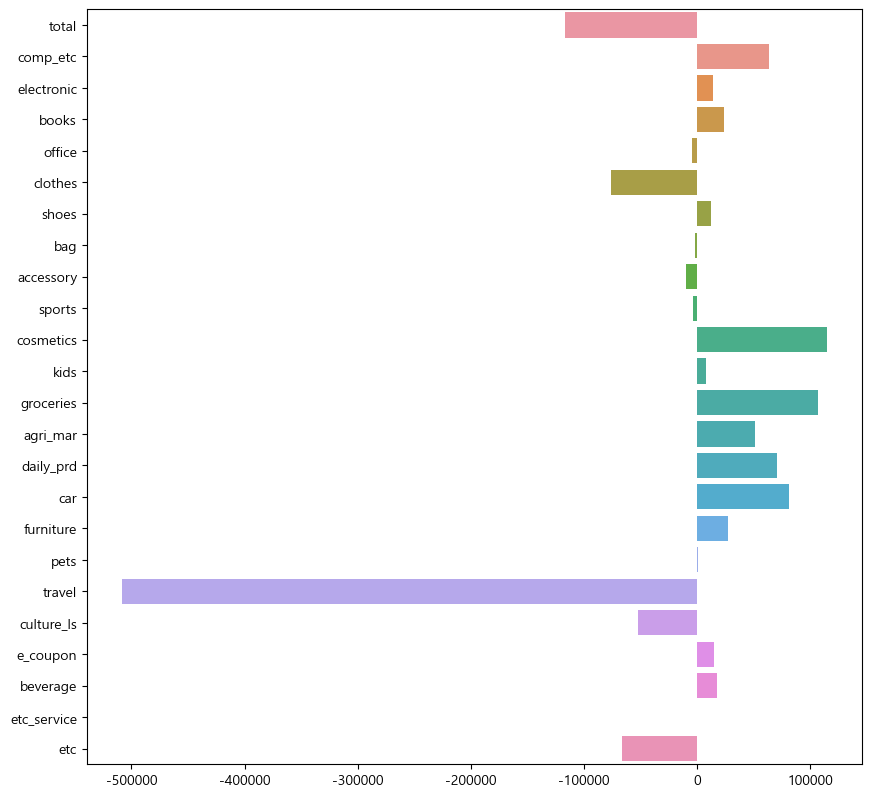
# 증감률 구하기 (2020, 2019년 3월) (인터넷)

internet\_march\_rate = np.divide(internet\_march\_subtract, internet\_march\_2019) \* 100

# 증감액 그래프 (인터넷쇼핑 가로막대그래프)

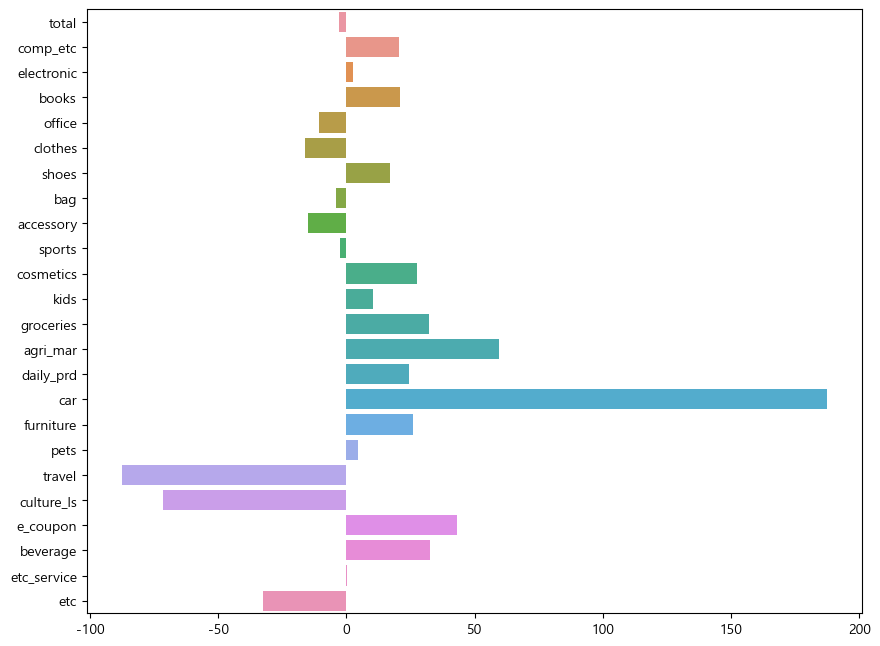
fig=plt.figure(figsize=(10,10))

sns.barplot(data=internet\_march\_subtract, orient='h')

****

# 증감률 그래프 (인터넷쇼핑 가로막대그래프)

sns.barplot(data=internet\_march\_rate, orient='h')



# 모바일 거래액 데이터 기초통계량 확인

mobile\_df.describe()

**#모바일 쇼핑 거래액 데이터 boxplot**

#drop을 할 때 axis=1을 해야 drop이 되는 이유

#axis=1: 열 날리기

#axis=0: 행 날리기 (default)

plt.title('Box plot of mobile\_df')

plt.xlabel('Industries')

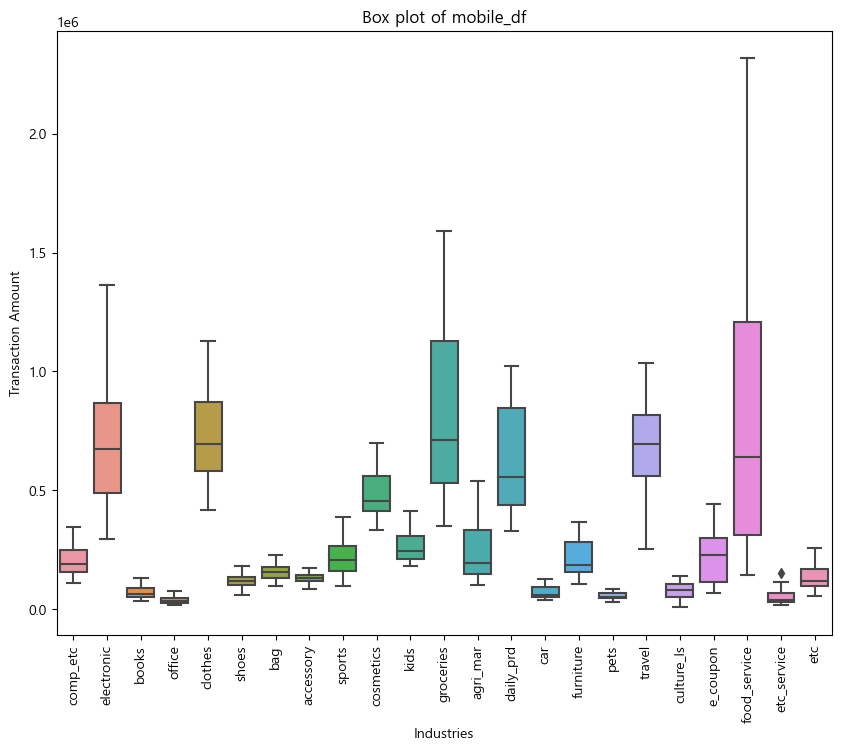
plt.ylabel('Transaction Amount')

sns.boxplot(data=mobile\_df.drop(['total'], axis=1))

plt.xticks(rotation=90)

plt.show()

#outlier가 etc\_service(기타 서비스)에만 있음



#etc\_service 거래액 (월별)

plt.figure(figsize=(12,6))

plt.bar(mobile\_df['date'], mobile\_df['etc\_service'], label='mobile')

plt.xticks(rotation=90)

plt.legend()

plt.show()

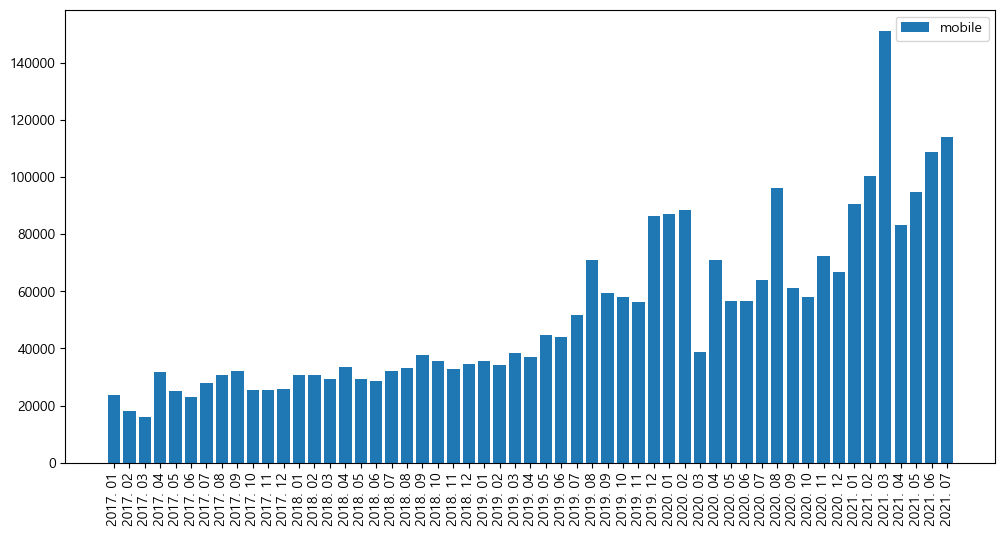
#mobile\_df.describe()로 ‘etc\_service’의 max=150986 인 걸 알아냄

#max값이 있는 날짜 = 2021.03

#2021.03에서 거래액이 급증한 이유:

#(추측) 항목 중 렌탈 서비스가 차지하는 비중이 제일 클 것이다. (차박을 위한 렌탈이 많지 않았을까)

#검색어 조회로 etc\_service에 해당하는 키워드를 비교한 후, 이사 키워드가 제일 많음



#인터넷 vs 모바일 etc\_service 거래액 (월별) 비교

plt.figure(figsize=(12,6))

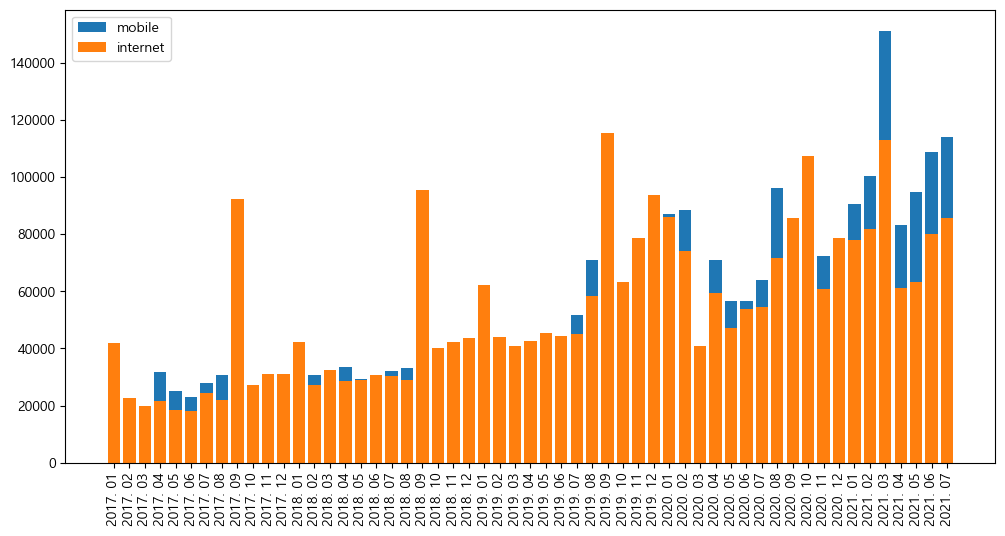
plt.bar(mobile\_df['date'], mobile\_df['etc\_service'], label='mobile')

plt.bar(internet\_df['date'], internet\_df['etc\_service'], label='internet')

plt.xticks(rotation=90)

plt.legend()

plt.show()



**#인터넷 쇼핑 거래액 데이터**

#drop을 할 때 axis=1을 해야 drop이 되는 이유

#axis=1: 열 날리기

#axis=0: 행 날리기 (default)

sns.boxplot(data=internet\_df.drop(['total'], axis=1))

plt.xticks(rotation=90)

plt.show()

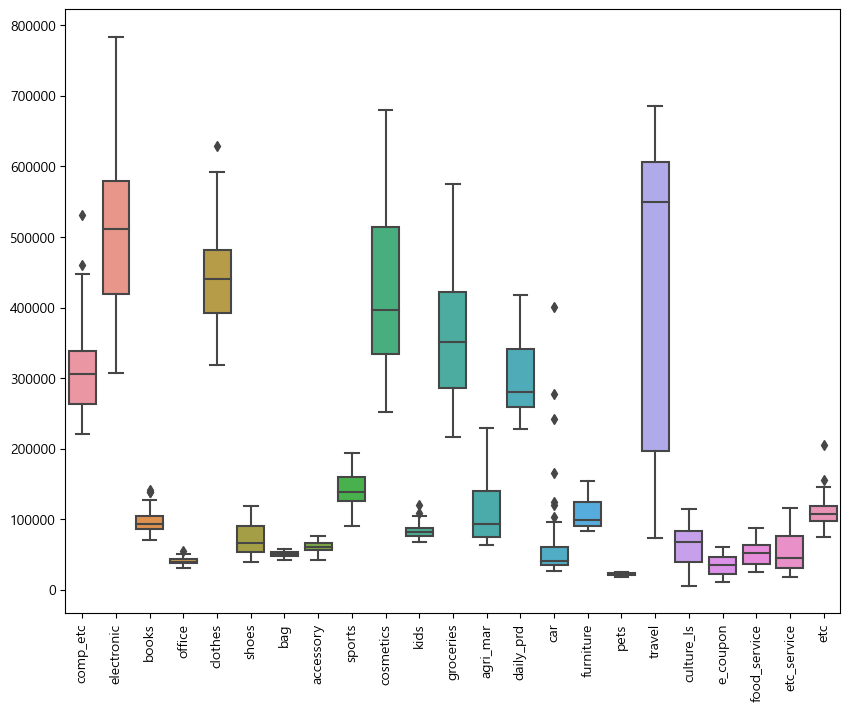
#거래액이 가장 많은 산업군: electronic(전자기기)

#거래액의 폭이 가장 좁은 산업군: pets(반려동물용품)

#outlier가 가장 많은 산업군: car(자동차 및 자동차용품)

#outlier가 있는 산업군(=거래액의 대폭 상승이 있었던 산업군): comp\_etc, books, office,

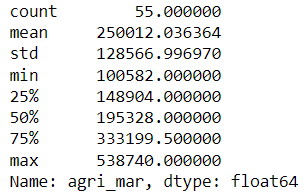
#clothes, kids, car, etc



# 1-1. 모바일쇼핑 증감률 1위 산업군: agri\_mar(농축산물)

# 중앙값 찾기

mobile\_df['agri\_mar'].describe()

****

# 1-1. 중앙값을 원 값들에서 빼기

mobile\_df['agri\_mar'] - 195328

# 중앙값 뺀 값이 0인 날짜 찾기

mobile\_df[(mobile\_df['agri\_mar']-195328 )== 0]

#결과: 2019년 4월

#잔차계산 & normalization 작업

#중앙값=195328

plt.figure(figsize=(12, 15))

**#(모바일) 잔차와 normalization 과정들을 나타낸 subplot**

#agri\_mar의 잔차를 나타낸 바 그래프

plt.subplot(221)

plt.title('Residual Bar plot of agri\_mar')

plt.bar(mobile\_df['date'], mobile\_df['agri\_mar']-195328, label='mobile')

plt.xticks(rotation=90)

#agri\_mar의 잔차를 나타낸 삼전도(scatter plot)

plt.subplot(222)

plt.title('Residual Scatter plot of agri\_mar with Best-fit-line')

slope, intercept, r\_value, p\_value, std\_err = stats.linregress(mobile\_df.index, mobile\_df['agri\_mar']-195328)

ax = sns.regplot(x=mobile\_df.index, y=mobile\_df['agri\_mar']-195328,

line\_kws={'label':"y={0:.1f}x+{1:.1f}".format(slope,intercept)})

ax.legend()

plt.xticks(rotation=90)

#그래프 결과를 보면 점들이 모인 형태가 하나의 선이라기보다 이차함수 곡선으로 보임

#agri\_mar의 y-transform(by using log-transformation)

plt.subplot(223)

plt.title('Log-transformed')

plt.scatter(mobile\_df['date'], np.log(mobile\_df['agri\_mar']))

plt.ylim(0,25)

plt.xticks(rotation=90)

#agri\_mar 원래의 값을 나타낸 삼전도(scatterplot)

plt.subplot(224)

plt.title('After Log-transformation with Best-fit-line')

slope, intercept, r\_value, p\_value, std\_err = stats.linregress(mobile\_df.index, np.log(mobile\_df['agri\_mar']))

ax = sns.regplot(x=mobile\_df.index, y=np.log(mobile\_df['agri\_mar']),

line\_kws={'label':"y={0:.1f}x+{1:.1f}".format(slope,intercept)})

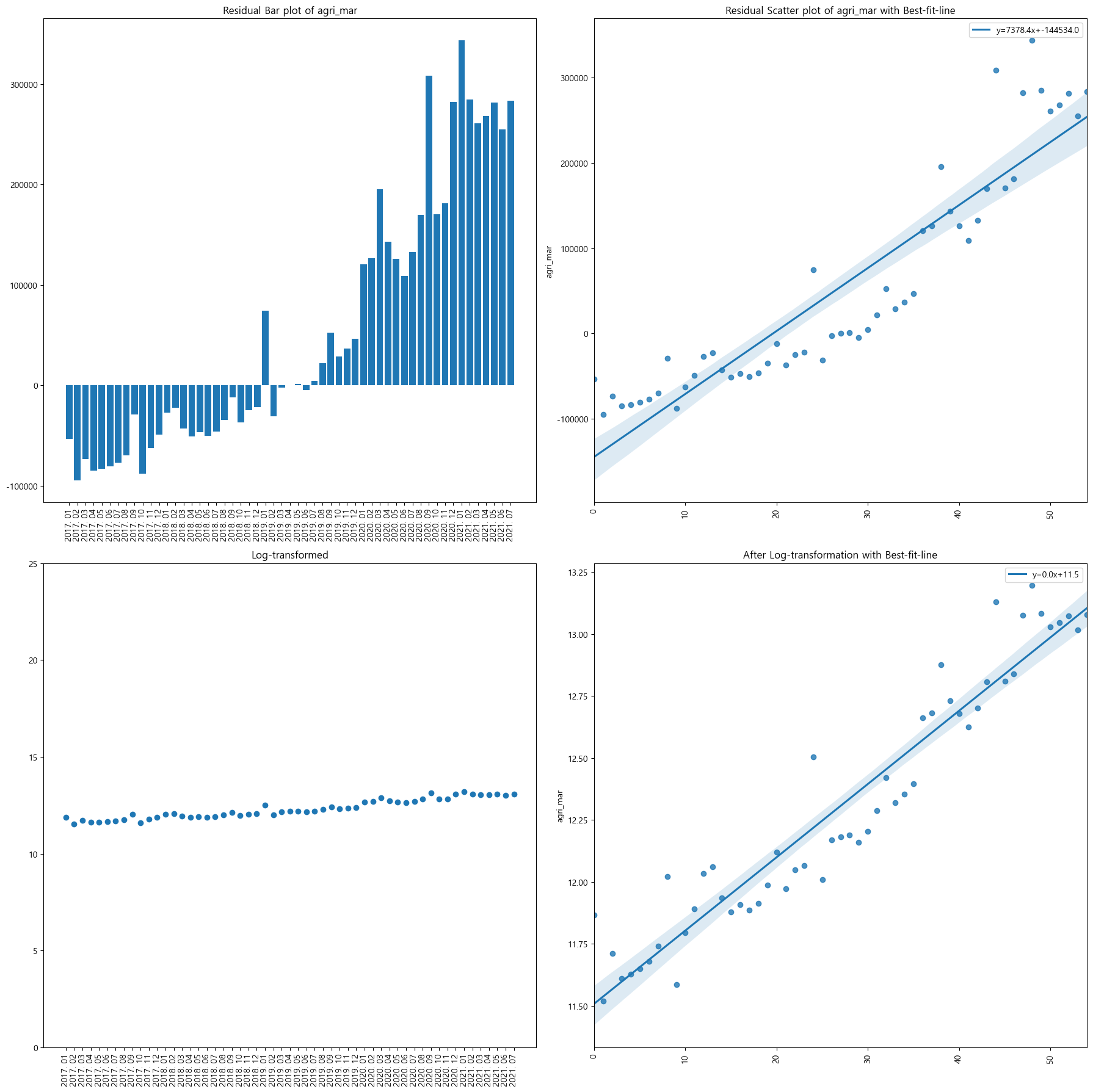
ax.legend()

plt.xticks(rotation=90)

plt.tight\_layout()

plt.grid

plt.show()

****

**## 확진자 수를 기준으로 모바일쇼핑 산업군별 Pearson's correlation 시각화 그래프 (2020.01 - 2021.07)**

# 참고: positive\_total = 국내 확진자 + 해외 유입 확진자

fig = plt.figure(figsize=(5,20))

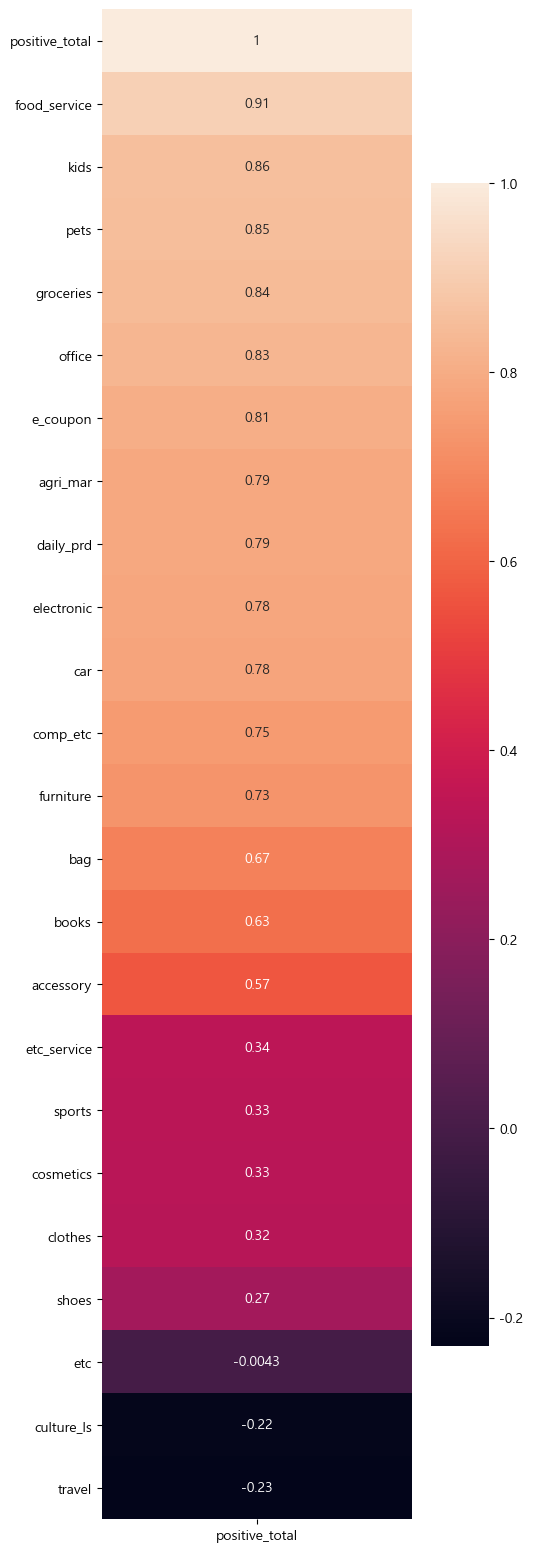
sns.heatmap(mobile\_corona.drop(['trans\_amnt\_total','kor', 'foreign', 'death'], axis=1).corr()[['positive\_total']].sort\_values(by='positive\_total',ascending=False), annot=True)

plt.show()

**#결과:**

#상위 5개: food\_service, electronic, pets, kids, groceries

#하위 5개: etc, travel, culture\_ls, shoes, clothes



**## 확진자 수를 기준으로 인터넷쇼핑 산업군별 Pearson's correlation 시각화 그래프 (2020.01 - 2021.07)**

# 참고: positive\_total = 국내 확진자 + 해외 유입 확진자

fig = plt.figure(figsize=(5,20))

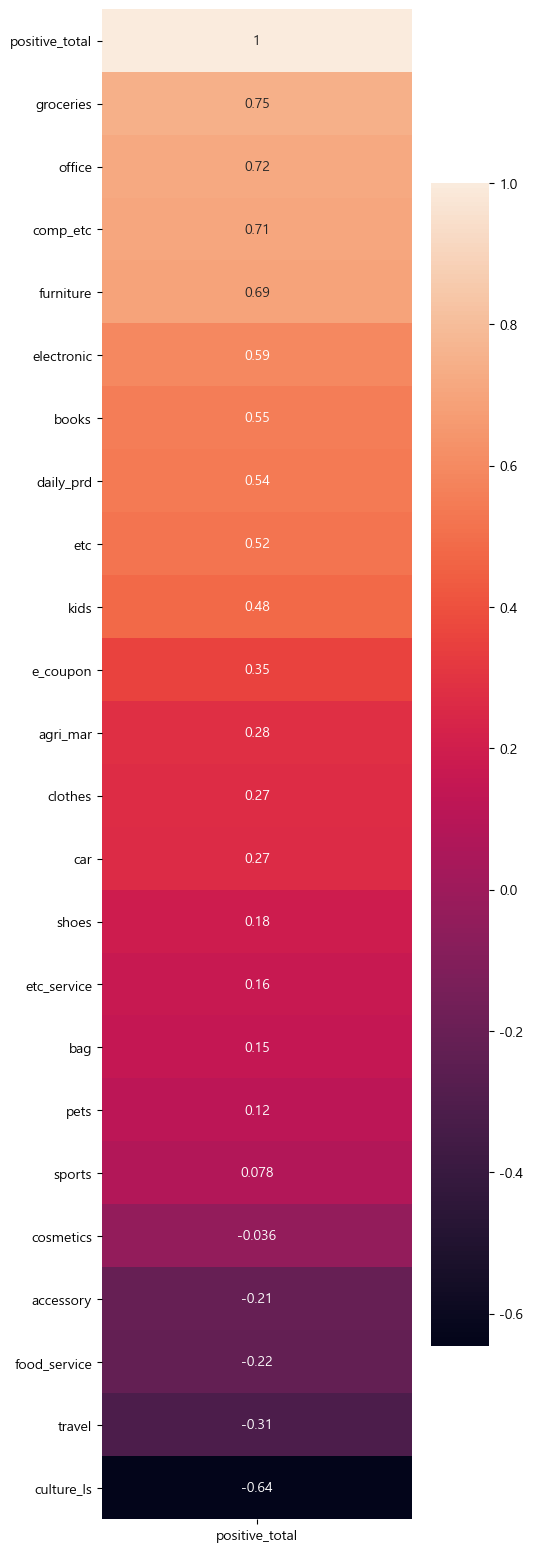
sns.heatmap(internet\_corona.drop(['trans\_amnt\_total','kor', 'foreign', 'death'], axis=1).corr()[['positive\_total']].sort\_values(by='positive\_total',ascending=False), annot=True)

plt.show()

**#결과:**

#상위 5개: groceries, office, comp\_etc, furniture, electronic

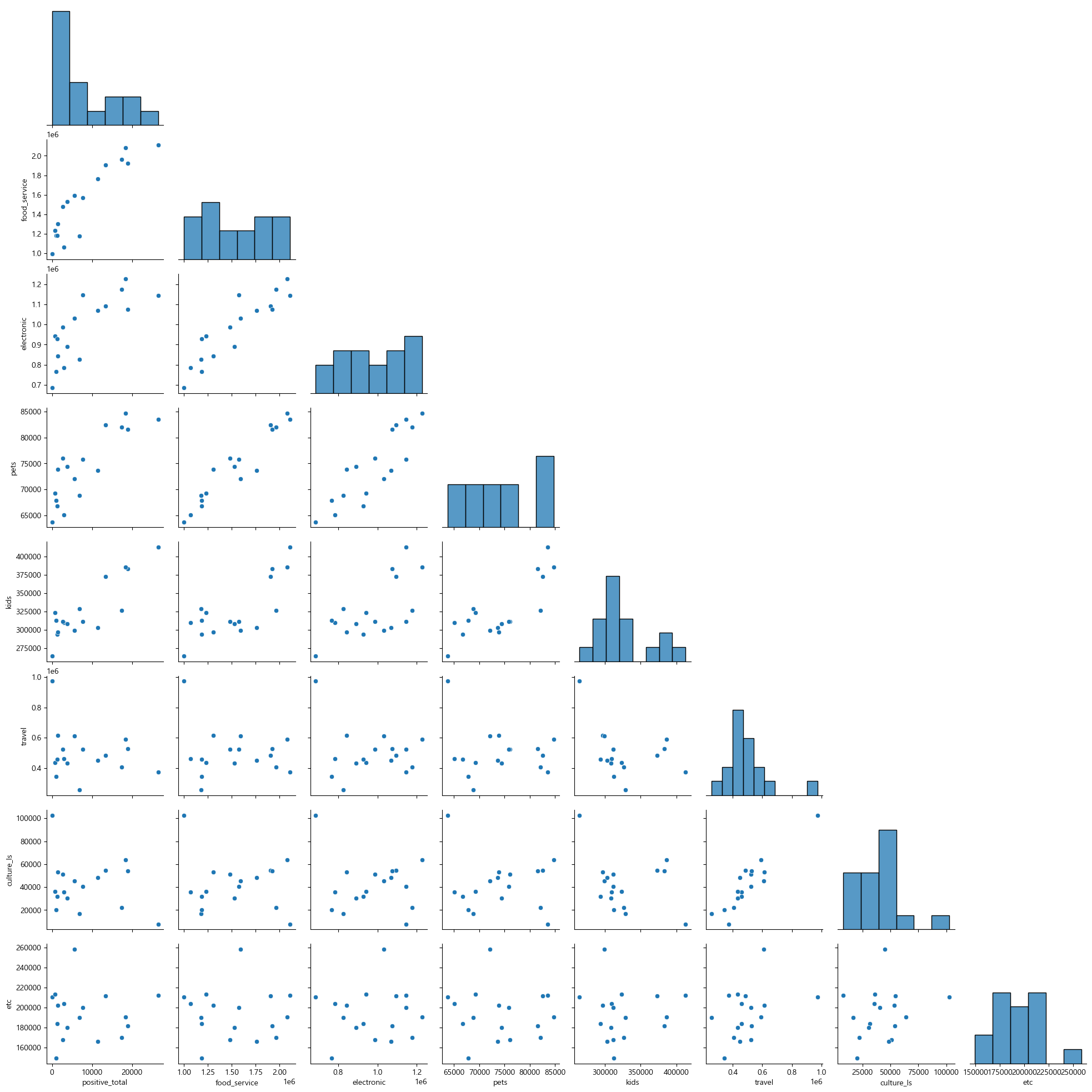
#하위 5개: culture\_ls, travel, food\_service, accessory, cosmetics



**## (모바일쇼핑)상위 3개(food\_service, electronic, pets)와 하위 3개(travel, culture\_ls, etc)와 확진자 수와의 pairplot**

#확진자 수와의 관계만을 보고싶을 때, 맨 왼쪽 column 라인을 보면 됨

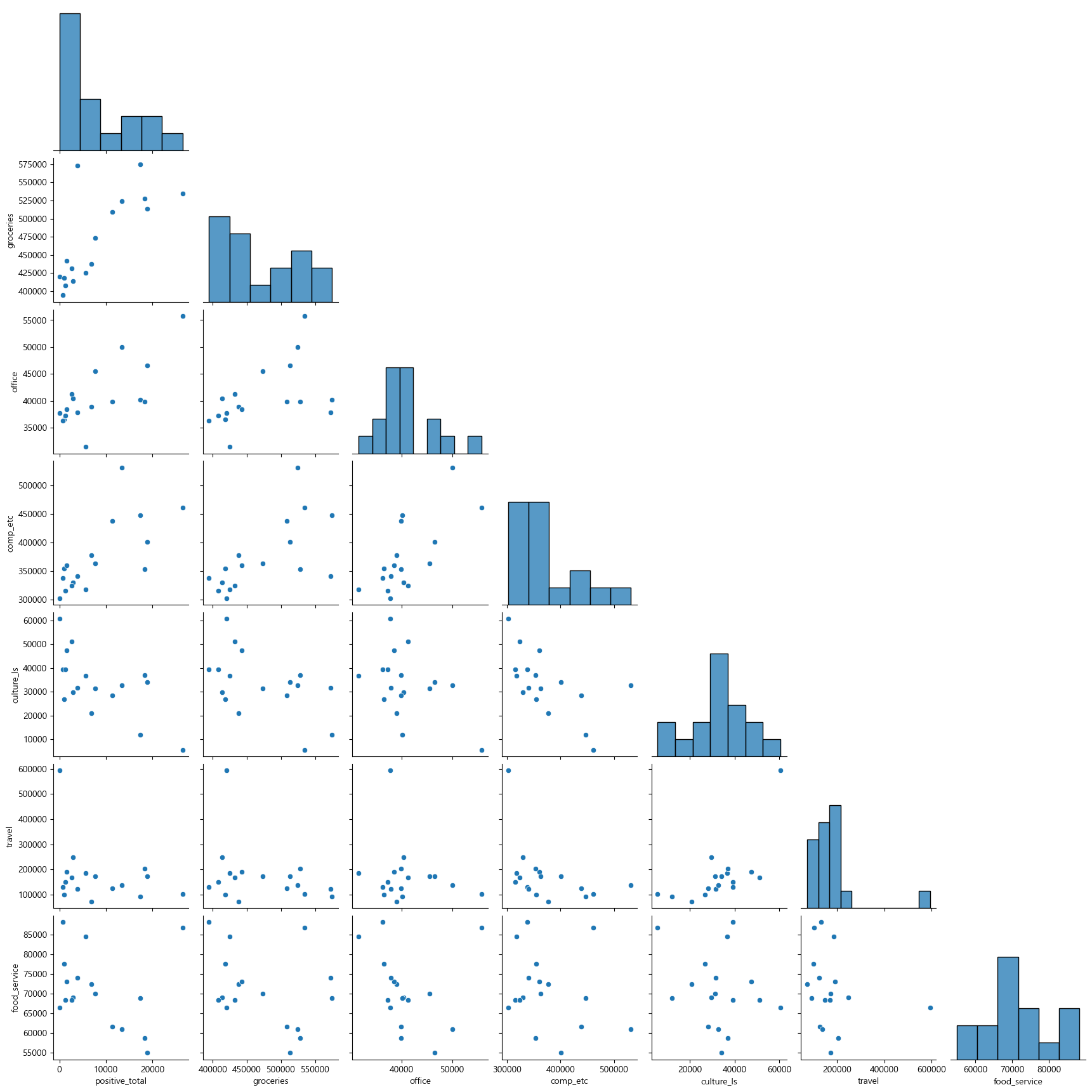
sns.pairplot(mobile\_corona, vars=['positive\_total','food\_service', 'electronic', 'pets', 'kids', 'travel','culture\_ls', 'etc'], corner=True)



**## (인터넷쇼핑)상위 3개(groceries, office, comp\_etc)와 하위 3개(culture\_ls, travel, food\_service)와 확진자 수와의 pairplot**

#확진자 수와의 관계만을 보고싶을 때, 맨 왼쪽 column 라인을 보면 됨

sns.pairplot(internet\_corona, vars=['positive\_total', 'groceries', 'office', 'comp\_etc','culture\_ls', 'travel', 'food\_service'], corner=True)



**## 확진자 수, 연령대별을 기준으로 모바일쇼핑 산업군별 Pearson's correlation 시각화 그래프**

# 기간: 2020.01 - 2021.07

# positive\_total = 국내 발생 + 해외 유입

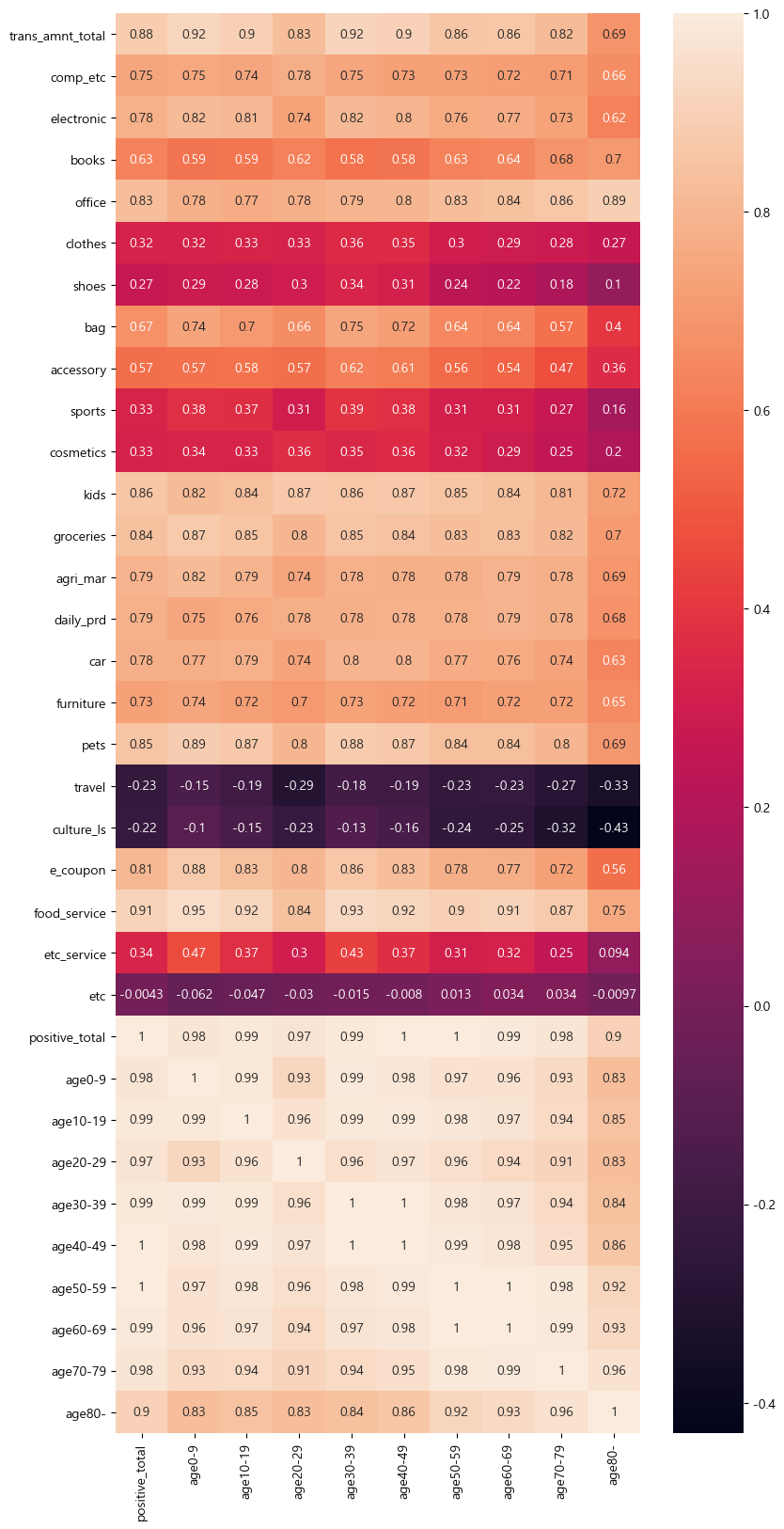
fig = plt.figure(figsize=(9,20))

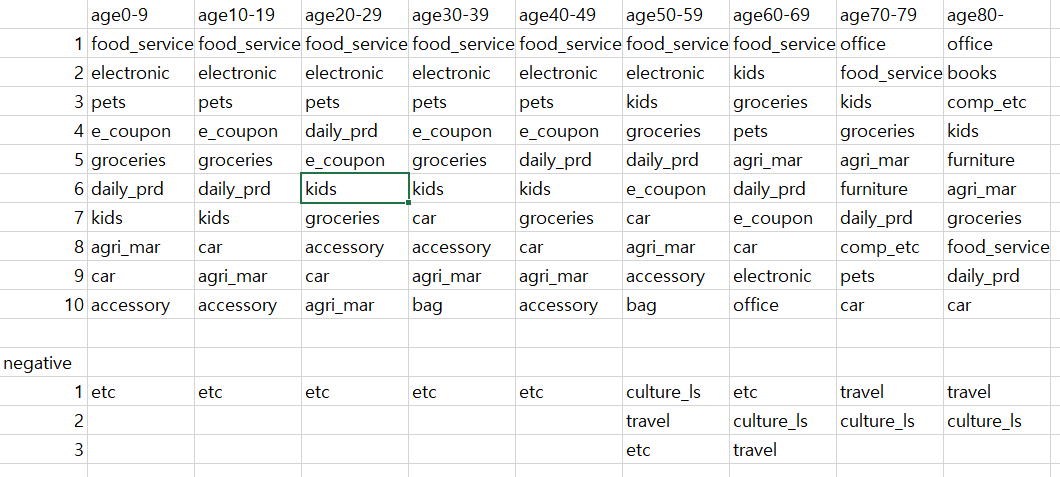
sns.heatmap(age\_mobile\_corona.drop(['kor','foreign','death','age\_total'], axis=1).corr()[['positive\_total', 'age0-9',

'age10-19', 'age20-29', 'age30-39', 'age40-49', 'age50-59', 'age60-69',

'age70-79', 'age80-']], annot=True)

plt.show()





**## 확진자 수, 연령대별을 기준으로 인터넷쇼핑 산업군별 Pearson's correlation 시각화 그래프**

# 기간: 2020.01 - 2021.07

# positive\_total = 국내 발생 + 해외 유입

fig = plt.figure(figsize=(9,20))

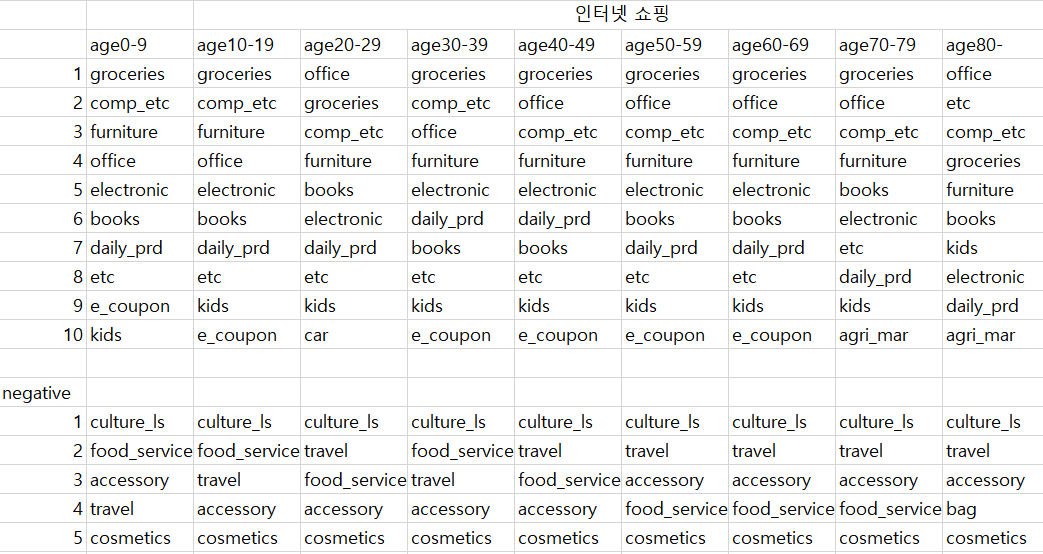
sns.heatmap(age\_internet\_corona.drop(['kor','foreign','death','age\_total'], axis=1).corr()[['positive\_total', 'age0-9',

'age10-19', 'age20-29', 'age30-39', 'age40-49', 'age50-59', 'age60-69',

'age70-79', 'age80-']], annot=True)

plt.show()





**## 확진자 수, 연령대별을 기준으로 모바일+인터넷쇼핑 산업군별 Pearson's correlation 시각화 그래프**

# 기간: 2020.01 - 2021.07

# positive\_total = 국내 발생 + 해외 유입

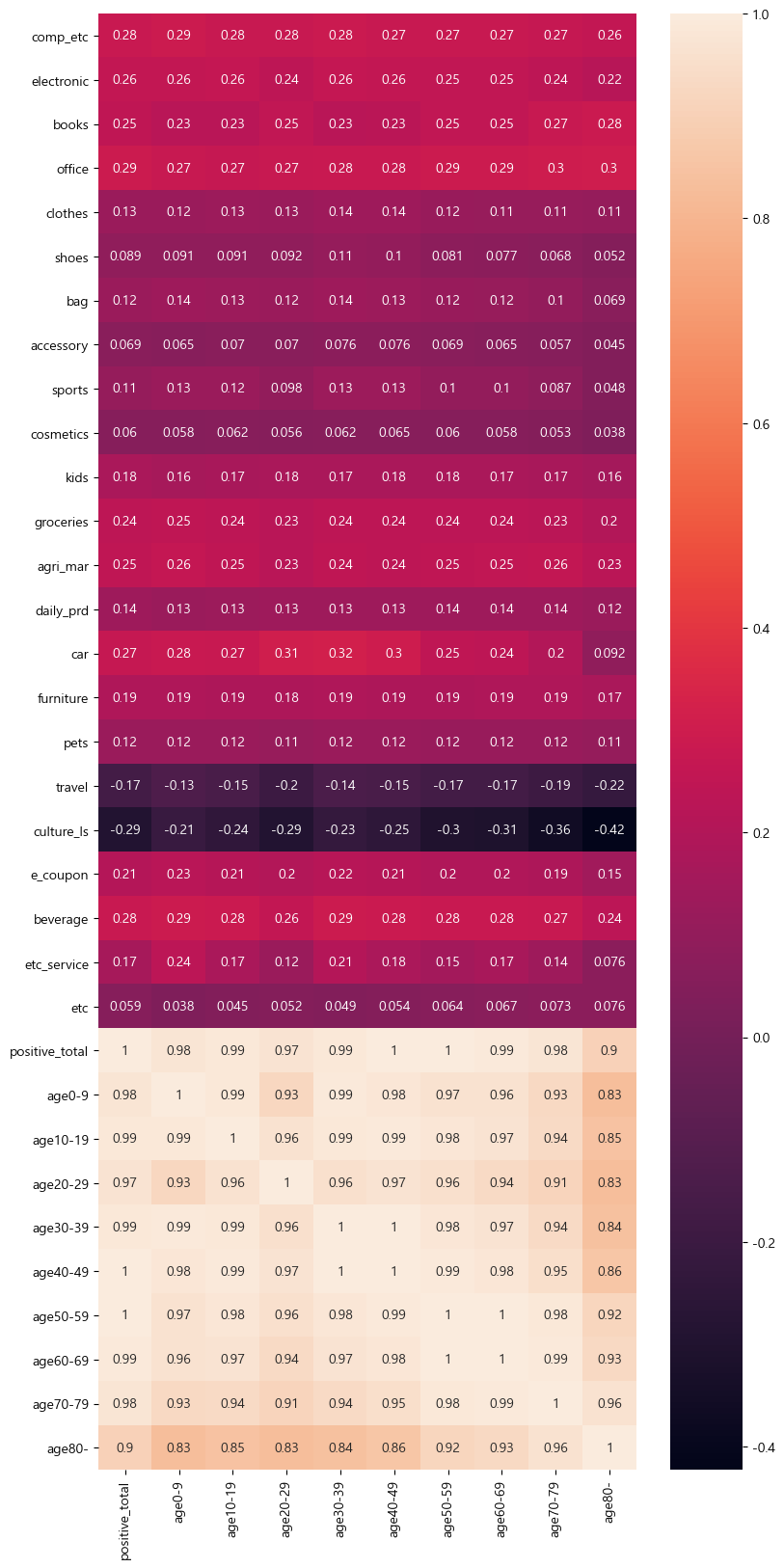
fig = plt.figure(figsize=(9,20))

sns.heatmap(total\_shopping\_corona.drop(['date', 'trans\_amnt\_total'], axis=1).corr()[['positive\_total', 'age0-9',

'age10-19', 'age20-29', 'age30-39', 'age40-49', 'age50-59', 'age60-69',

'age70-79', 'age80-']], annot=True)

plt.show()

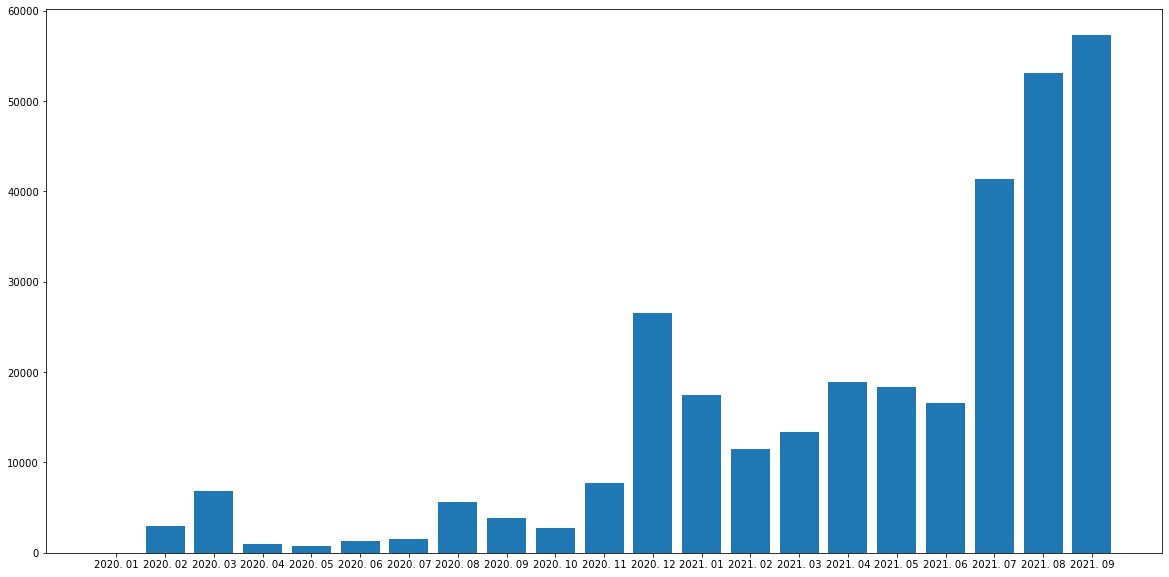


# 월별 코로나 확진자 수 그래프 (by 태양님)

plt.figure(figsize=(20, 10))

plt.bar(corona\_df['date'], corona\_df['total'])

plt.show()



**#(인터넷) 문화 카테고리에 대한 확진자 연령대별 Pearson's correlation 상관관계 heatmap**

fig = plt.figure(figsize=(9,20))

sns.heatmap(click\_data\_total\_age.drop(['date', 'total', 'elec\_age10-19', 'elec\_age20-29', 'elec\_age30-39', 'elec\_age40-49',

'elec\_age50-59', 'elec\_age60-', 'gro\_age10-19', 'gro\_age20-29',

'gro\_age30-39', 'gro\_age40-49', 'gro\_age50-59', 'gro\_age60-',

'kids\_age10-19', 'kids\_age20-29', 'kids\_age30-39', 'kids\_age40-49',

'kids\_age50-59', 'kids\_age60-', 'trav\_age10-19', 'trav\_age20-29',

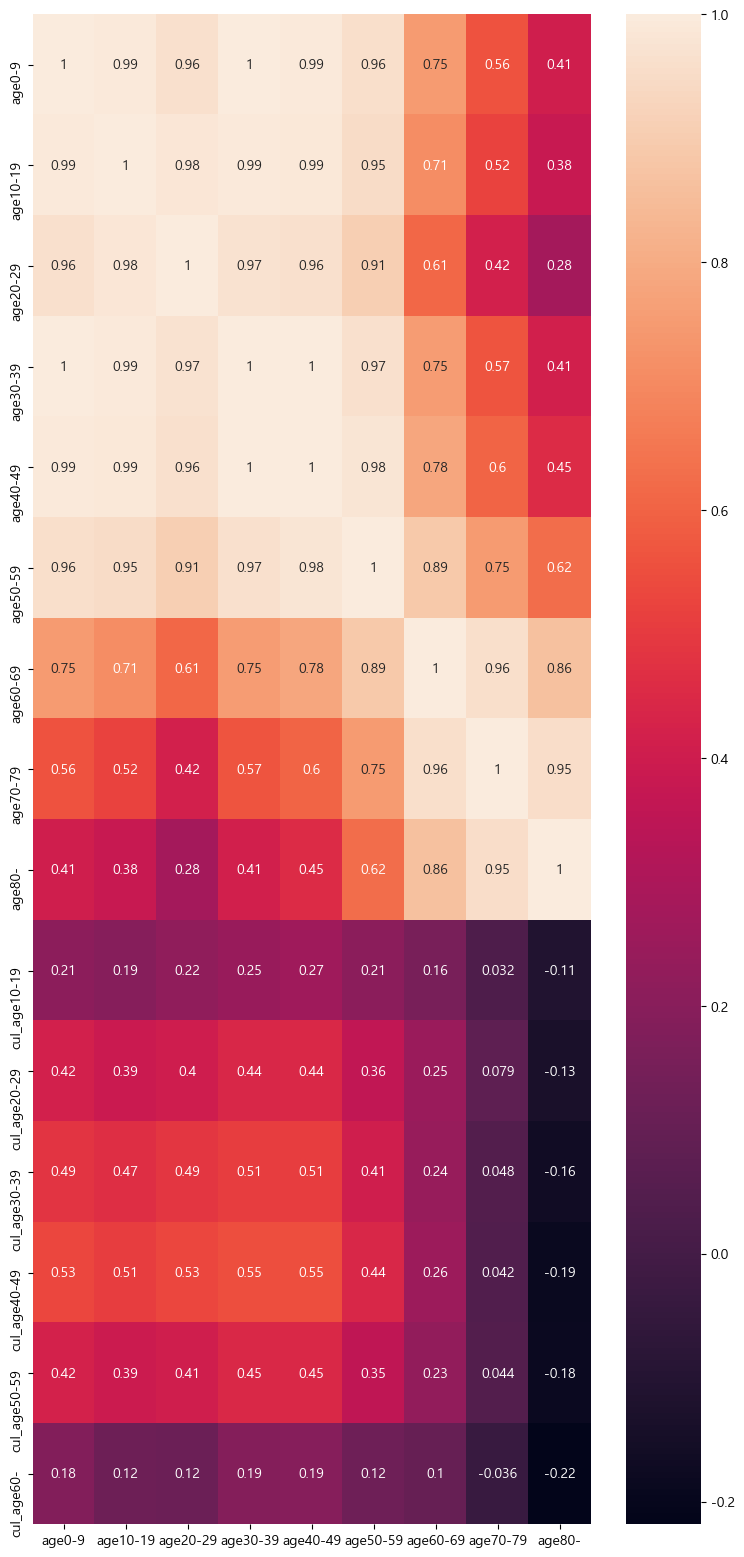
'trav\_age30-39', 'trav\_age40-49', 'trav\_age50-59', 'trav\_age60-',

'offi\_age10-19', 'offi\_age20-29',

'offi\_age30-39', 'offi\_age40-49', 'offi\_age50-59', 'offi\_age60-'], axis=1).corr()[['age0-9', 'age10-19', 'age20-29', 'age30-39',

'age40-49', 'age50-59', 'age60-69', 'age70-79', 'age80-']], annot=True)

plt.show()



**#(인터넷) 음식료품(grocery) 카테고리에 대한 확진자 연령대별 Pearson's correlation 상관관계 heatmap**

fig = plt.figure(figsize=(9,20))

sns.heatmap(click\_data\_total\_age.drop(['date', 'total',

'elec\_age10-19', 'elec\_age20-29', 'elec\_age30-39', 'elec\_age40-49',

'elec\_age50-59', 'elec\_age60-',

'kids\_age10-19', 'kids\_age20-29', 'kids\_age30-39', 'kids\_age40-49',

'kids\_age50-59', 'kids\_age60-', 'trav\_age10-19', 'trav\_age20-29',

'trav\_age30-39', 'trav\_age40-49', 'trav\_age50-59', 'trav\_age60-',

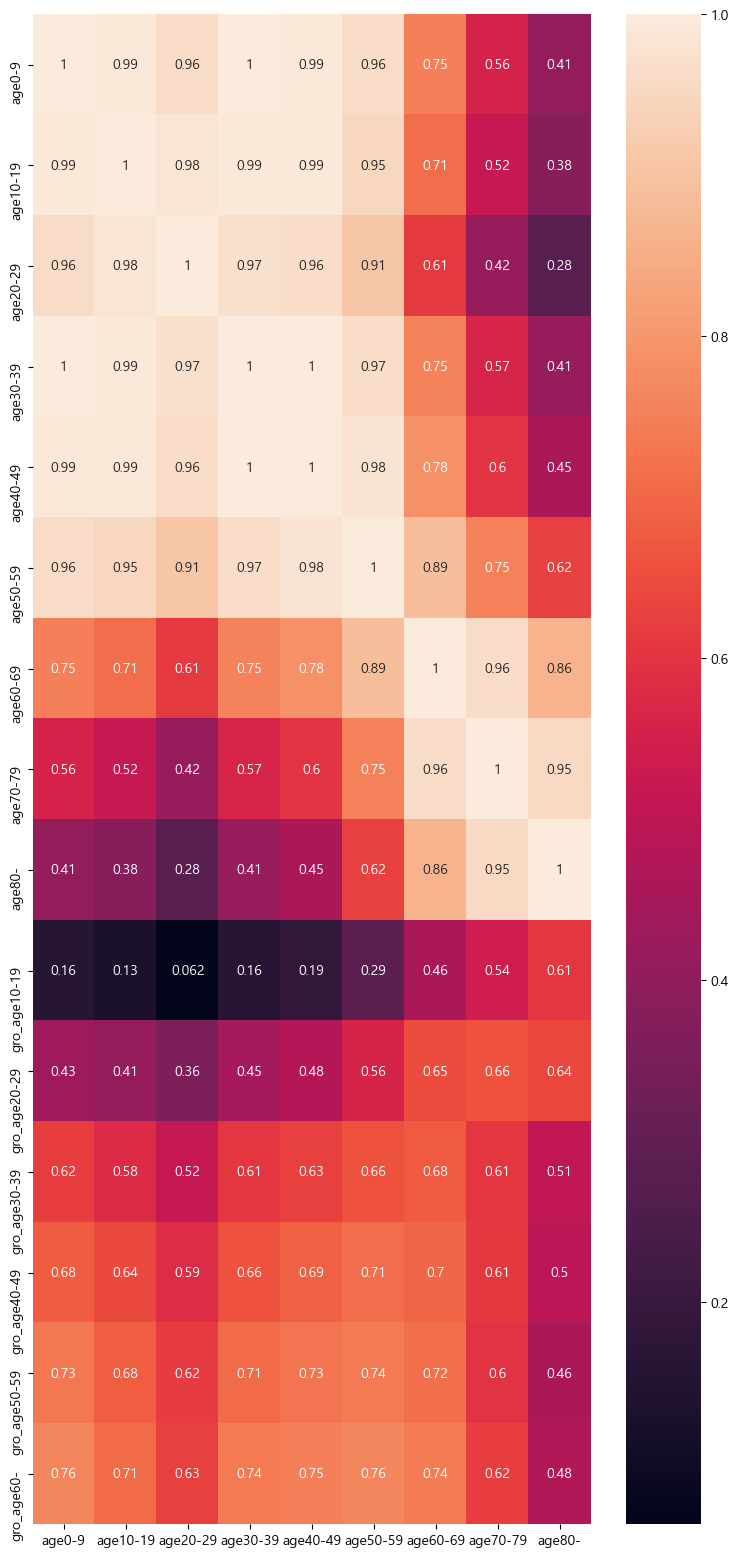
'cul\_age10-19', 'cul\_age20-29', 'cul\_age30-39', 'cul\_age40-49',

'cul\_age50-59', 'cul\_age60-', 'offi\_age10-19', 'offi\_age20-29',

'offi\_age30-39', 'offi\_age40-49', 'offi\_age50-59', 'offi\_age60-'], axis=1).corr()[['age0-9', 'age10-19', 'age20-29', 'age30-39',

'age40-49', 'age50-59', 'age60-69', 'age70-79', 'age80-']], annot=True)

plt.show()



**#(인터넷) 전자기기(electronics) 카테고리에 대한 확진자 연령대별 Pearson's correlation 상관관계 heatmap**

fig = plt.figure(figsize=(9,20))

sns.heatmap(click\_data\_total\_age.drop(['date', 'total',

'gro\_age10-19', 'gro\_age20-29',

'gro\_age30-39', 'gro\_age40-49', 'gro\_age50-59', 'gro\_age60-',

'kids\_age10-19', 'kids\_age20-29', 'kids\_age30-39', 'kids\_age40-49',

'kids\_age50-59', 'kids\_age60-', 'trav\_age10-19', 'trav\_age20-29',

'trav\_age30-39', 'trav\_age40-49', 'trav\_age50-59', 'trav\_age60-',

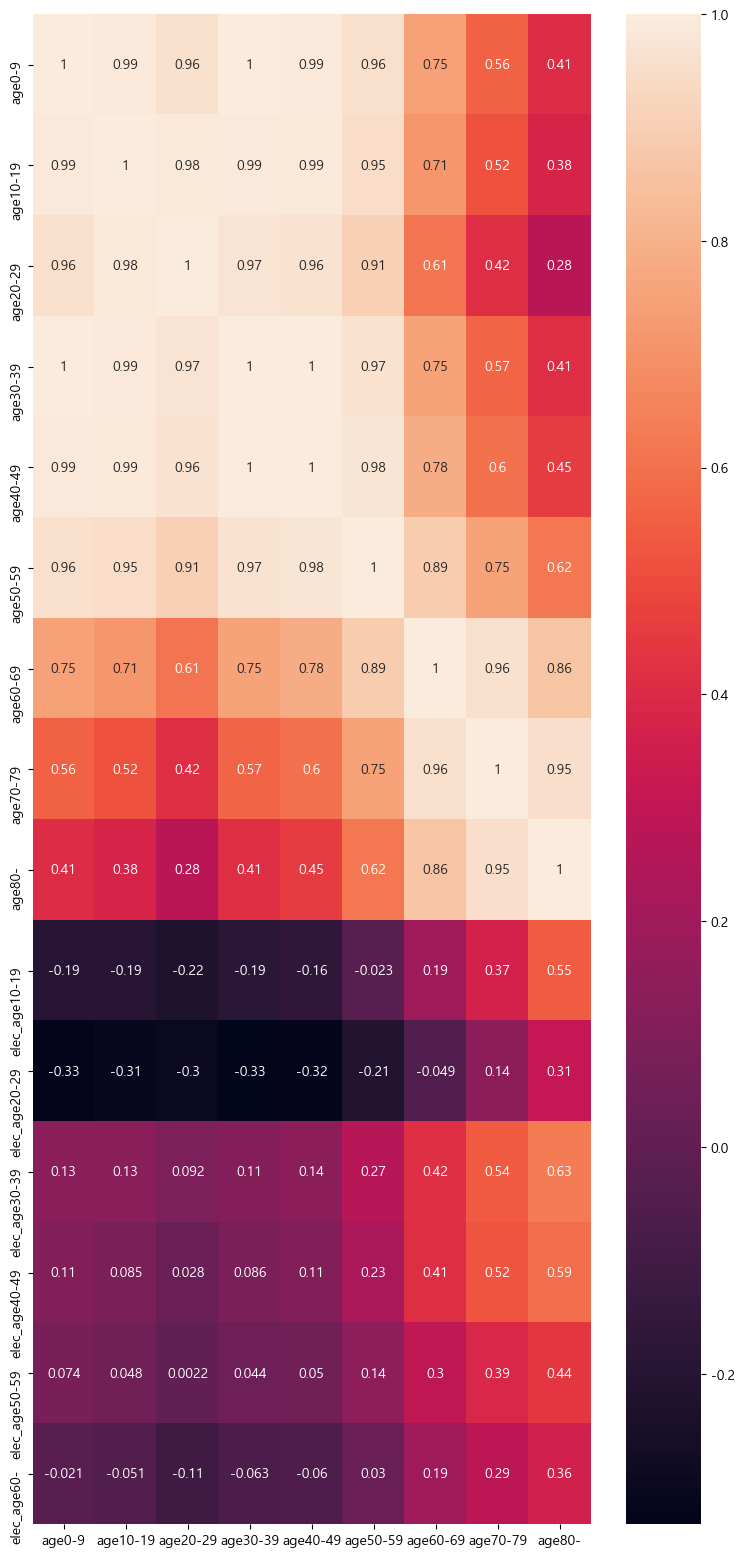
'cul\_age10-19', 'cul\_age20-29', 'cul\_age30-39', 'cul\_age40-49',

'cul\_age50-59', 'cul\_age60-', 'offi\_age10-19', 'offi\_age20-29',

'offi\_age30-39', 'offi\_age40-49', 'offi\_age50-59', 'offi\_age60-'], axis=1).corr()[['age0-9', 'age10-19', 'age20-29', 'age30-39',

'age40-49', 'age50-59', 'age60-69', 'age70-79', 'age80-']], annot=True)

plt.show()



**#(인터넷) 아동 유아용품(kids) 카테고리에 대한 확진자 연령대별 Pearson's correlation 상관관계 heatmap**

fig = plt.figure(figsize=(9,20))

sns.heatmap(click\_data\_total\_age.drop(['date', 'total',

'elec\_age10-19', 'elec\_age20-29', 'elec\_age30-39', 'elec\_age40-49',

'elec\_age50-59', 'elec\_age60-', 'gro\_age10-19', 'gro\_age20-29',

'gro\_age30-39', 'gro\_age40-49', 'gro\_age50-59', 'gro\_age60-',

'trav\_age10-19', 'trav\_age20-29',

'trav\_age30-39', 'trav\_age40-49', 'trav\_age50-59', 'trav\_age60-',

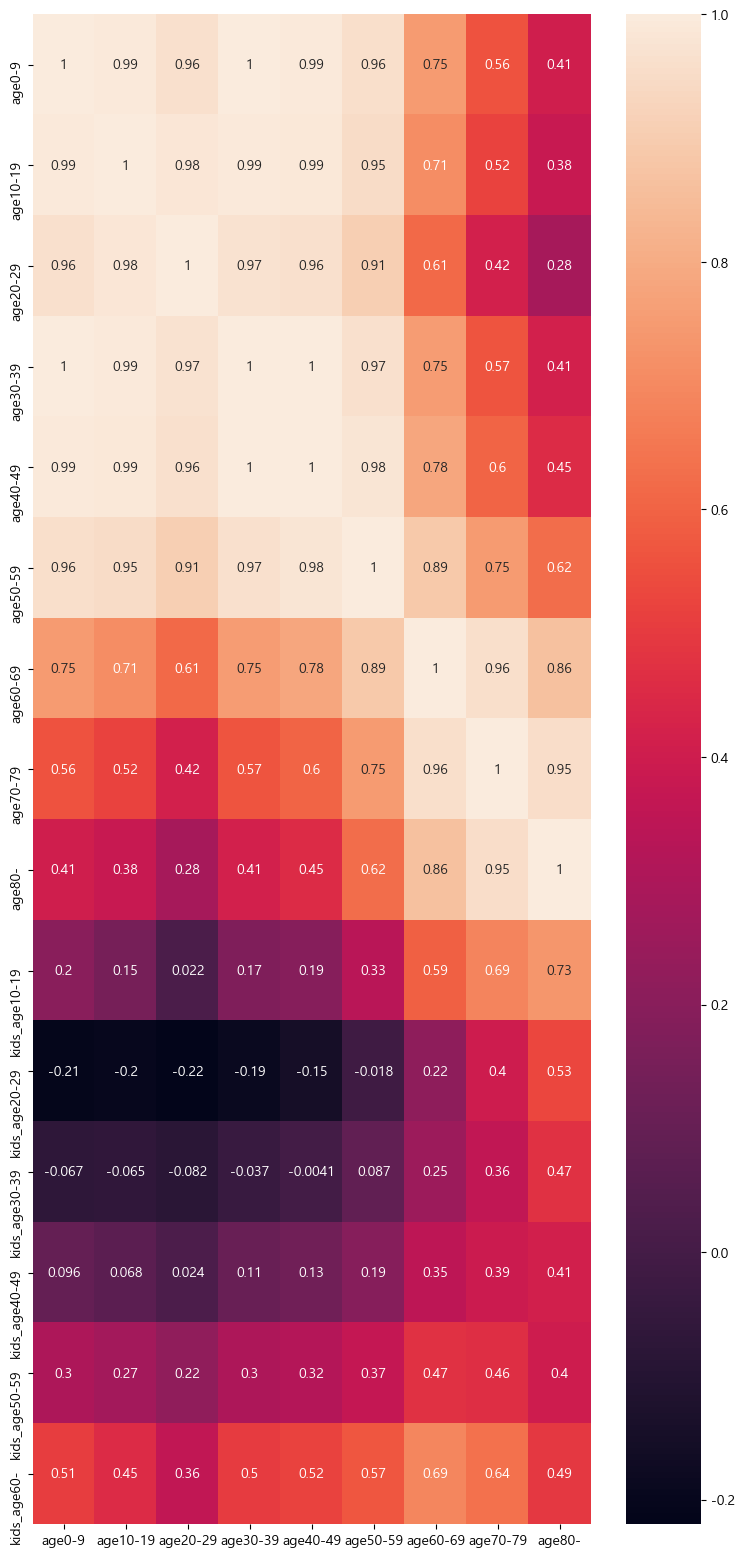
'cul\_age10-19', 'cul\_age20-29', 'cul\_age30-39', 'cul\_age40-49',

'cul\_age50-59', 'cul\_age60-', 'offi\_age10-19', 'offi\_age20-29',

'offi\_age30-39', 'offi\_age40-49', 'offi\_age50-59', 'offi\_age60-'], axis=1).corr()[['age0-9', 'age10-19', 'age20-29', 'age30-39',

'age40-49', 'age50-59', 'age60-69', 'age70-79', 'age80-']], annot=True)

plt.show()



**#(인터넷) 여행(travel) 카테고리에 대한 확진자 연령대별 Pearson's correlation 상관관계 heatmap**

fig = plt.figure(figsize=(9,20))

sns.heatmap(click\_data\_total\_age.drop(['date', 'total',

'elec\_age10-19', 'elec\_age20-29', 'elec\_age30-39', 'elec\_age40-49',

'elec\_age50-59', 'elec\_age60-', 'gro\_age10-19', 'gro\_age20-29',

'gro\_age30-39', 'gro\_age40-49', 'gro\_age50-59', 'gro\_age60-',

'kids\_age10-19', 'kids\_age20-29', 'kids\_age30-39', 'kids\_age40-49',

'kids\_age50-59', 'kids\_age60-',

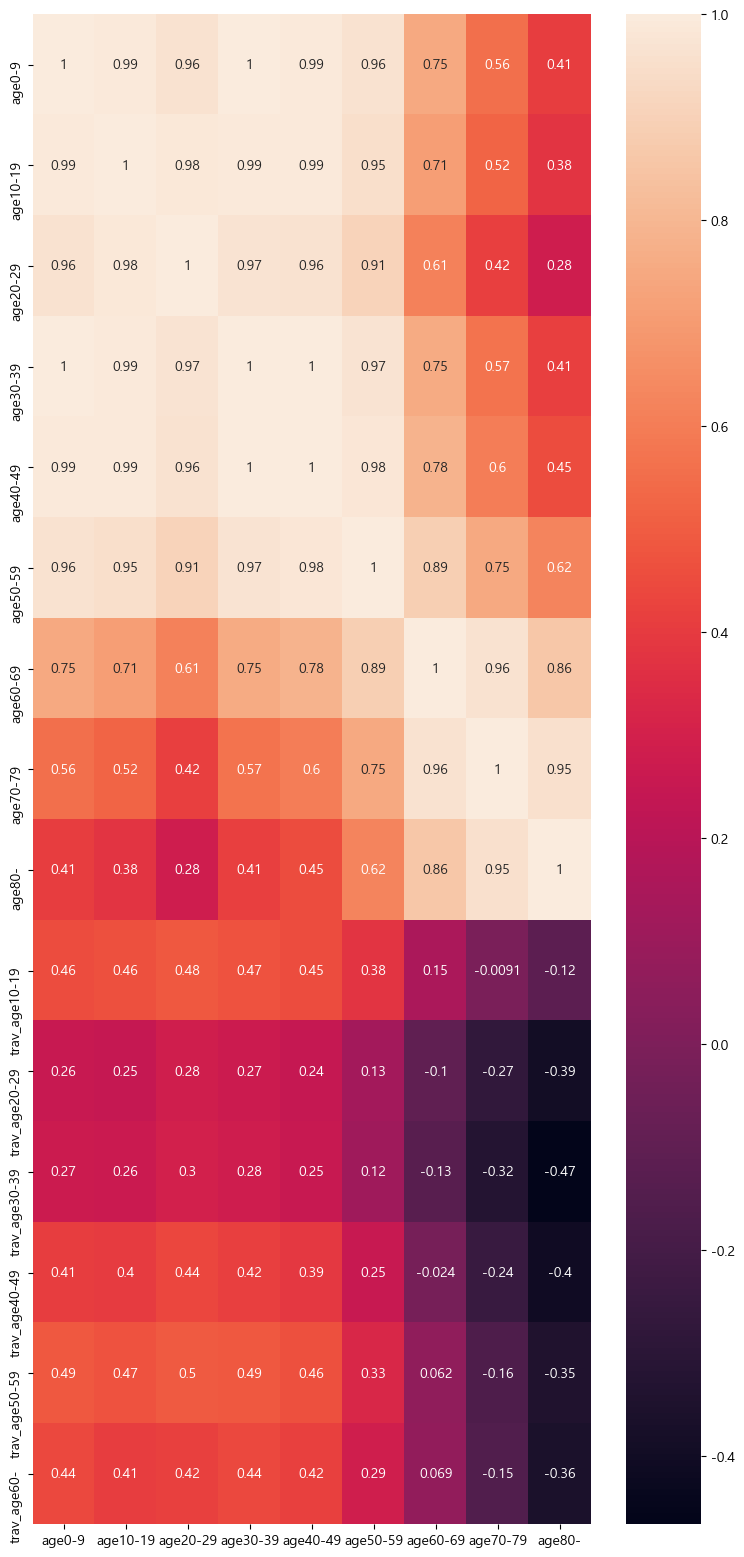
'cul\_age10-19', 'cul\_age20-29', 'cul\_age30-39', 'cul\_age40-49',

'cul\_age50-59', 'cul\_age60-', 'offi\_age10-19', 'offi\_age20-29',

'offi\_age30-39', 'offi\_age40-49', 'offi\_age50-59', 'offi\_age60-'], axis=1).corr()[['age0-9', 'age10-19', 'age20-29', 'age30-39',

'age40-49', 'age50-59', 'age60-69', 'age70-79', 'age80-']], annot=True)

plt.show()



**#(인터넷) 사무용품(office) 카테고리에 대한 확진자 연령대별 Pearson's correlation 상관관계 heatmap**

fig = plt.figure(figsize=(9,20))

sns.heatmap(click\_data\_total\_age.drop(['date', 'total',

'elec\_age10-19', 'elec\_age20-29', 'elec\_age30-39', 'elec\_age40-49',

'elec\_age50-59', 'elec\_age60-', 'gro\_age10-19', 'gro\_age20-29',

'gro\_age30-39', 'gro\_age40-49', 'gro\_age50-59', 'gro\_age60-',

'kids\_age10-19', 'kids\_age20-29', 'kids\_age30-39', 'kids\_age40-49',

'kids\_age50-59', 'kids\_age60-', 'trav\_age10-19', 'trav\_age20-29',

'trav\_age30-39', 'trav\_age40-49', 'trav\_age50-59', 'trav\_age60-',

'cul\_age10-19', 'cul\_age20-29', 'cul\_age30-39', 'cul\_age40-49',

'cul\_age50-59', 'cul\_age60-'], axis=1).corr()[['age0-9', 'age10-19', 'age20-29', 'age30-39',

'age40-49', 'age50-59', 'age60-69', 'age70-79', 'age80-']], annot=True)

plt.show()

