빅데이터분석 A조

김가연 김성현 남승현 이나영 최진오

목차

- 연구주제 및 가설
- 지역 및 업종 선택이유
- 기술통계량
- 소비자패턴분석
- 회귀분석

PART 1, 연구주제 및 가설

연구주제

COVID-19 상황 이후의 대구,경북 지역의 업종별 카드 사용 패턴 분석

가설

H0 : 코로나 상황 전, 후 5가지 서비스업의 매출 변화가 없다.
H1 : 코로나 상황 전, 후 5가지 서비스업의 매출 변화가 있다.

연구목표

데이터를 회귀 분석하여 예측모델을 세운다.

PART 2, 지역 및 업종 선택이유

대구 경북 지역을 선택한 이유

확진자 지역별 발생현황 (3월 3일 00시 기준, 4,812명)

지역	확진환자수	(%)	발생률*	지역	확진환자수	(%)	발생률*
서울	98	(2.0)	1.0	경기	94	(2.0)	0.7
부산	90	(1.9)	2.6	강원	20	(0.4)	1.3
대구	3,601	(74.8)	147.8	충북	11	(0.2)	0.7
인천	7	(0.1)	0.2	충남	81	(1.7)	3.8
광주	11	(0.2)	0.8	전북	7	(0.1)	0.4
대전	14	(0.3)	0.9	전남	5	(0.1)	0.3
울산	20	(0.4)	1.7	경북	685	(14.2)	25.7
세종	1	(0.0)	0.3	경남	64	(1.3)	1.9
				제주	3	(0.1)	0.4
				총합계	4,812	(100.0)	9.3

* 인구 10만 명당 (지역별 인구 출처 : 행정안전부, 주민등록인구현황 (2020년 1월 기준))

출처 : https://mdon.co.kr/news/article.html?no=25882
보건복지부

코로나19 신규 환자, 첫 슈퍼전파 사례 발생

국내 신종 코로나바이러스 감염증(코로나19) 환자 가운데 10명이 대구에 있는 한 신천지교회에 다니는 것으로 드러나면서 국내에서도 첫 슈퍼전파 사례가 나왔다. 19일 중앙방역대책본부(중대본)에 따르면 감염경로가 불분명한 31번 환자가 증상 발현 전후 4번 방문한 교회에서 집단으로 감염자가 나왔다. 국내서 10명 이상의 집단감염이 발생한 건 이번이 처음이다. 중대본은 한 장소에서 여러 명의 환자가 발생한 만큼 교회 감염자들을 슈퍼 전파 사례라고 인정했다. 다만 교회에서 발생한 확진자들의 공통 감염원이 31번 환자인지는 아직 확인되지 않았다고 전했다.

코로나19 확진자 하루만에 142명 폭증… 대구·경북 92% 차지(종합)

출처:moneys.mt.co.kr

신종 코로나바이러스 감염증(코로나19)이 장기화하면서 전 산업분야에 걸쳐 '고용 대란'이 현실화되고 있다. 1인 이상 30인 미만 소규모 사업체 종사자 증가율이 역대 가장 큰 폭으로 감소했고 코로나19 직격탄을 맞은 여행업, 관광숙박업, 관광운송업, 공연업 등에서 고용이 급감한 것으로 조사됐다. 또한 코로나19 피해가 가장 큰 대구·경북지역은 일자리가 줄어들고 있어 불안감이 증폭되고 있다.

출처 : '디지털타임스' 뉴스

대구 경북피해가 막대한 것을 알 수 있다.

5가지 업종을 선택한 이유

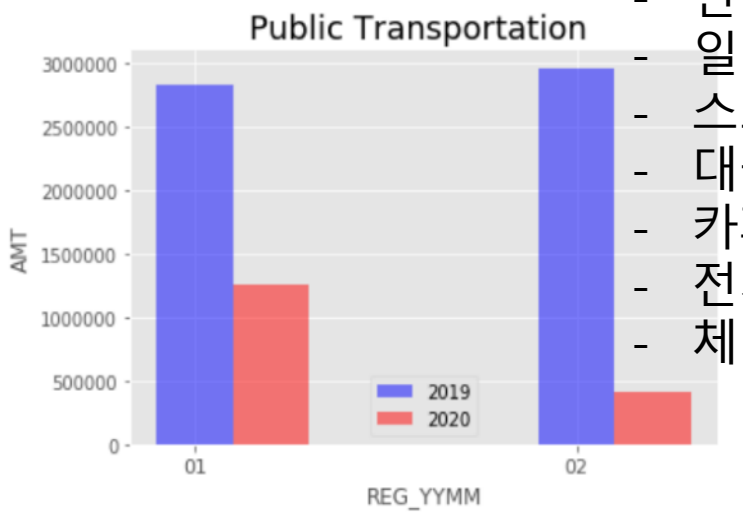
선택한 업종 :

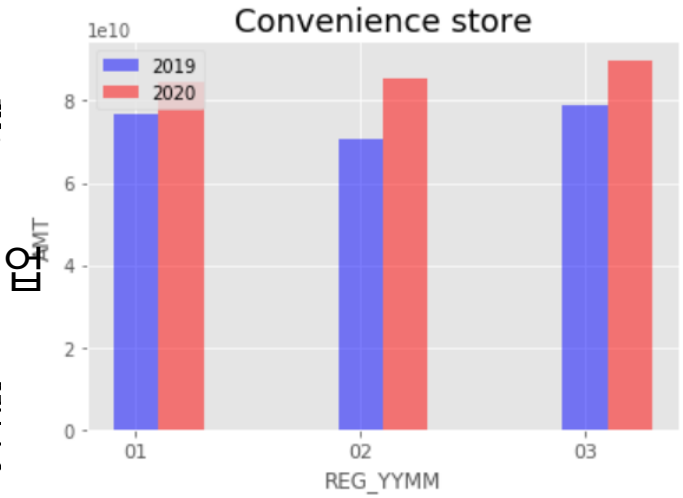
- **관광업**
- **일반 음식점업**
- 관광업
- 일반 음식점업
- 스포츠업
- 대중교통
- 카페
- 전시 및 행사 대행업
- 체인화 편의점

행사 대

가지 업종

위 5가지





5가지 업종을 선택한 이유

<데이터 전처리 과정>

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
data = pd.read_csv("201901-202003.csv")
```

```
data.columns = ['년월','사용지_시도','사용지_시군구','업종명','거주지_시도','거주지_시군구','연령대','성별','생애주기','고객수','금액','건수']
```

	년월	사용지_시도	사용지_시군구	업종명	거주지_시도	거주지_시군구	연령대	성별	생애주기	고객수	금액	건수
0	201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	20s	1	1	4	311200	4
1	201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	30s	1	2	7	1374500	8
2	201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	30s	2	2	6	818700	6
3	201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	40s	1	3	4	1717000	5
4	201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	40s	1	4	3	1047300	3

→ 변수들을 영어에서 한글로 바꾸었다.

5가지 업종을 선택한 이유

```
data_dg = data[(data['사용지_시도'] == "대구")]
data_gb = data[(data['사용지_시도'] == "경북")]
data_m = data_1[(data_1["년월"] == 201901) | (data_1["년월"] == 201902)|(data
01)

data_m
```

신종 코로나바이러스 국내 첫 확진자 발생

고신정 기자 ksj8855@doctorsnews.co.kr | 승인 2020.01.20 13:38 | 댓글 0

국내에서도 신종 코로나바이러스 감염증, 이른바 '우한 폐렴' 확진자가 나왔다.

질병관리본부는 1월 19일 중국 우한시에서 입국한 중국 국적의 35세 여성(중국 우한시 거주)에 대해 신종 코로나바이러스 감염증 검사를 시행한 결과, 20일 오전 확진자로 확정됐다고 밝혔다.

출처:doctorsnews.co.kr(의협신문)

20.01.19 국내 코로나 첫 확진자 발생

20년 1,2,3월
19년 1,2,3월
데이터 추출

							거주지_시군구	연령대	성별	생애주	고객수	금액	건수
597							시흥시	50s	1		3	633000	3
597							경산시	60s	2		3	445000	3
597							남구						3
597							남구						4
597							남구						5
...	...	...	...	...	...	...	...						...
23899480	202003	경북	포항시 북구	화장품 및 방향제 소매업	경북	포항시 북구							49
23899481	202003	경북	포항시 북구	화장품 및 방향제 소매업	경북	포항시 북구	70s	2	5		49	3425290	54
23899482	202003	경북	포항시 북구	화장품 및 방향제 소매업	대구	남구	20s	2	1		4	43800	5
23899483	202003	경북	포항시 북구	화장품 및 방향제 소매업	대구	달서구	20s	2	1		7	160280	15
23899484	202003	경북	포항시 북구	화장품 및 방향제 소매업	대구	북구	20s	2	1		3	15800	3

5가지 업종을 선택한 이유

```
data_s1 = data_m1.replace({"일식 음식점업":"일반 음식점업","한식 음식점업":"일반 음식점업","중식 음식점업":"일반 음식점업","일반유흥 주점업":"일반 음식점업",
                           "슈퍼마켓":"편의점","체인화 편의점":"편의점",
                           "호텔업":"관광업","여행사업":"관광업","정기 항공 운송업":"관광업",
                          "버스 운송업":"대중교통", "비알콜 음료점업":"카페"})
```

```
data_123 = data_s1[((data_s1['업종명'] == '일반 음식점업') | (data_s1['업종명'] == "전시 및 행사 대행업")
                    |(data_s1["업종명"] == '관광업') | (data_s1["업종명"] == "카페")| (data_s1["업종명"] == "스포츠 및 레크레이션 용품 임대업"))]
data_123.head()
```

	Unnamed: 0	년월	업종명	연령대	성별	생애주기	고객수	금액	건수
0	598167	201901	카페	20s	2	1	3	24890	3
1	598168	201901	카페	20s	2	1	3	17800	3
2	598169	201901	카페	30s	2	2	3	28500	4
3	598170	201901	카페	20s	1	1	3	27900	4
4	598171	201901	카페	40s	1	3	3	20600	3
...	...	...	...	...	...	...	...	...	...
363235	23899428	202003	일반 음식점업	20s	1	1	8	904800	16
363236	23899429	202003	일반 음식점업	30s	1	1	3	331800	4
363237	23899430	202003	일반 음식점업	30s	1	2	7	1153000	28
363238	23899431	202003	일반 음식점업	40s	1	3	3	73000	3
363239	23899432	202003	일반 음식점업	30s	1	2	4	321000	5

일반음식점업
한식, 중식, 일식 음식점업

관광업
호텔업, 여행사업, 정기항공운송업

카페
비알콜음료점업

5가지 업종을 선택한 이유

감소율

: 100-(비교대상)/(기준)*100)

1월 대비 2월 매출 감소율
: 100-(2월 총 매출합계(비교대상)/1월 총 매출합계(기준)*100)

2월 대비 3월 매출 감소율
: 100-(3월 총 매출합계(비교대상)/2월 총 매출합계(기준)*100)

매출 감소 : 양수 값(+)
매출 증가 : 음수 값(-)

5가지 업종을 선택한 이유

관광업의 매출 감소율

```
df_tour_1901=df_tour[(df_tour['년월']==201901)]
df_tour_1901
df_tour_1901_금액=df_tour_1901['금액']
df_tour_sum_1901=sum(df_tour_1901_금액)
df_tour_1902=df_tour[(df_tour['년월']==201902)]
df_tour_1902
df_tour_1902_금액=df_tour_1902['금액']
df_tour_sum_1902=sum(df_tour_1902_금액)
df_tour_1903=df_tour[(df_tour['년월']==201903)]
df_tour_1903
df_tour_1903_금액=df_tour_1903['금액']
df_tour_sum_1903=sum(df_tour_1903_금액)
df_tour_2001=df_tour[(df_tour['년월']==202001)]
df_tour_2001
df_tour_2001_금액=df_tour_2001['금액']
df_tour_sum_2001=sum(df_tour_2001_금액)
df_tour_2002=df_tour[(df_tour['년월']==202002)]
df_tour_2002
df_tour_2002_금액=df_tour_2002['금액']
df_tour_sum_2002=sum(df_tour_2002_금액)
df_tour_2003=df_tour[(df_tour['년월']==202003)]
df_tour_2003
df_tour_2003_금액=df_tour_2003['금액']
df_tour_sum_2003=sum(df_tour_2003_금액) #대구/경
```

```
print('20년 대구/경북 관광업 1,2월 매출감소율 :',100-((df_tour_sum_2002/df_tour_sum_2001)*100),'%')
```

20년 대구/경북 관광업 1,2월 매출감소율 : 49.839031241494204 %

```
print('20년 대구/경북 관광업 2,3월 매출감소율 :',100-((df_tour_sum_2003/df_tour_sum_2002)*100),'%')
```

20년 대구/경북 관광업 2,3월 매출감소율 : 90.39982870572885 %

5가지 업종을 선택한 이유

```
print('20년 대구/경북 카페 1,2월 매출감소율 :',100-((df_cafe_sum_2002/df_cafe_sum_2001)*100),'%')
```

20년 대구/경북 카페 1,2월 매출감소율 : 26.10094661723967 %

```
print('20년 대구/경북 일반음식점업 1,2월 매출감소율 :',100-((df_food_sum_2002/df_food_sum_2001)*100),'%')
```

20년 대구/경북 일반음식점업 1,2월 매출감소율 : 28.86065578454165 %

```
print('20년 대구/경북 카페 2,3월 매출감소율 :',100-((df_cafe_sum_2003/df_cafe_sum_2002)*100),'%')
```

20년 대구/경북 카페 2,3월 매출감소율 : 20.08255859576404 %

```
print('20년 대구/경북 일반음식점업 2,3월 매출감소율 :',100-((df_food_sum_2003/df_food_sum_2002)*100),'%')
```

20년 대구/경북 일반음식점업 2,3월 매출감소율 : 31.562636891469694 %

```
print('20년 대구/경북 스포츠업 1,2월 매출감소율 :',100-((df_sport_sum_2002/df_sport_sum_2001)*100),'%')
```

20년 대구/경북 스포츠업 1,2월 매출감소율 : 33.9707965910755 %

```
print('20년 대구/경북 전시 및 행사업 1,2월 매출감소율 :',100-((df_dis_sum_2002/df_dis_sum_2001)*100),'%')
```

20년 대구/경북 전시 및 행사업 1,2월 매출감소율 : 20.405423921057462 %

```
print('20년 대구/경북 스포츠업 2,3월 매출감소율 :',100-((df_sport_sum_2003/df_sport_sum_2002)*100),'%')
```

20년 대구/경북 스포츠업 2,3월 매출감소율 : 1.8687284575058243 %

```
print('20년 대구/경북 전시 및 행사업 2,3월 매출감소율 :',100-((df_dis_sum_2003/df_dis_sum_2002)*100),'%')
```

20년 대구/경북 전시 및 행사업 2,3월 매출감소율 : 79.20315402433431 %

5가지 업종을 선택한 이유

5가지 업종 전체의 매출 감소율

```
print('20년 대구/경북 1,2월 매출감소율 :',100-((df_sum_2002/df_sum_2001)*100),'%')
```

```
20년 대구/경북 1,2월 매출감소율 : 29.204145838426655 %
```

```
print('20년 대구/경북 2,3월 매출감소율 :',100-((df_sum_2003/df_sum_2002)*100),'%')
```

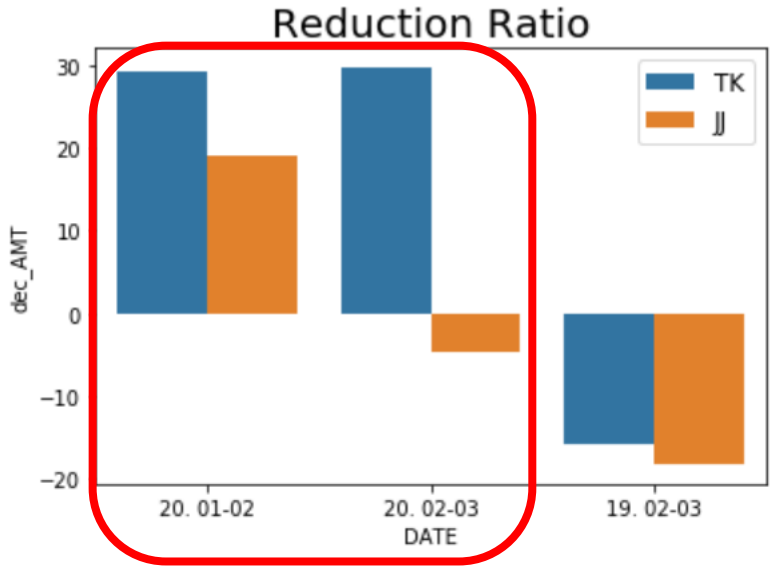
```
20년 대구/경북 2,3월 매출감소율 : 29.686100386679698 %
```

5개 업종은 평균적으로 '30%' 매출이 감소하였음을 알 수 있다.

매출 감소율 그래프는 뒷부분에

지역 및 업종을 선택한 이유

대구경북, 전북전남 5가지 업종의 매출 감소율 비교



TK : 대구경북
JJ : 전북전남

매출감소율이 클수록 막대가 위로 높게 올라간다.

두 지역을 비교해본 결과, **대구,경북지역의 매출감소율이 큰 변화가 있는 것**을 알 수 있다.
그리고, 발생하지 않은 19년 2-3월에는 매출이 증가했던 것을 확인할 수 있다.

PART 3, 기술통계량

기술통계량(bar plot)

매출 감소량

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
from scipy import stats
```

```
from matplotlib import font_manager, rc
fn_name = font_manager.FontProperties(fname='c:/Windows/Fonts/malgun.ttf').get_name()
rc('font',family=fn_name)
```

```
df = pd.read_csv("1,2,3월_2.csv")
df['사용지_시도']='대구경북'
```

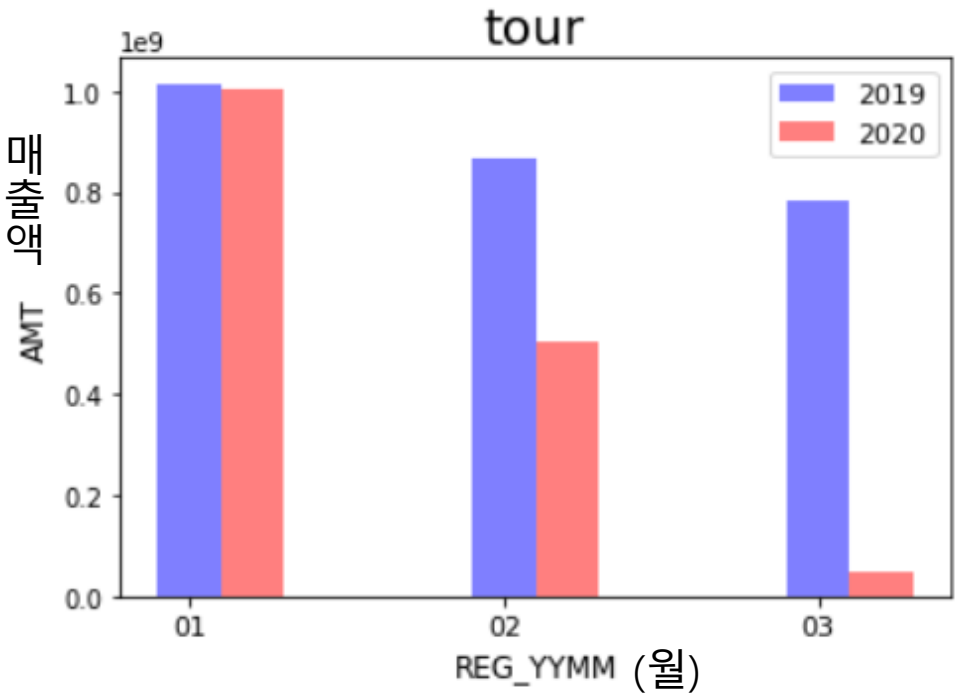
```
df_cafe=df[df['업종명']=='카페']
df_food=df[df['업종명']=='일반 음식점업']
df_tour=df[df['업종명']=='관광업']
df_sport=df[df['업종명']=='스포츠 및 레크레이션 용품 임대업']
df_dis=df[df['업종명']=='전시 및 행사 대행업']
```

연도, 월별매출액 저장

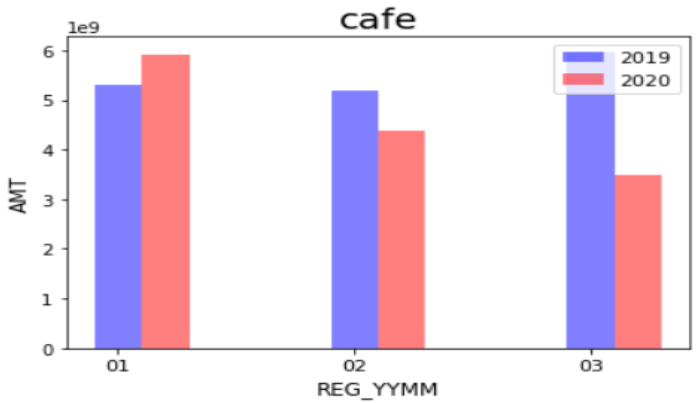
```
df_tour_1901=df_tour[(df_tour['년월']==201901)]
df_tour_1901
df_tour_1901_금액=df_tour_1901['금액']
df_tour_sum_1901=sum(df_tour_1901_금액)
df_tour_1902=df_tour[(df_tour['년월']==201902)]
df_tour_1902
df_tour_1902_금액=df_tour_1902['금액']
df_tour_sum_1902=sum(df_tour_1902_금액)
df_tour_1903=df_tour[(df_tour['년월']==201903)]
df_tour_1903
df_tour_1903_금액=df_tour_1903['금액']
df_tour_sum_1903=sum(df_tour_1903_금액)
df_tour_2001=df_tour[(df_tour['년월']==202001)]
df_tour_2001
df_tour_2001_금액=df_tour_2001['금액']
df_tour_sum_2001=sum(df_tour_2001_금액)
df_tour_2002=df_tour[(df_tour['년월']==202002)]
df_tour_2002
df_tour_2002_금액=df_tour_2002['금액']
df_tour_sum_2002=sum(df_tour_2002_금액)
df_tour_2003=df_tour[(df_tour['년월']==202003)]
df_tour_2003
df_tour_2003_금액=df_tour_2003['금액']
df_tour_sum_2003=sum(df_tour_2003_금액) #대구/경
```

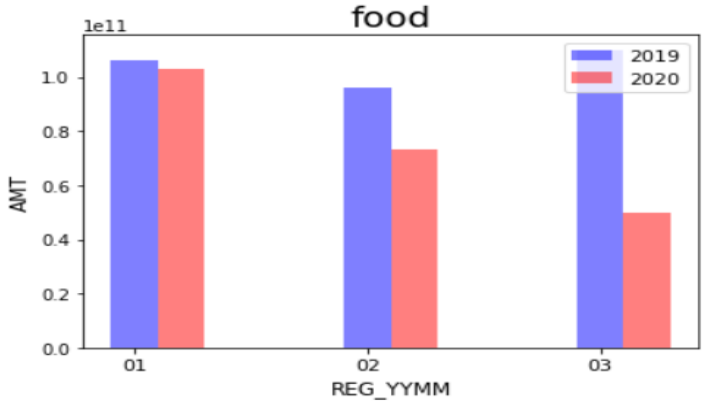
기술통계량(bar plot)

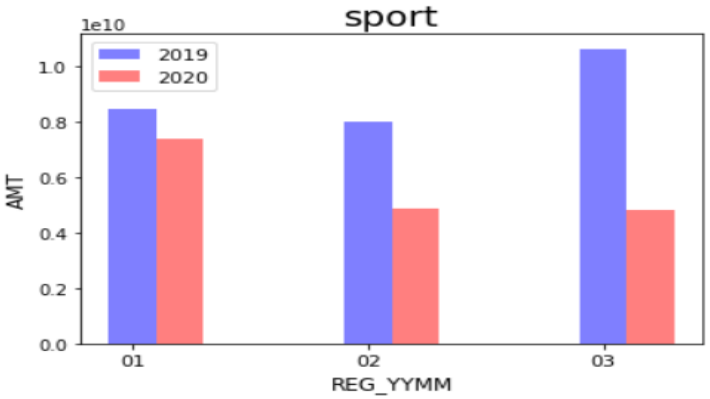
```
df_tour_2019 = pd.DataFrame({'년월' : [201901,201902,201903],
                             '금액' : [df_tour_sum_1901,df_tour_sum_1902,df_tour_sum_1903]})
df_tour_2020 = pd.DataFrame({'년월' : [202001,202002,202003],
                             '금액' : [df_tour_sum_2001,df_tour_sum_2002,df_tour_sum_2003]})

bar_width = 0.2
alpha = 0.5
p1 = plt.bar(index, df_tour_2019['금액'],

             bar_width,

             color='b',

             alpha=alpha,

             label='2019')

p2 = plt.bar(index + bar_width, df_tour_2020['금액'],

             bar_width,

             color='r',

             alpha=alpha,

             label='2020')


plt.title('tour', fontsize=20)
plt.ylabel('AMT', fontsize=12)
plt.xlabel('REG_YYMM', fontsize=12)
plt.xticks(index, label, fontsize=11)
plt.legend((p1[0], p2[0]), ('2019', '2020'), fontsize=11)
plt.show()
```

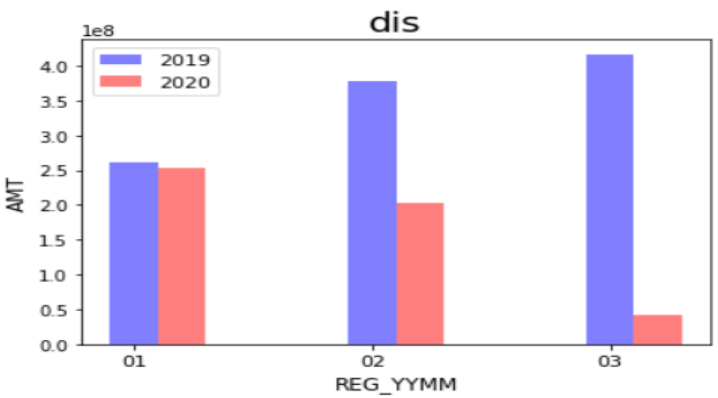


기술통계량(bar plot)









기술통계량(bar plot)

매출 감소율

감소율 : 100-(비교대상)/(기준)*100)

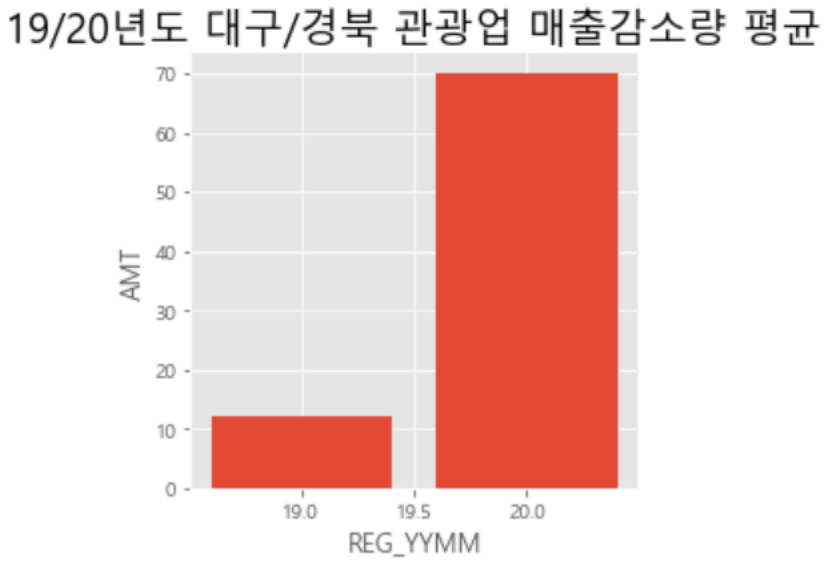
```
print('20년 대구/경북 관광업 1,2월 매출감소율 :',100-((df_tour_sum_2002/df_tour_sum_2001)*100),'%')
```

20년 대구/경북 관광업 1,2월 매출감소율 : 49.839031241494204 % -> 2월 매출 감소율

```
print('20년 대구/경북 관광업 2,3월 매출감소율 :',100-((df_tour_sum_2003/df_tour_sum_2002)*100),'%')
```

20년 대구/경북 관광업 2,3월 매출감소율 : 90.39982870572885 % -> 3월 매출 감소율

2월과 3월 매출감소율의 **평균**을 기준!



2019년 대구/경북 관광업 매출 감소량 평균

```
tour19=(100-((df_tour_sum_1902/df_tour_sum_1901)*100))+(100-((df_tour_sum_1903/df_tour_sum_1902)*100))
```

2020년 대구/경북 관광업 매출 감소량 평균

```
tour20=(100-((df_tour_sum_2002/df_tour_sum_2001)*100))+(100-((df_tour_sum_2003/df_tour_sum_2002)*100))
```

기술통계량(bar plot)

```
print('20년 대구/경북 카페 1,2월 매출감소율 :',100-((df_cafe_sum_2002/df_cafe_sum_2001)*100),'%')
```

20년 대구/경북 카페 1,2월 매출감소율 : 26.10094661723967 %

```
print('20년 대구/경북 카페 2,3월 매출감소율 :',100-((df_cafe_sum_2003/df_cafe_sum_2002)*100),'%')
```

20년 대구/경북 카페 2,3월 매출감소율 : 20.082558595764 04 %

```
print('20년 대구/경북 일반음식점업 1,2월 매출감소율 :',100-((df_food_sum_2002/df_food_sum_2001)*100),'%')
```

20년 대구/경북 일반음식점업 1,2월 매출감소율 : 28.86065578454165 %

```
print('20년 대구/경북 일반음식점업 2,3월 매출감소율 :',100-((df_food_sum_2003/df_food_sum_2002)*100),'%')
```

20년 대구/경북 일반음식점업 2,3월 매출감소율 : 31.562636891469694 %

```
print('20년 대구/경북 스포츠업 1,2월 매출감소율 :',100-((df_sport_sum_2002/df_sport_sum_2001)*100),'%')
```

20년 대구/경북 스포츠업 1,2월 매출감소율 : 33.9707965910755 %

```
print('20년 대구/경북 스포츠업 2,3월 매출감소율 :',100-((df_sport_sum_2003/df_sport_sum_2002)*100),'%')
```

20년 대구/경북 스포츠업 2,3월 매출감소율 : 1.8687284575058243 %

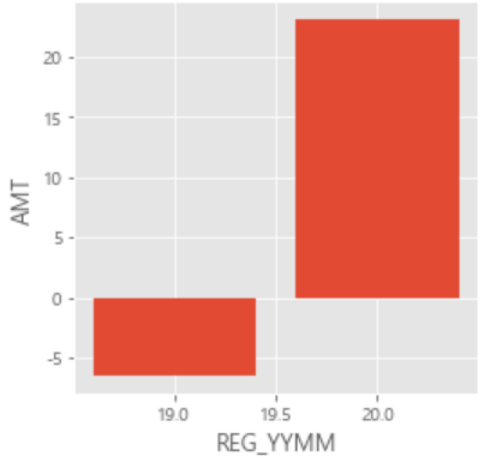
```
print('20년 대구/경북 전시 및 행사업 1,2월 매출감소율 :',100-((df_dis_sum_2002/df_dis_sum_2001)*100),'%')
```

20년 대구/경북 전시 및 행사업 1,2월 매출감소율 : 20.405423921057462 %

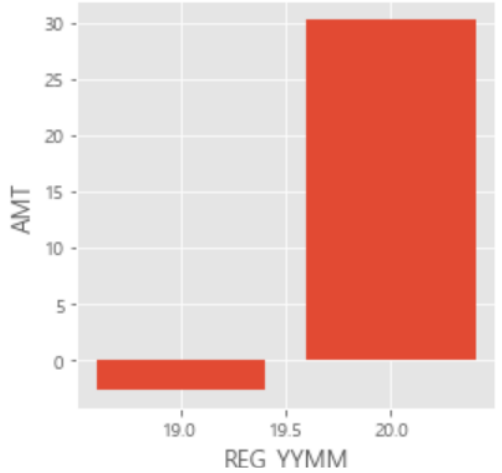
```
print('20년 대구/경북 전시 및 행사업 2,3월 매출감소율 :',100-((df_dis_sum_2003/df_dis_sum_2002)*100),'%')
```

20년 대구/경북 전시 및 행사업 2,3월 매출감소율 : 79.20315402433431 %

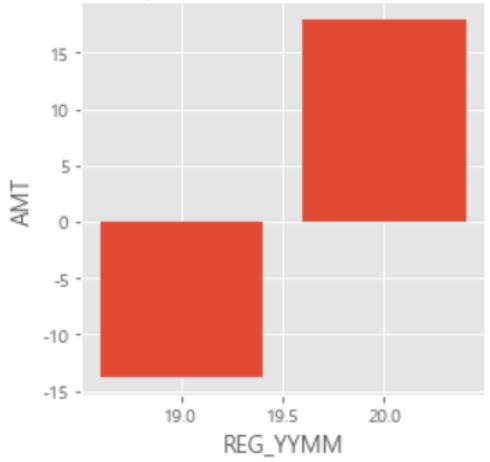
19/20년도 대구/경북 카페 매출감소량 평균



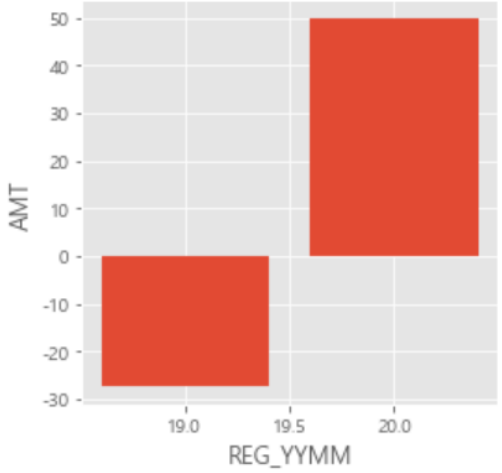
19/20년도 대구/경북 음식점업 매출감소량 평균



19/20년도 대구/경북 스포츠업 매출감소량 평균



19/20년도 대구/경북 전시 및 대행 사업 매출감소량 평균



기술통계량(box plot)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
from scipy import stats
```

```
from matplotlib import font_manager, rc
fn_name = font_manager.FontProperties(fname='c:/Windows/Fonts/malgun.ttf').get_name()
rc('font',family=fn_name)
```

```
df = pd.read_csv("1,2,3월_2.csv")
df['사용지_시도']='대구경북'
```

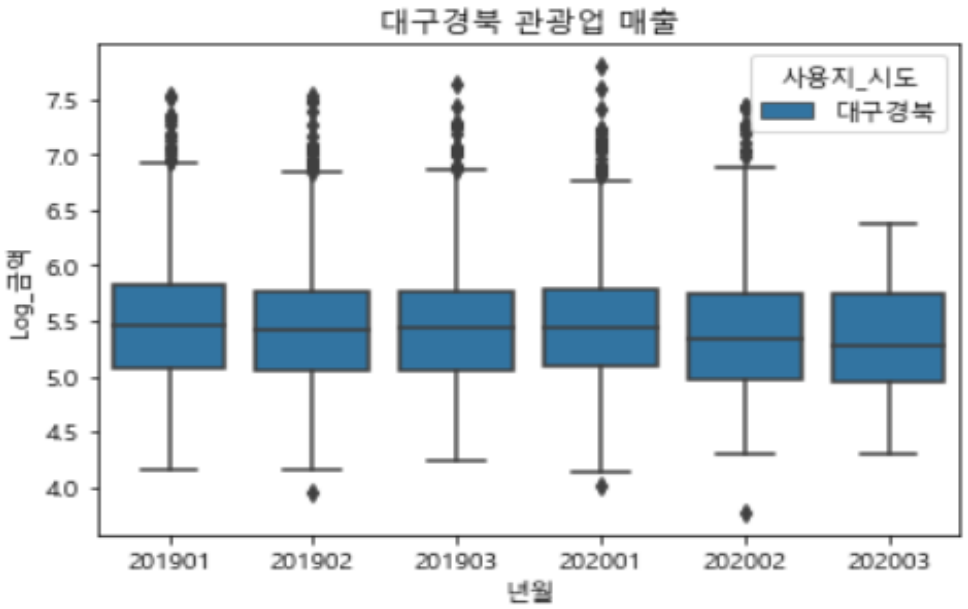
```
df_cafe=df[df['업종명']=='카페']
df_food=df[df['업종명']=='일반 음식점업']
df_tour=df[df['업종명']=='관광업']
df_sport=df[df['업종명']=='스포츠 및 레크레이션 용품 임대업']
df_dis=df[df['업종명']=='전시 및 행사 대행업']
```

```
df_tour_1901=df_tour[(df_tour['년월']==201901)]
df_tour_1901
df_tour_1901_금액=df_tour_1901['금액']
df_tour_sum_1901=sum(df_tour_1901_금액)
df_tour_1902=df_tour[(df_tour['년월']==201902)]
df_tour_1902
df_tour_1902_금액=df_tour_1902['금액']
df_tour_sum_1902=sum(df_tour_1902_금액)
df_tour_1903=df_tour[(df_tour['년월']==201903)]
df_tour_1903
df_tour_1903_금액=df_tour_1903['금액']
df_tour_sum_1903=sum(df_tour_1903_금액)
df_tour_2001=df_tour[(df_tour['년월']==202001)]
df_tour_2001
df_tour_2001_금액=df_tour_2001['금액']
df_tour_sum_2001=sum(df_tour_2001_금액)
df_tour_2002=df_tour[(df_tour['년월']==202002)]
df_tour_2002
df_tour_2002_금액=df_tour_2002['금액']
df_tour_sum_2002=sum(df_tour_2002_금액)
df_tour_2003=df_tour[(df_tour['년월']==202003)]
df_tour_2003
df_tour_2003_금액=df_tour_2003['금액']
df_tour_sum_2003=sum(df_tour_2003_금액)
```

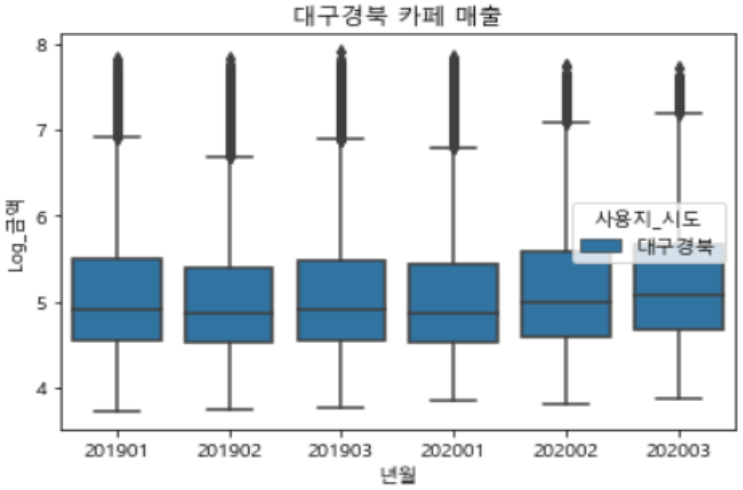
기술통계량(box plot)

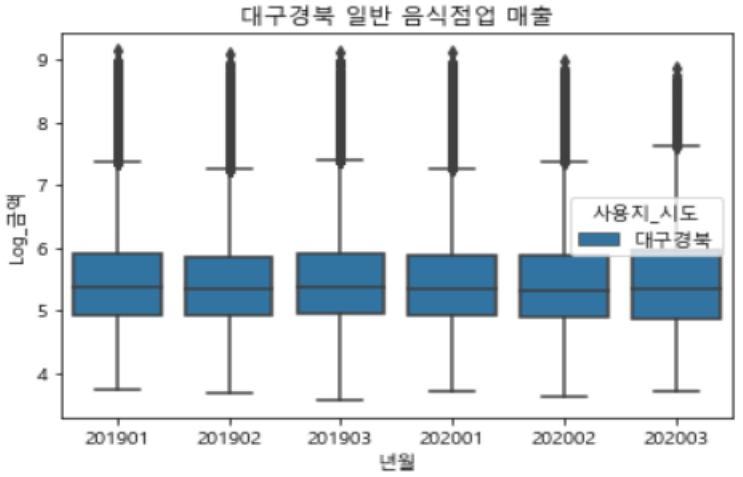
```
df_tour['Log_금액']=np.log10(df_tour['금액'])
tour=sns.boxplot(x='년월',y='Log_금액',hue='사용지_시도',data=df_tour)
tour.set_title('대구경북 관광업 매출')
plt.show()
```

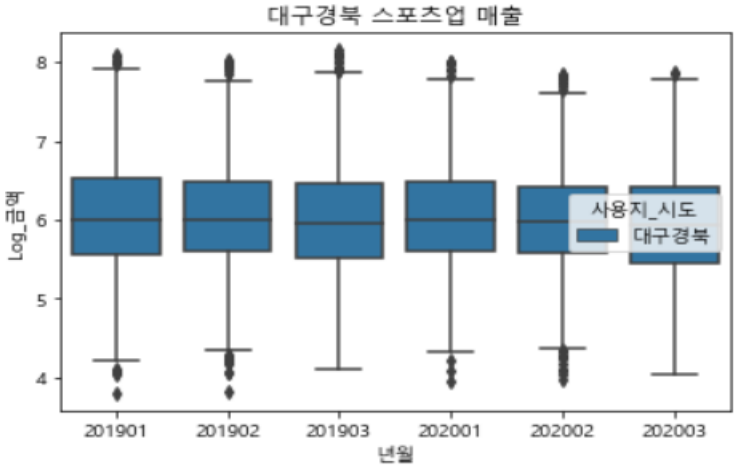
금액의 수치가 너무 커서
log10를 씌워 실행!

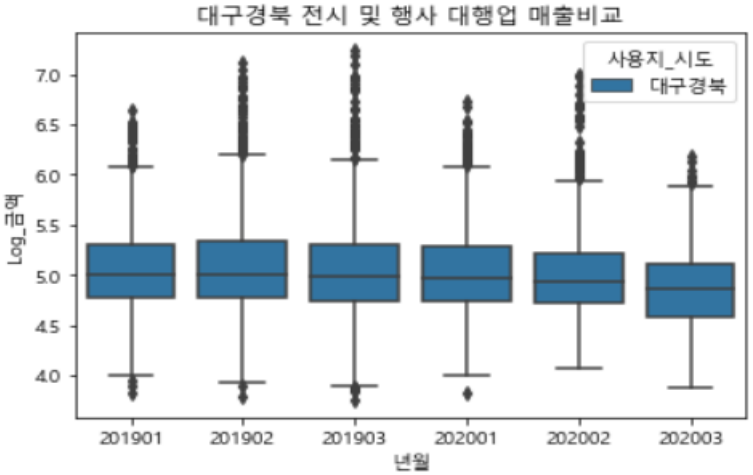


기술통계량(box plot)









기술통계량(pie chart)

```
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
```

```
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname="C:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)
```

```
plt.rcParams['axes.unicode_minus']=False
```

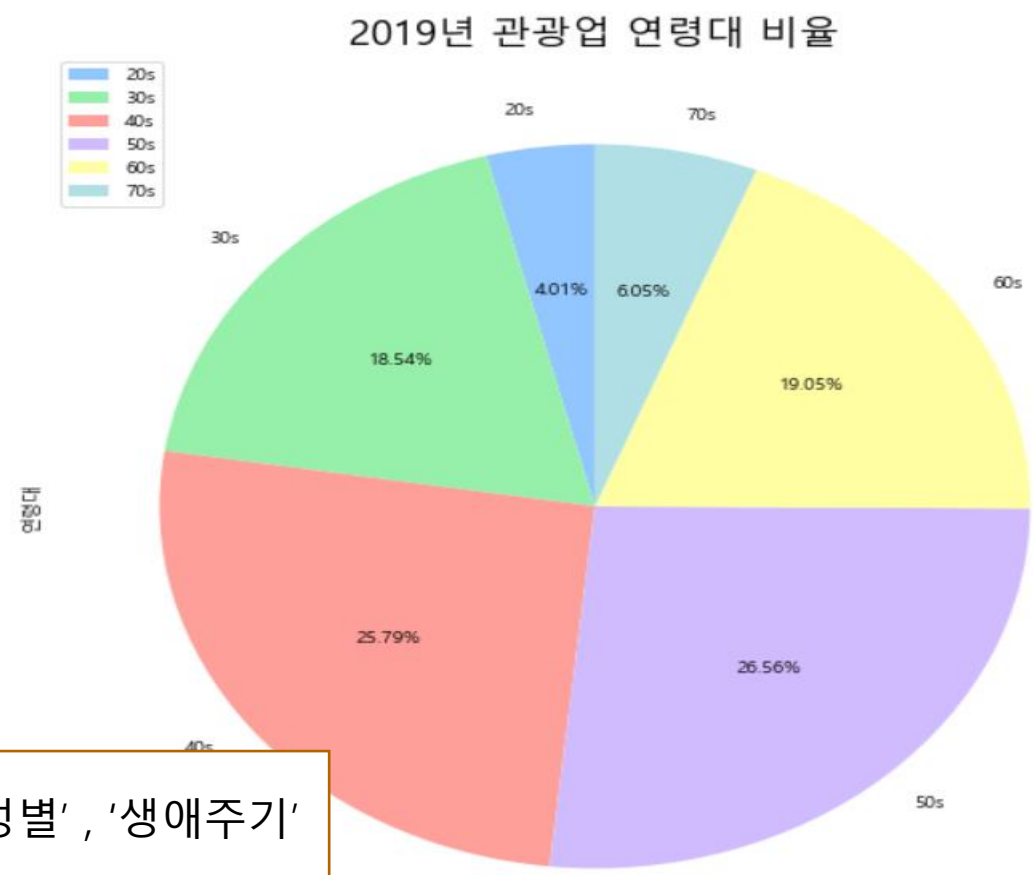
```
df_sub=df[['년월', '업종명', '연령대', '성별', '생애주기', '고객수', '금액','건수']]
```

```
df_a=df_sub
df_a['연령대'] = df_a['연령대'].str.rstrip('s').astype('int')
df_a
```

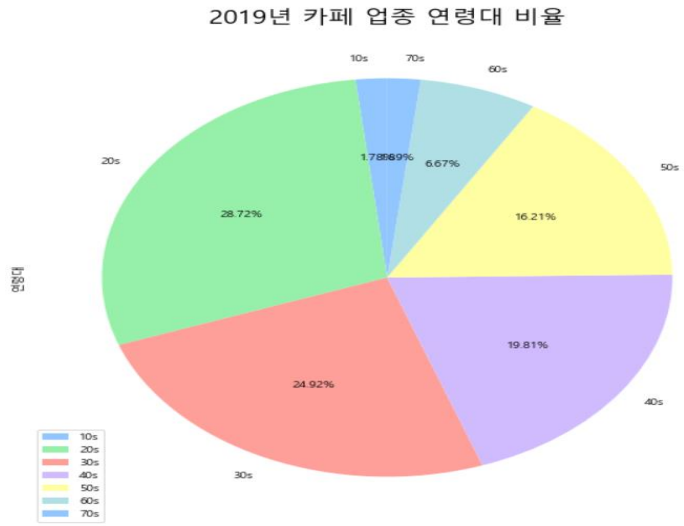
```
df_tour_19=df_tour[(df_tour['년월']==201901)|(df_tour['년월']==201902)|(df_tour['년월']==201903)]
df_tour_19
df_tour_19_금액=df_tour_19['금액']
df_tour_sum_19=sum(df_tour_19_금액)
df_tour_20=df_tour[(df_tour['년월']==202001)|(df_tour['년월']==202002)|(df_tour['년월']==202003)]
df_tour_20
df_tour_20_금액=df_tour_20['금액']
df_tour_sum_20=sum(df_tour_20_금액)
```

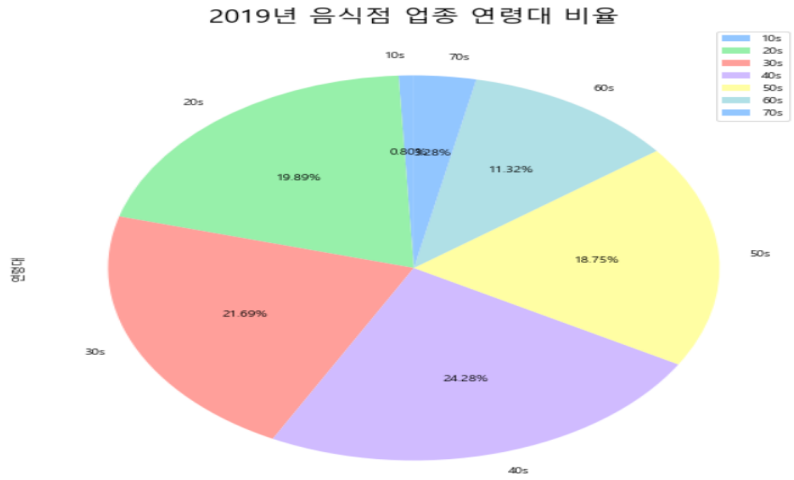
```
plt.style.use('seaborn-pastel')
df_tour_19_age.plot(kind = 'pie', figsize = (10,10), autopct = '%1.2f%%', startangle = 90, sub
plt.title('2019년 관광업 연령대 비율', size = 20)
plt.show()
```

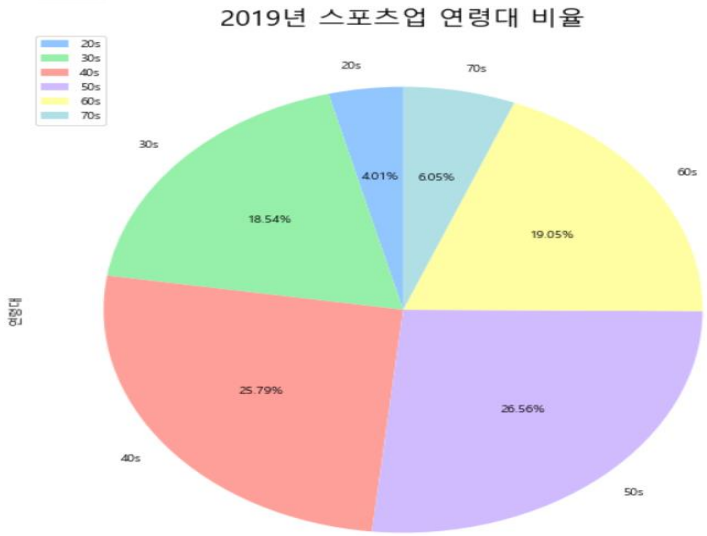
'연령대', '성별', '생애주기'



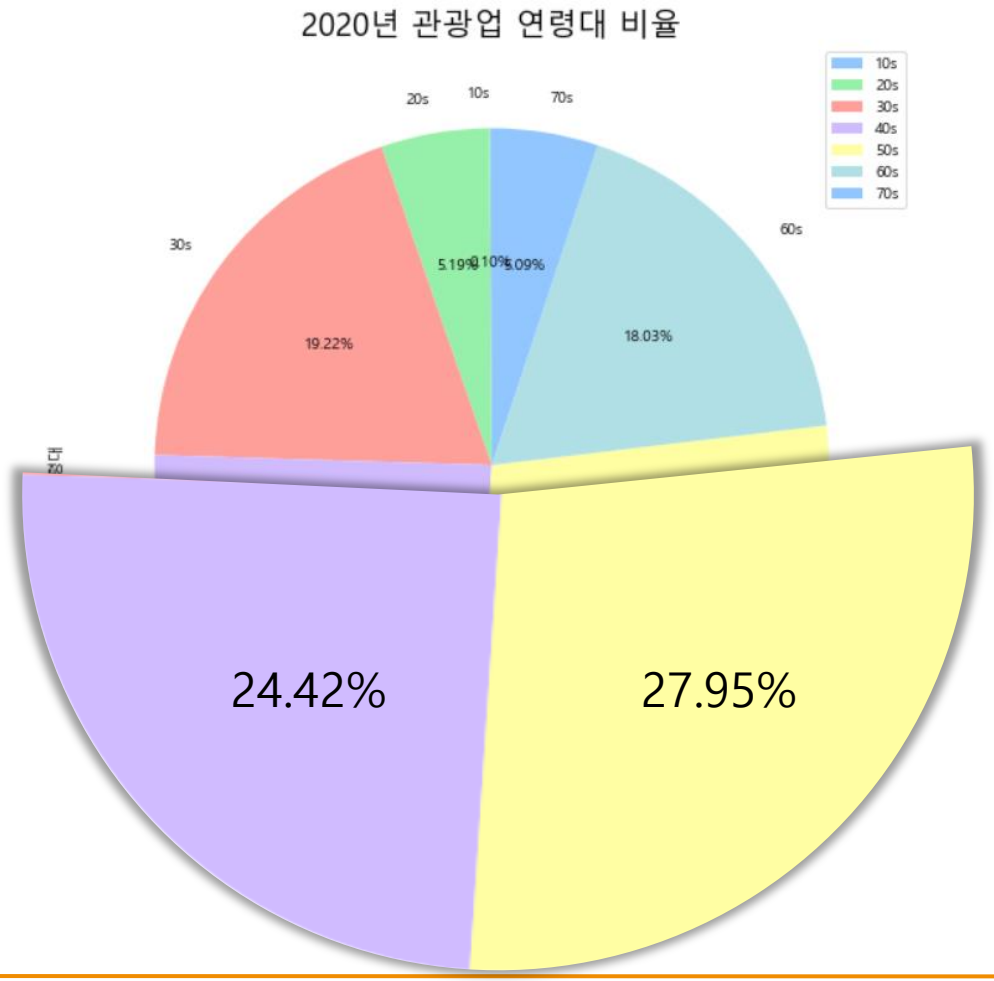
기술통계량(2019년 업종별 '연령대' pie chart)

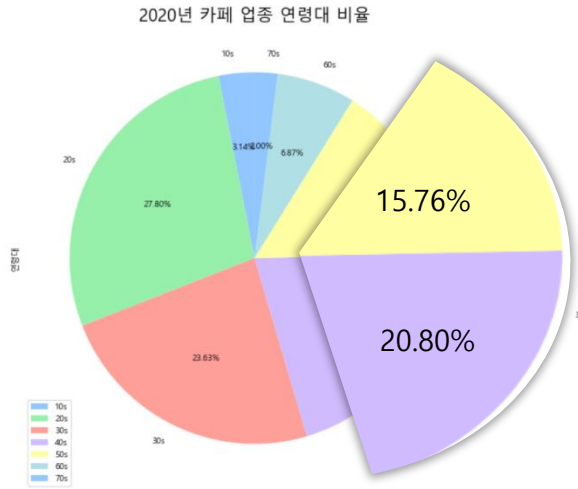


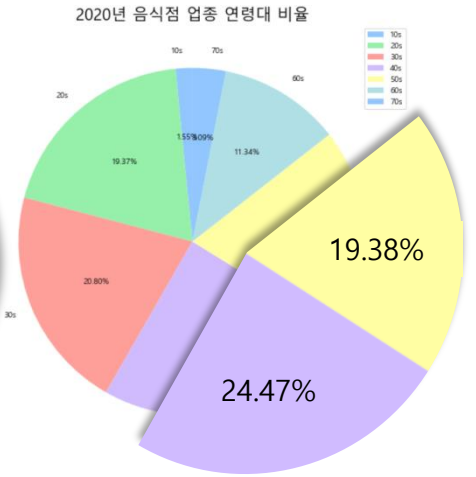


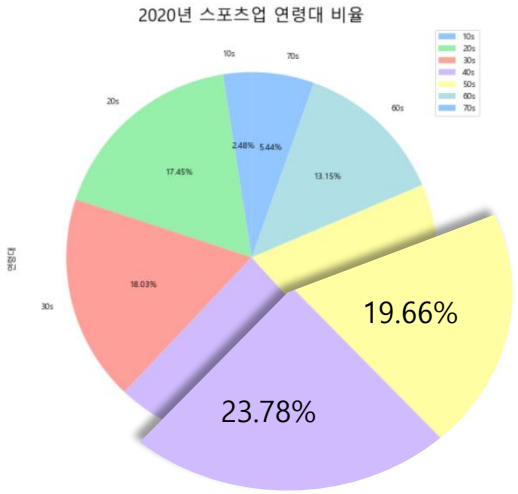


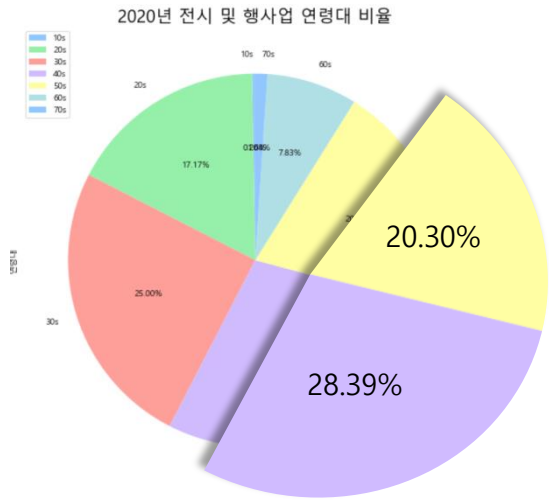
기술통계량(2020 업종별 '연령대' pie chart)



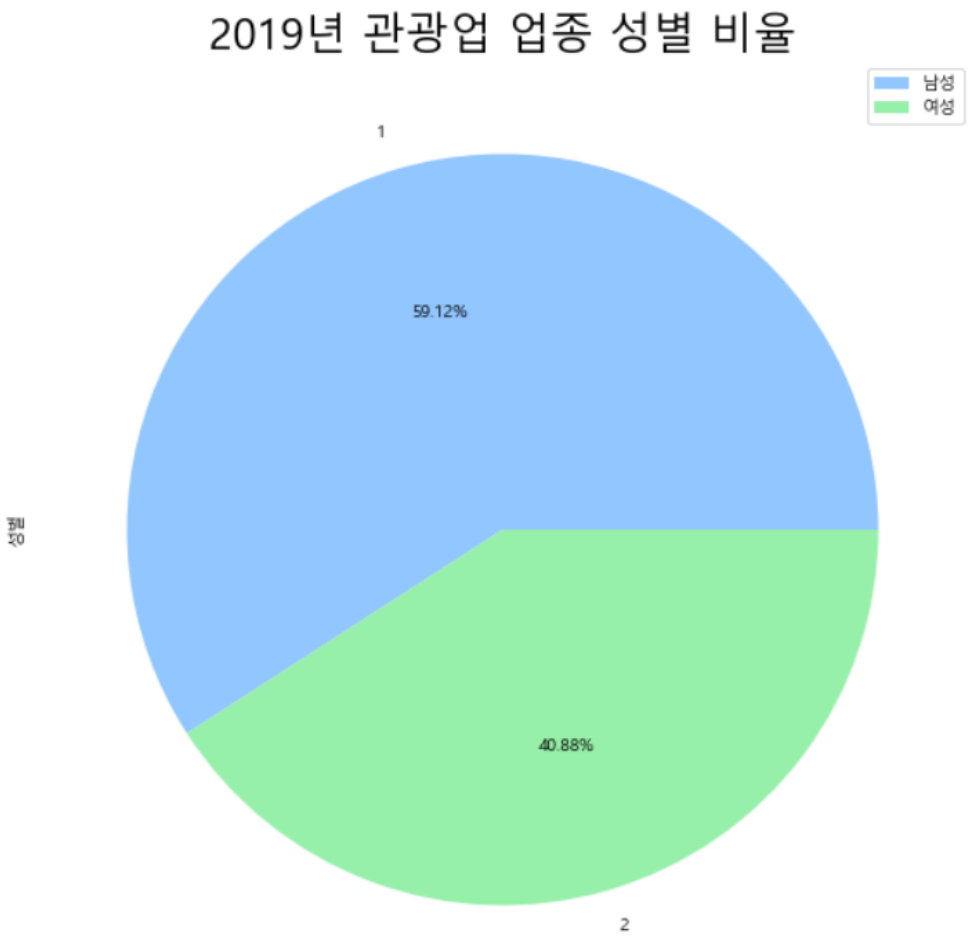


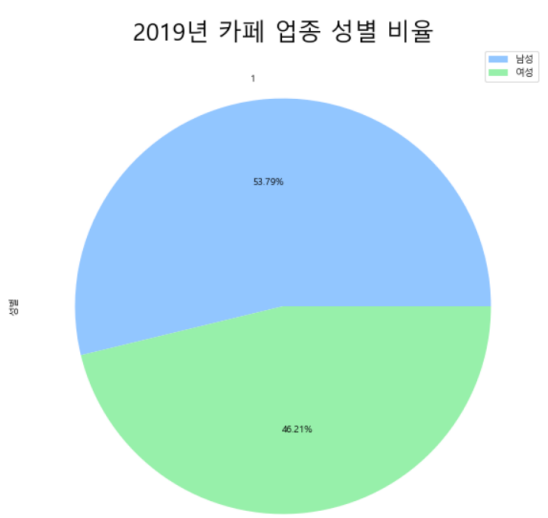


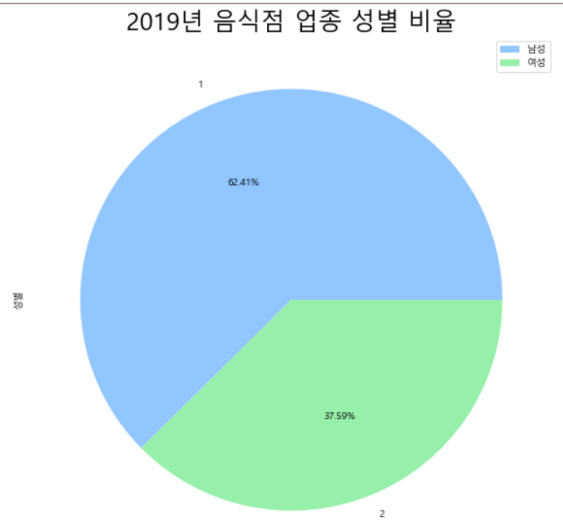


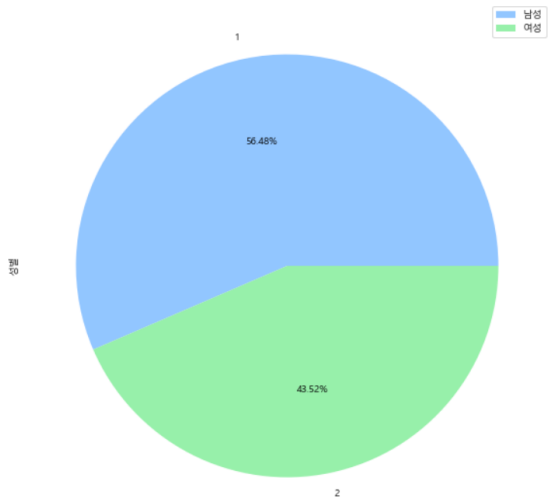


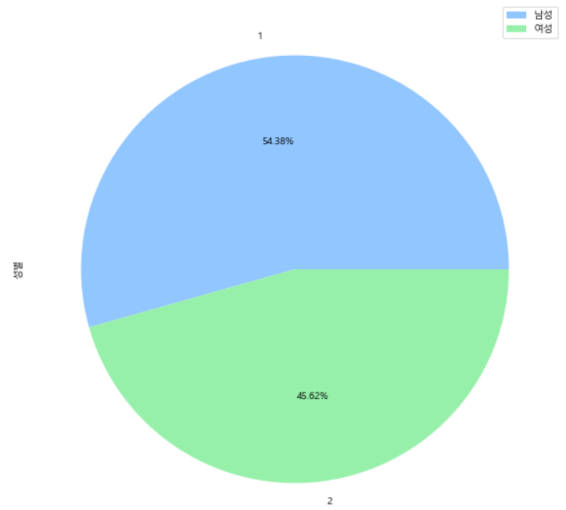
기술통계량(2019 업종별 '성별' pie chart)



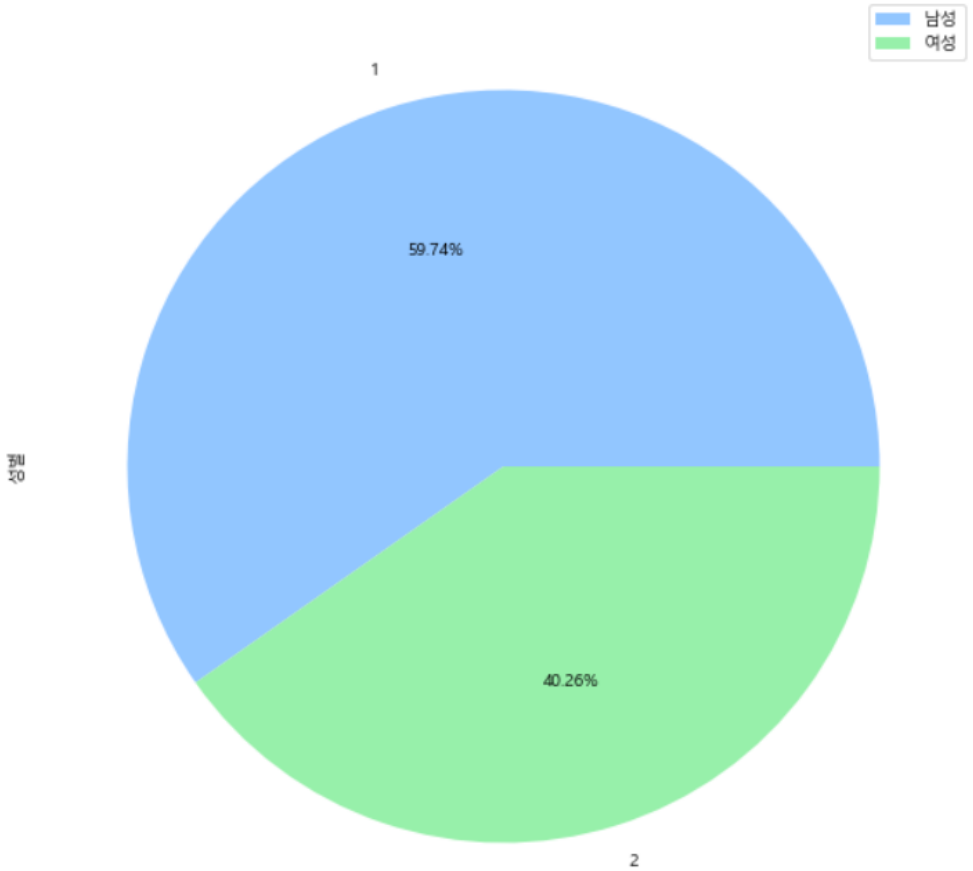


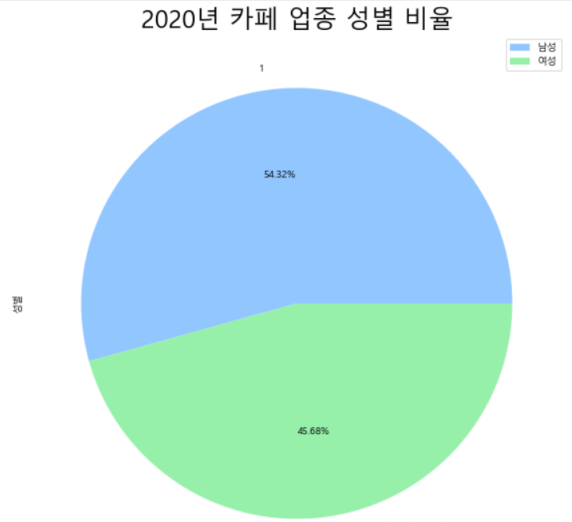


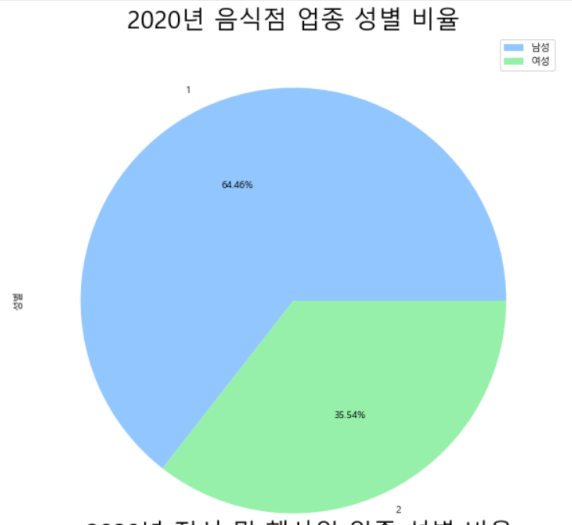


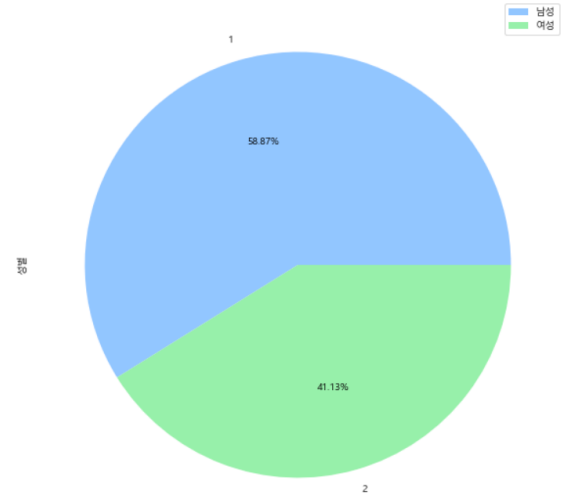


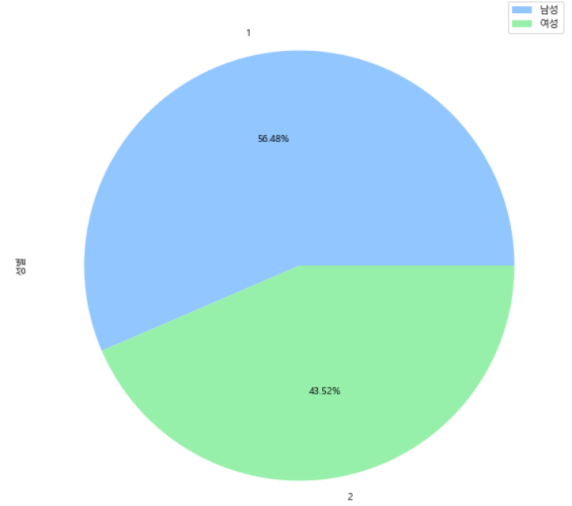
기술통계량(2020 업종별 '성별' pie chart)





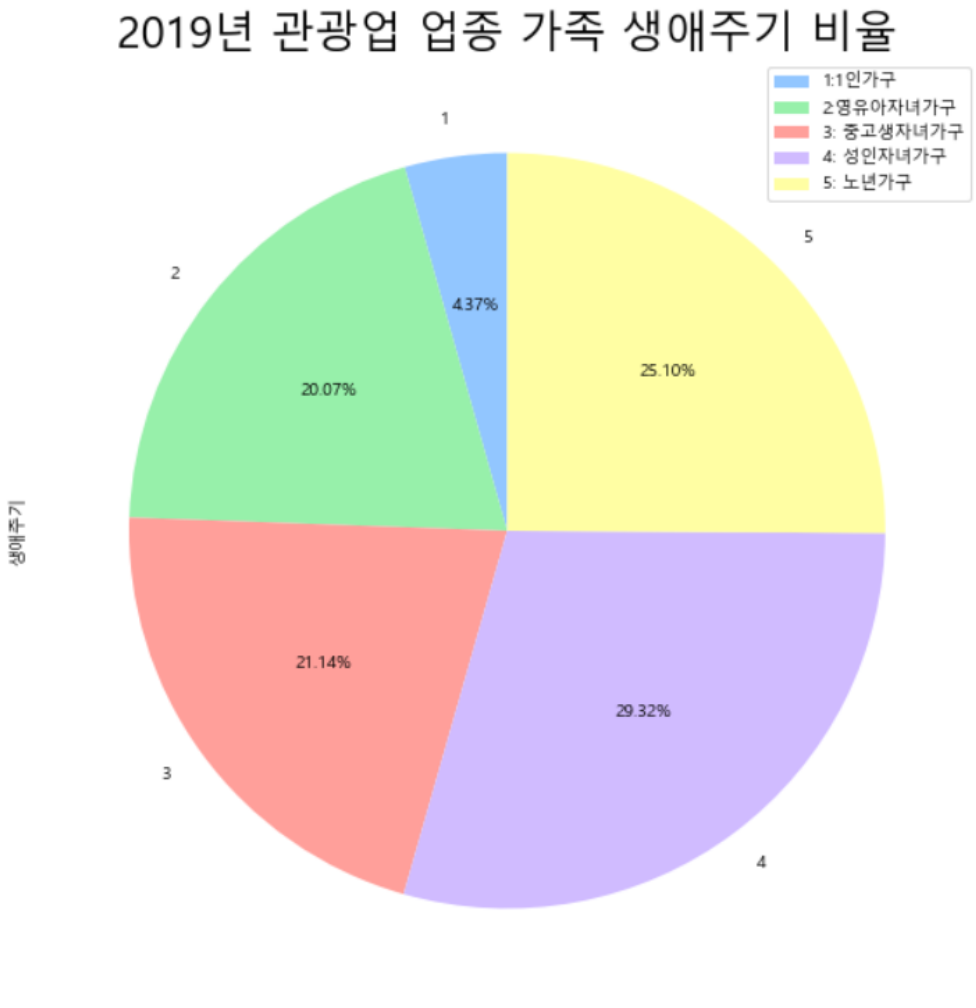


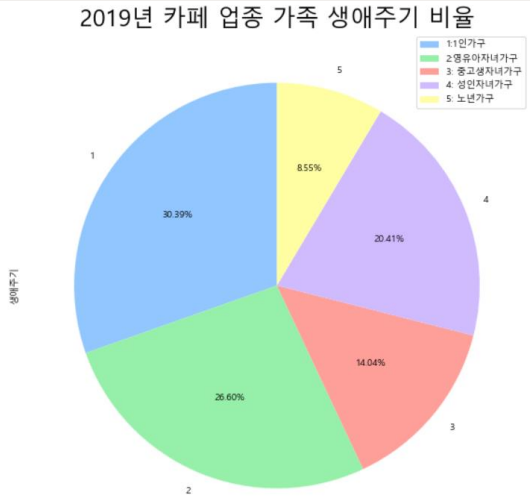


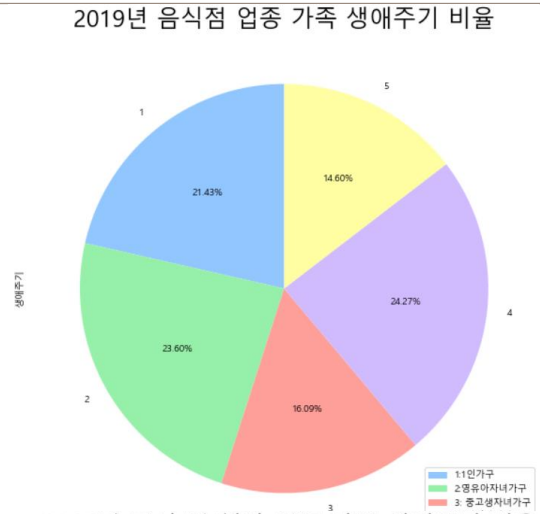


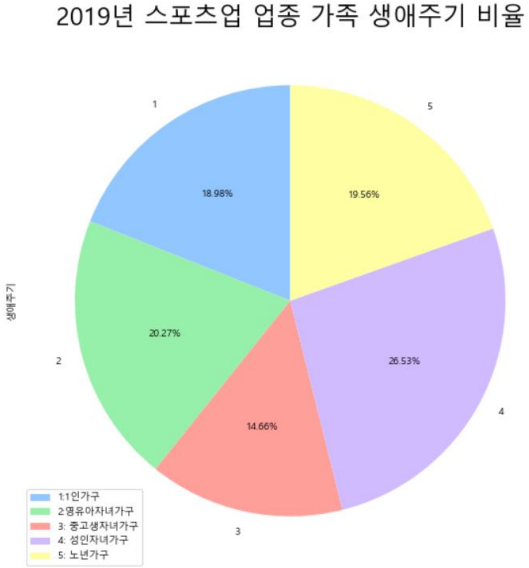
전체적으로 남성의 비율이 높게 나타났다

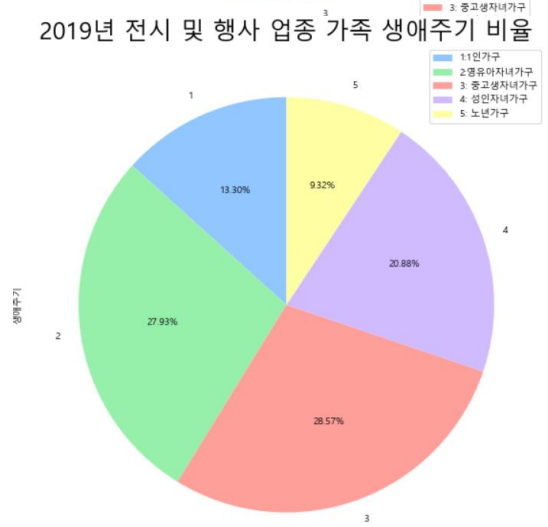
기술통계량(2019 업종별 '생애주기' pie chart)



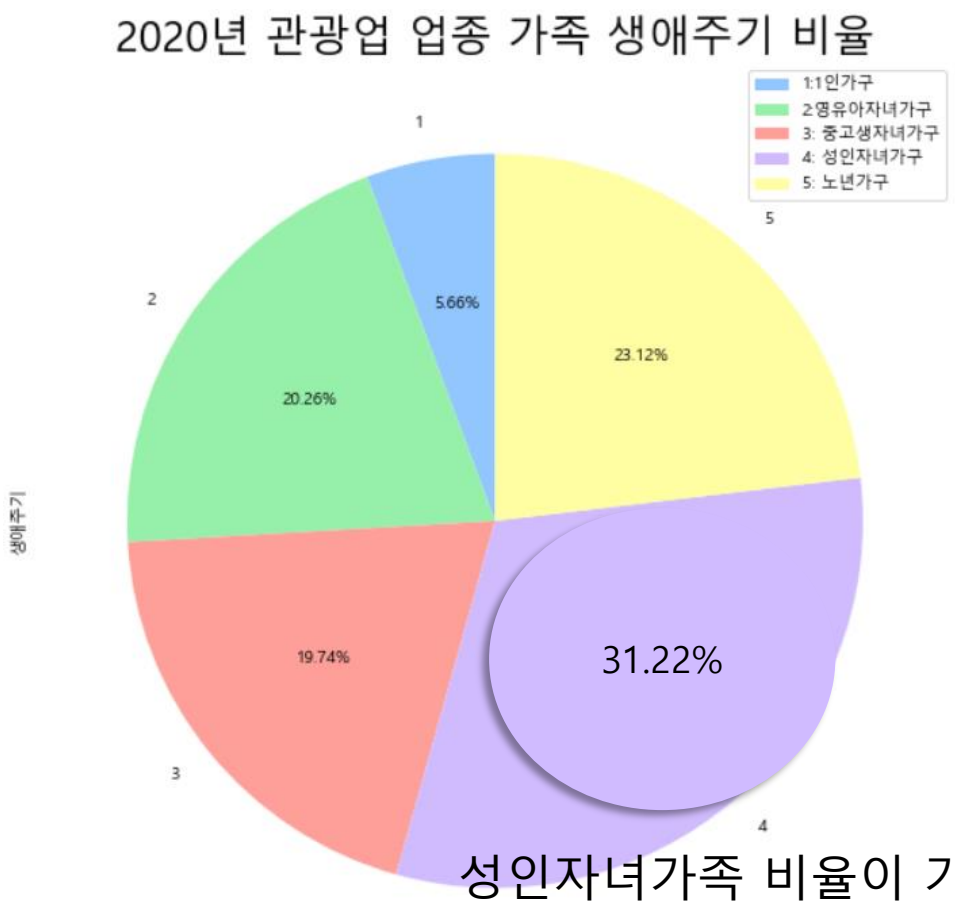




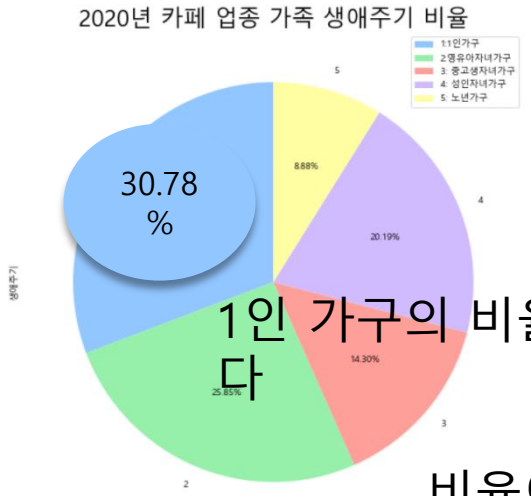




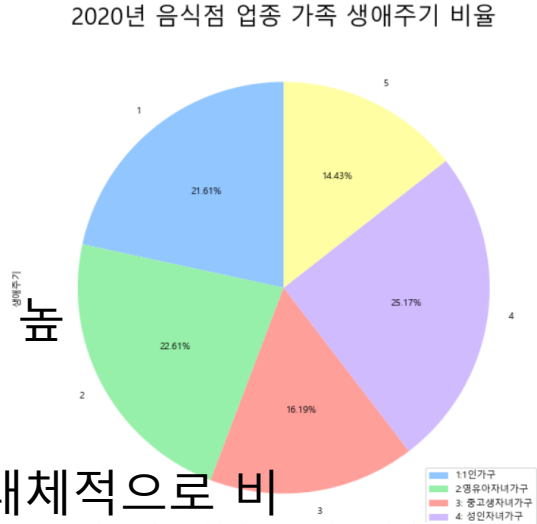
기술통계량(2020 업종별 '생애주기' pie chart)



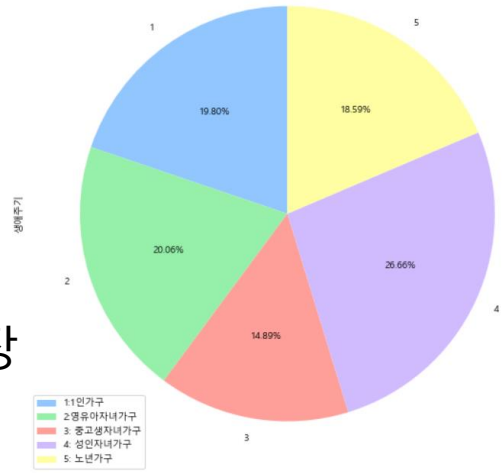
성인자녀가족 비율이 가장 높다

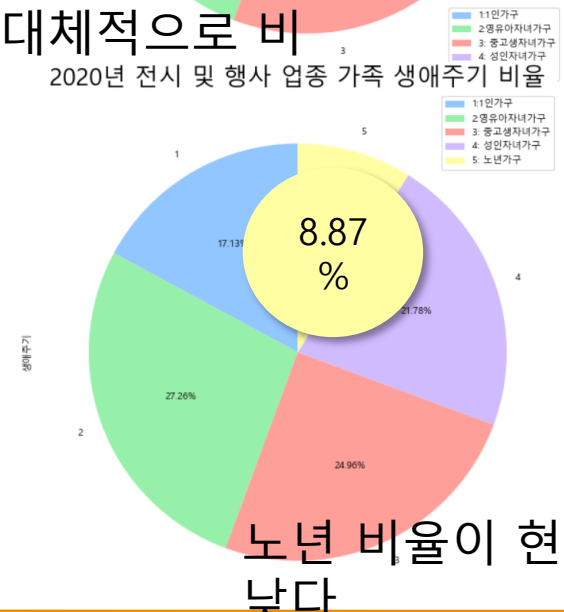


1인 가구의 비율이 높다



비율이 대체적으로 비슷





노년 비율이 현저히 낮다

PART 4, 소비자패턴분석

소비자패턴분석(전처리)

```
import pandas as pd
import numpy as np
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt
```

```
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname="C:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)
```

```
df = pd.read_csv('1,2,3월_2.csv')
```

```
df['연령대'] = df['연령대'].str.rstrip('s').astype('int')
```

```
df['Log10_금액'] = np.log10(df['금액'])
df
```

	Unnamed: 0	년월	업종명	연령대	성별	생애주기	고객수	금액	건수	Log10_금액
0	598167	201901	카페	20	2	1	3	24890	3	4.396025
1	598168	201901	카페	20	2	1	3	17800	3	4.250420
2	598169	201901	카페	30	2	2	3	28500	4	4.454845
3	598170	201901	카페	20	1	1	3	27900	4	4.445604
4	598171	201901	카페	40	1	3	3	20600	3	4.313867
...	...	...	...	...	...	...	...	...	...	...
224535	23899428	202003	일반 음식점업	20	1	1	8	904800	16	5.956553
224536	23899429	202003	일반 음식점업	30	1	1	3	331800	4	5.520876
224537	23899430	202003	일반 음식점업	30	1	2	7	1153000	28	6.061829
224538	23899431	202003	일반 음식점업	40	1	3	3	73000	3	4.863323
224539	23899432	202003	일반 음식점업	30	1	2	4	321000	5	5.506505

소비자패턴분석(전처리)

```
tour = df['업종명']=='관광업'
sports = df['업종명']=='스포츠 및 레크레이션 용품 임대업'
food = df['업종명']=='일반 음식점업'
cafe = df['업종명']=='카페'
dis = df['업종명']=='전시 및 행사 대행업'

df_201901 = df['년월']==201901
df_201902 = df['년월']==201902
df_201903 = df['년월']==201903
df_202001 = df['년월']==202001
df_202002 = df['년월']==202002
df_202003 = df['년월']==202003
```

```
# 카페
df_cafe = df[cafe]
df_cafe_1901 = df[cafe & df_201901]
df_cafe_1902 = df[cafe & df_201902]
df_cafe_1903 = df[cafe & df_201903]
df_cafe_2001 = df[cafe & df_202001]
df_cafe_2002 = df[cafe & df_202002]
df_cafe_2003 = df[cafe & df_202003]
```

19년, 20년 월별로 변수 정의

```
df_cafe_2003
```

	Unnamed: 0	년월	업종명	연령대	성별	생애주기	고객수	금액	건수	Log10_금액
72861	23922543	202003	카페	20	1	1	3	41060	6	4.613419
72862	23922544	202003	카페	20	2	1	3	25700	5	4.409933
72863	23922545	202003	카페	20	1	1	26	237900	32	5.376394
72864	23922546	202003	카페	20	1	2	5	43800	5	4.641474
72865	23922547	202003	카페	20	2	1	41	480170	54	5.681395
...	...	...	...	...	...	...	...	...	...	...
223758	23896787	202003	카페	30	1	2	3	29600	5	4.471292
223759	23896788	202003	카페	40	1	3	3	21800	7	4.338456
223760	23896789	202003	카페	20	2	1	3	7500	3	3.875061
223761	23896790	202003	카페	20	1	1	6	54900	7	4.739572
223762	23896791	202003	카페	20	2	1	3	53000	7	4.724276

소비자패턴분석(전처리)

	Unnamed: 0	년월	업종명	연령대	성별	생애주기	고객수	금액	건수	Log10_금액
72861	23922543	202003	카페	20	1	1	3	41060	6	4.613419
72862	23922544	202003	카페	20	2	1	3	25700	5	4.409933
72863	23922545	202003	카페	20	1	1	26	237900	32	5.376394
72864	23922546	202003	카페	20	1	2	5	43800	5	4.641474
72865	23922547	202003	카페	20	2	1	41	480170	54	5.681395
...	...	...	...	...	...	...	...	...	...	...
223758	23896787	202003	카페	30	1	2	3	29600	5	4.471292
223759	23896788	202003	카페	40	1	3	3	21800	7	4.338456
223760	23896789	202003	카페	20	2	1	3	7500	3	3.875061
223761	23896790	202003	카페	20	1	1	6	54900	7	4.739572
223762	23896791	202003	카페	20	2	1	3	53000	7	4.724276

연령대, 고객 수, 건수, Log10금액을 선택하여 t-test 진행!

소비자패턴분석(t-test 금액)

카페 1월

```
df_c_1901_1 = np.array(df_cafe_1901['Log10_금액'])
df_c_2001_1 = np.array(df_cafe_2001['Log10_금액'])
print('201901 평균 : ',np.mean(df_c_1901_1))
print('202001 평균 : ',np.mean(df_c_2001_1))
```

```
201901 평균 :  5.098097934016562
202001 평균 :  5.064275744654157
```

```
stats.levene(df_c_1901_1, df_c_2001_1)
```

```
LeveneResult(statistic=2.120779681949321, pvalue=0.14533592236532547)
```

```
stats.ttest_ind(df_c_1901_1, df_c_2001_1, equal_var=True)
```

```
Ttest_indResult(statistic=2.729191628793607, pvalue=0.0063572439586522915)
```

카페 2월

```
df_c_1902_1 = np.array(df_cafe_1902['Log10_금액'])
df_c_2002_1 = np.array(df_cafe_2002['Log10_금액'])
print('201901 평균 : ',np.mean(df_c_1902_1))
print('202002 평균 : ',np.mean(df_c_2002_1))
```

```
201901 평균 :  5.0483683414481835
202002 평균 :  5.1556312886380375
```

```
stats.levene(df_c_1902_1, df_c_2002_1)
```

```
LeveneResult(statistic=22.200998283809334, pvalue=2.4833890725977257e-06)
```

```
stats.ttest_ind(df_c_1902_1, df_c_2002_1, equal_var=False)
```

```
Ttest_indResult(statistic=-8.060985099856008, pvalue=8.397364457197866e-16)
```

카페 3월

```
df_c_1903_1 = np.array(df_cafe_1903['Log10_금액'])
df_c_2003_1 = np.array(df_cafe_2003['Log10_금액'])
print('201903 평균 : ',np.mean(df_c_1903_1))
print('202003 평균 : ',np.mean(df_c_2003_1))
```

```
201903 평균 :  5.09509369831541
202003 평균 :  5.222014859135214
```

```
stats.levene(df_c_1903_1, df_c_2003_1)
```

```
LeveneResult(statistic=4.116625771582053, pvalue=0.04248820956197639)
```

```
stats.ttest_ind(df_c_1903_1, df_c_2003_1, equal_var=False)
```

```
Ttest_indResult(statistic=-8.794643765192566, pvalue=1.7359425280512503e-18)
```

1,2,3월 모두 pvalue 값이 0.05보다 낮다 → 금액의 평균 차이가 있다

소비자패턴분석(t-test 연령대)

카페 1월

```
df_c_1901_2 = np.array(df_cafe_1901['연령대'])
df_c_2001_2 = np.array(df_cafe_2001['연령대'])
print('201901 평균 : ',np.mean(df_c_1901_2))
print('202001 평균 : ',np.mean(df_c_2001_2))
```

```
201901 평균 :  35.287596401028274
202001 평균 :  34.74486966167498
```

```
stats.levene(df_c_1901_2, df_c_2001_2)
```

```
LeveneResult(statistic=0.9882761413157473, pvalue=0.32018197117556835)
```

```
stats.ttest_ind(df_c_1901_2, df_c_2001_2, equal_var=True)
```

```
Ttest_indResult(statistic=2.271231090295986, pvalue=0.02314874301754909)
```

카페 2월

```
df_c_1902_2 = np.array(df_cafe_1902['연령대'])
df_c_2002_2 = np.array(df_cafe_2002['연령대'])
print('201902 평균 : ',np.mean(df_c_1902_2))
print('202002 평균 : ',np.mean(df_c_2002_2))
```

```
201902 평균 :  34.70579710144928
202002 평균 :  34.387631366208566
```

```
stats.levene(df_c_1902_2, df_c_2002_2)
```

```
LeveneResult(statistic=22.91394639806433, pvalue=1.7146637160056196e-06)
```

```
stats.ttest_ind(df_c_1902_2, df_c_2002_2, equal_var=False)
```

```
Ttest_indResult(statistic=1.2296468591116396, pvalue=0.2188574982750557)
```

카페 3월

```
df_c_1903_2 = np.array(df_cafe_1903['연령대'])
df_c_2003_2 = np.array(df_cafe_2003['연령대'])
print('201903 평균 : ',np.mean(df_c_1903_2))
print('202003 평균 : ',np.mean(df_c_2003_2))
```

```
201903 평균 :  34.30984695350852
202003 평균 :  34.961139896373055
```

```
stats.levene(df_c_1903_2, df_c_2003_2)
```

```
LeveneResult(statistic=3.0670919002725747, pvalue=0.07992069877072126)
```

```
stats.ttest_ind(df_c_1903_2, df_c_2003_2, equal_var=True)
```

```
Ttest_indResult(statistic=-2.3213100037040966, pvalue=0.020288662540453556)
```

2월만 pvalue 값이 0.05보다 크게 나타난다 → 금액의 평균 차이가 없다

소비자패턴분석(t-test 이용건수)

카페 1월

```
df_c_1901_3 = np.array(df_cafe_1901['건수'])
df_c_2001_3 = np.array(df_cafe_2001['건수'])
print('201901 평균 : ',np.mean(df_c_1901_3))
print('202001 평균 : ',np.mean(df_c_2001_3))
```

```
201901 평균 :  103.12596401028277
202001 평균 :  101.54783693843594
```

```
stats.levene(df_c_1901_3, df_c_2001_3)
```

```
LeveneResult(statistic=0.026081736671766127, pvalue=0.871703291578059)
```

```
stats.ttest_ind(df_c_1901_3, df_c_2001_3, equal_var=True)
```

```
Ttest_indResult(statistic=0.1959608538776967, pvalue=0.8446553291458273)
```

카페 2월

```
df_c_1902_3 = np.array(df_cafe_1902['건수'])
df_c_2002_3 = np.array(df_cafe_2002['건수'])
print('201902 평균 : ',np.mean(df_c_1902_3))
print('202002 평균 : ',np.mean(df_c_2002_3))
```

```
201902 평균 :  89.54579710144928
202002 평균 :  114.53556992724333
```

```
stats.levene(df_c_1902_3, df_c_2002_3)
```

```
LeveneResult(statistic=8.888774945397536, pvalue=0.0028751253797117504)
```

```
stats.ttest_ind(df_c_1902_3, df_c_2002_3, equal_var=False)
```

```
Ttest_indResult(statistic=-3.0400063314624046, pvalue=0.0023718543495278304)
```

카페 3월

```
df_c_1903_3 = np.array(df_cafe_1903['건수'])
df_c_2003_3 = np.array(df_cafe_2003['건수'])
print('201903 평균 : ',np.mean(df_c_1903_3))
print('202003 평균 : ',np.mean(df_c_2003_3))
```

```
201903 평균 :  110.11651747040139
202003 평균 :  124.42487046632124
```

```
stats.levene(df_c_1903_3, df_c_2003_3)
```

```
LeveneResult(statistic=1.782265711072532, pvalue=0.18189953739148224)
```

```
stats.ttest_ind(df_c_1903_3, df_c_2003_3, equal_var=True)
```

```
Ttest_indResult(statistic=-1.4592114129431242, pvalue=0.14453605483195406)
```

2월만 pvalue 값이 0.05보다 작게 나타난다 → 금액의 평균 차이가 있다

소비자패턴분석(t-test 고객 수)

카페 1월

```
df_c_1901_4 = np.array(df_cafe_1901['고객수'])
df_c_2001_4 = np.array(df_cafe_2001['고객수'])
print('201901 평균 : ',np.mean(df_c_1901_4))
print('202001 평균 : ',np.mean(df_c_2001_4))
```

```
201901 평균 :  69.83901028277634
202001 평균 :  66.87160288408208
```

```
stats.levene(df_c_1901_4, df_c_2001_4)
```

```
LeveneResult(statistic=0.3141975524192457, pvalue=0.5751250973064959)
```

```
stats.ttest_ind(df_c_1901_4, df_c_2001_4, equal_var=True)
```

```
Ttest_indResult(statistic=0.586861828092420, pvalue=0.5573063698967304)
```

1월은 pvalue 값이 0.05보다 크게 나타난다.
→ 금액의 평균 차이가 없다

카페 2월

```
df_c_1902_4 = np.array(df_cafe_1902['고객수'])
df_c_2002_4 = np.array(df_cafe_2002['고객수'])
print('201902 평균 : ',np.mean(df_c_1902_4))
print('202002 평균 : ',np.mean(df_c_2002_4))
```

```
201902 평균 :  62.65449275362319
202002 평균 :  74.37247372675829
```

```
stats.levene(df_c_1902_4, df_c_2002_4)
```

```
LeveneResult(statistic=4.772703825778774, pvalue=0.028933912233557447)
```

```
stats.ttest_ind(df_c_1902_4, df_c_2002_4, equal_var=False)
```

```
Ttest_indResult(statistic=-2.2542973260419483, pvalue=0.024198082303582467)
```

2월만 pvalue 값이 0.05보다 작게 나타난다.
→ 금액의 평균 차이가 있다

카페 3월

```
df_c_1903_4 = np.array(df_cafe_1903['고객수'])
df_c_2003_4 = np.array(df_cafe_2003['고객수'])
print('201903 평균 : ',np.mean(df_c_1903_4))
print('202003 평균 : ',np.mean(df_c_2003_4))
```

```
201903 평균 :  71.7014149581288
202003 평균 :  75.66943005181348
```

```
stats.levene(df_c_1903_4, df_c_2003_4)
```

```
LeveneResult(statistic=0.31466232406303324, pvalue=0.5748448446675191)
```

```
stats.ttest_ind(df_c_1903_4, df_c_2003_4, equal_var=True)
```

```
Ttest_indResult(statistic=-0.6761165988785088, pvalue=0.49898111806974776)
```

3월은 pvalue 값이 0.05보다 크게 나타난다.
→ 금액의 평균 차이가 없다

소비자패턴분석(t-test)

전시회 1월

```
df_d_1901_1 = np.array(df_dis_1901['Log10_금액'])
df_d_2001_1 = np.array(df_dis_2001['Log10_금액'])
print('201901 평균 : ',np.mean(df_d_1901_1))
print('202001 평균 : ',np.mean(df_d_2001_1))
```

```
201901 평균 :  5.066695356872308
202001 평균 :  5.032321569221859
```

```
stats.levene(df_d_1901_1, df_d_2001_1)
```

```
LeveneResult(statistic=0.23403816623854443, pvalue=0.6285888211977807)
```

```
stats.ttest_ind(df_d_1901_1, df_d_2001_1, equal_var=True)
```

```
Ttest_indResult(statistic=1.988898538391989, pvalue=0.046823367886434215)
```

스포츠 1월

```
df_s_1901_1 = np.array(df_sports_1901['Log10_금액'])
df_s_2001_1 = np.array(df_sports_2001['Log10_금액'])
print('201901 평균 : ',np.mean(df_s_1901_1))
print('202001 평균 : ',np.mean(df_s_2001_1))
```

```
201901 평균 :  6.053863178577422
202001 평균 :  6.047532410746838
```

```
stats.levene(df_s_1901_1, df_s_2001_1)
```

```
LeveneResult(statistic=5.408900464709576, pvalue=0.020086756719661928)
```

```
stats.ttest_ind(df_s_1901_1, df_s_2001_1, equal_var=False)
```

```
Ttest_indResult(statistic=0.28051188874156546, pvalue=0.7791001206047452)
```

관광업 1월

```
df_t_1901_4 = np.array(df_tour_1901['고객수'])
df_t_2001_4 = np.array(df_tour_2001['고객수'])
print('201901 평균 : ',np.mean(df_t_1901_4))
print('202001 평균 : ',np.mean(df_t_2001_4))
```

```
201901 평균 :  8.956378600823045
202001 평균 :  8.9265625
```

```
stats.levene(df_t_1901_4, df_t_2001_4)
```

```
LeveneResult(statistic=0.00985184479897685, pvalue=0.920942627928422)
```

```
stats.ttest_ind(df_t_1901_4, df_t_2001_4, equal_var=True)
```

```
Ttest_indResult(statistic=0.06103698490573511, pvalue=0.95133464680
```

일반 음식점업 1월

```
df_f_1901_1 = np.array(df_food_1901['Log10_금액'])
df_f_2001_1 = np.array(df_food_2001['Log10_금액'])
print('201901 평균 : ',np.mean(df_f_1901_1))
print('202001 평균 : ',np.mean(df_f_2001_1))
```

```
201901 평균 :  5.486198127699646
202001 평균 :  5.4665891428578375
```

```
stats.levene(df_f_1901_1, df_f_2001_1)
```

```
LeveneResult(statistic=15.023572291318024, pvalue=0.0001062846624850121 3)
```

```
stats.ttest_ind(df_f_1901_1, df_f_2001_1, equal_var=False)
```

```
Ttest_indResult(statistic=3.212253894316517, pvalue=0.0013176482190835998)
```

소비자패턴분석(t-test결과정리)

금액

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 있다 (2020년 3월의 매출액이 더 적다)

관광업

- 1월 : 차이가 있다 (2020년 1월의 매출액이 더 적다)
- 2월 : 차이가 있다 (2020년 2월의 매출액이 더 적다)
- 3월 : 차이가 있다 (2020년 3월의 매출액이 더 적다)

전시 및 대행 사업

- 1월 : 차이가 있다 (2020년 1월의 매출액이 더 적다)
- 2월 : 차이가 있다 (2020년 2월의 매출액이 더 적다)
- 3월 : 차이가 있다 (2020년 3월의 매출액이 더 적다)

카페

- 1월 : 차이가 없다
- 2월 : 차이가 있다 (2020년 2월의 매출액이 더 적다)
- 3월 : 차이가 있다 (2020년 3월의 매출액이 더 적다)

스포츠업

- 1월 : 차이가 있다 (2020년 3월의 매출액이 더 적다)
- 2월 : 차이가 없다
- 3월 : 차이가 없다

일반음식점업

연령대

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 있다 (2020년 3월의 연령대가 더 높다)

관광업

- 1월 : 차이가 없다.
- 2월 : 차이가 있다 (2020년 2월의 연령대가 더 낮다)
- 3월 : 차이가 있다 (2020년 3월의 연령대가 더 낮다)

전시 및 대행 사업

- 1월 : 차이가 있다 (2020년 1월의 연령대가 더 낮다)
- 2월 : 차이가 없다
- 3월 : 차이가 있다 (2020년 3월의 연령대가 더 높다)

카페

- 1월 : 차이가 있다 (2020년 1월의 연령대가 더 낮다)
- 2월 : 차이가 없다
- 3월 : 차이가 없다

스포츠업

- 1월 : 차이가 있다 (2020년 1월의 연령대가 더 낮다)
- 2월 : 차이가 없다
- 3월 : 차이가 있다 (2020년 3월의 연령대가 더 높다)

일반음식점업

소비자패턴분석(t-test결과정리)

이용건수

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 없다

관광업

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 없다

전시 및 대행 사업

- 1월 : 차이가 없다
- 2월 : 차이가 있다 (2020년 2월의 사용 건수가 더 많다)
- 3월 : 차이가 없다

카페

- 1월 : 차이가 없다
- 2월 : 차이가 있다 (2020년 2월의 사용 건수가 더 적다)
- 3월 : 차이가 있다 (2020년 3월의 사용 건수가 더 적다)

스포츠업

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 없다

일반음식점업

고객수

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 없다

관광업

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 없다

전시 및 대행 사업

- 1월 : 차이가 없다
- 2월 : 차이가 있다 (2020년 2월의 고객수가 더 많다)
- 3월 : 차이가 없다

카페

- 1월 : 차이가 없다
- 2월 : 차이가 있다 (2020년 2월의 고객수가 더 적다)
- 3월 : 차이가 있다 (2020년 3월의 고객수가 더 적다)

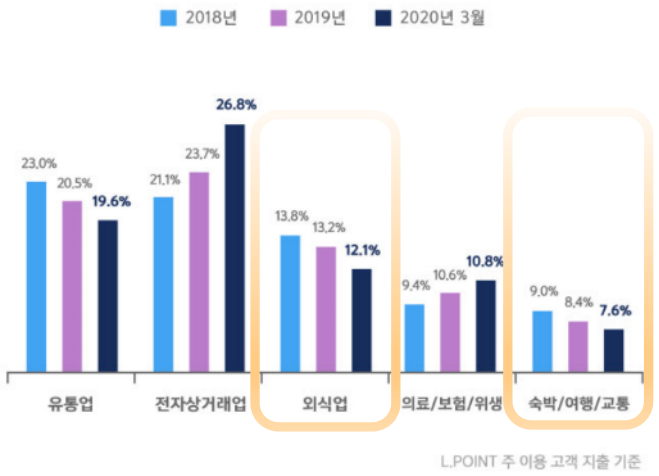
스포츠업

- 1월 : 차이가 없다
- 2월 : 차이가 없다
- 3월 : 차이가 없다

일반음식점업

소비자패턴분석(t-test 결론)

1. 업종별 지출 구성비 변화



L.POINT 주 이용 고객 지출 기준

3. 연령대별 지출 구성비 변화

	20대 이하	30대	40대	50대 이상
전자상거래업	+3.9%p	+4.5%p	+3.0%p	+2.4%p
유통업	+0.5%p	-1.4%p	-1.2%p	-0.5%p
의료/보험/위생	+0.1%p	0.0%p	+0.3%p	+0.3%p
숙박/여행/교통	-1.0%p	-1.1%p	-0.7%p	-0.7%p
외식업	-2.0%p	-1.6%p	-1.1%p	-0.6%p

2019년 1~3월 대비 2020년 1~3월 L.POINT 주 이용 고객 지출비중 증감 기준

L.POINT 주 이용 고객 통계
출처 : 매일경제 20.03.31 기사

1. 2030 젊은 층에서 소비 감소 폭이 컸다.
2. 고객 수와 이용건수의 평균 차이가 별로 없는 것에 비해 매출의 변화는 대부분 업종에서 나타나는 것으로 보아 고객 별 사용금액이 적어진 것으로 예측된다.

5가지 업종 모두 코로나로 인해 피해를 받았다.

PART 5, 회귀분석

회귀분석 시 변수를 선택하는 기준과 방법은 여러 가지가 존재한다.
그 중 선택한 기준과 방법은 다음과 같다.

- 변수 선택 기준

1) AIC : 최소한의 정보 손실을 갖는 모델이 가장 적합한 모델로 선택되는 방법

$$AIC = -2\ln(L) + 2k$$

-2ln(L)은 모형의 적합도, K는 추정된 파라미터 개수이고 **AIC값이 낮을수록 모형의 적합도가 높은 것을 의미**한다.

2) 변수의 p-value : 유의 수준 0.05를 기준으로 변수의 p-value가 유의 수준을 넘지 않는다면 채택, 넘는다면 회귀모델에서 제외한다.

- 변수 선택 방법

1) 전진 선택법 : 모형 적합에 가장 큰 영향을 미치는 독립변수를 순서대로 추가하는 방식

2) 후진 제거법 : 모형 적합에 가장 약하게 영향을 미치는 독립변수를 순서대로 제거하는 방식

3) 단계 선택법 : 전진선택법과 후진제거법을 결합한 방식

AIC 또는 변수의 p-value를 기준으로

변수를 추가 또는 제거를 반복하여

가장 적합한 모델을 찾는 방식

회귀분석

- AIC를 기준 변수 선택 방법 코드

모든 경우의 수를 실행해 단순히 aic가 가장 낮은 모델 선택하는 방법

```
# 변수선택을 통해 형성한 모델의 aic 구하는 함수
# aic가 낮을수록 모델이 좋다고 평가

def processSubset(X, y ,feature_set):
    model = sm.OLS(y,X[list(feature_set)]) #Modeling
    regr = model.fit() # model fitting
    AIC = regr.aic # models' AIC
    return {'model' : regr, 'AIC' : AIC}
```

```
import time
import itertools

#getBest : 가장 낮은 AIC 를 가지는 모델을 선택하고 저장하는 함수

def getBest(X, y, k):
    tic = time.time() # 시작시간
    results =[] # 결과 저장 공간
    for combo in itertools.combinations(X.columns.difference(['const']),k):
                                    # 각 변수 조합을 고려한 경우의수

        combo = (list(combo)+['const'])
        # 상수항을 추가하여 combo를 결성

        results.append(processSubset(X,y,feature_set = combo)) # 모델링된것을 저장

        # 만약 k=2이면 여기서 두가지 변수만 뽑아서 경우의 수를 분석하여
        # 저장 후 그 중 AIC가 가장 낮은 모델을 선택하도록 함

    models = pd.DataFrame(results) # 데이터프레임으로 모델결과 변환
    best_model = models.loc[models['AIC'].argmin()] # argmin은 최소값의 인덱스를 뽑는 함수
    toc = time.time() # 종료시간
    print('Prcessed', models.shape[0], 'models on', k, 'predictors in', (toc-tic), 'seconds')
    return best_model
```

```
### 전진 선택법(step=1)

def forward(X,y,predictors):

    # predictor - 현재 선택되어있는 변수
    # 데이터 변수들이 미리정의된 predictors에 있는지 없는지 확인 및 분류

    remaining_predictors = [p for p in X.columns.difference(['const']) if p not in predictors]
    tic = time.time()
    results = []
    for p in remaining_predictors :
        results.append(processSubset(X=X,y=y,feature_set=predictors+[p]+['const']))

    # 데이터프레임으로 변환
    models = pd.DataFrame(results)

    # AIC가 가장 낮은 것을 선택
    best_model = models.loc[models['AIC'].argmin()]
    toc = time.time()
    print("Processed ",models.shape[0], "models on", len(predictors)+1, "predictors in", (toc-tic))
    print("Selected predictors:",best_model["model"].model.exog_names,"AIC: ",best_model[0])
    return best_model

### 전진선택법 모델

def forward_model(X,y):

    Fmodels = pd.DataFrame(columns=["AIC","model"])
    tic = time.time()

    # 미리 정의된 데이터 변수
    predictors = []

    # 변수 1~10개 : 0-9 -> 1-10
    for i in range(1,len(X.columns.difference(['const']))+1):
        Forward_result = forward(X=X,y=y,predictors=predictors)
        if i > 1 :
            if Forward_result["AIC"] > Fmodel_before:
                break
        Fmodels.loc[i] = Forward_result
        predictors = Fmodels.loc[i]["model"].model.exog_names
        Fmodel_before = Fmodels.loc[i]["AIC"]
        predictors = [k for k in predictors if k != 'const']
    toc = time.time()
    print("Total elapsed time:",(toc-tic), "seconds.")

    return (Fmodels['model'][len(Fmodels['model'])])
```

회귀분석

- AIC를 기준 변수 선택 방법 코드

```
### 후진소거법(step=1)

def backward(X,y,predictors):
    tic = time.time()
    results = []

    # 데이터 변수들이 미리 정의된 predictors 조합 확인

    for combo in itertools.combinations(predictors, len(predictors) - 1):
        results.append(processSubset(X=X,y=y,feature_set=list(combo)+['const']))
    models = pd.DataFrame(results)

    # 가장 낮은 AIC를 가진 모델을 선택
    best_model = models.loc[models['AIC'].argmin()]
    toc = time.time()

    print("Processed ",models.shape[0], "models on", len(predictors) - 1, "predictors in",(toc-tic))
    print("Selected predictors:",best_model['model'].model.exog_names,' AIC:',best_model[0])
    return best_model

def backward_model(X,y) :
    Bmodels = pd.DataFrame(columns=["AIC","model"], index = range(1,len(X.columns)))
    tic = time.time()
    predictors = X.columns.difference(['const'])
    Bmodel_before = processSubset(X,y,predictors)['AIC']
    while (len(predictors) > 1):
        Backward_result = backward(X=X, y= y, predictors=predictors)
        if Backward_result['AIC'] > Bmodel_before :
            break
        Bmodels.loc[len(predictors) -1] = Backward_result
        predictors = Bmodels.loc[len(predictors) - 1]['model'].model.exog_names
        Bmodel_before = Backward_result["AIC"]
        predictors = [k for k in predictors if k != 'const']

    toc = time.time()
    print("Total elapsed time:",(toc-tic),"seconds.")
    return (Bmodels["model"].dropna().iloc[0])
```

```
### 단계적 선택법
def Stepwise_model(X,y):
    Stepmodels = pd.DataFrame(columns = ["AIC","model"])
    tic = time.time()
    predictors = []
    Smodel_before = processSubset(X,y,predictors + ['const'])['AIC']

    # 변수 1~10개 0-9 -> 1-10
    for i in range(1,len(X.columns.difference(['const']))+1) :
        Forward_result = forward(X=X,y=y,predictors = predictors) # constant added
        print('forward')
        Stepmodels.loc[i] = Forward_result
        predictors = Stepmodels.loc[i]['model'].model.exog_names
        predictors = [k for k in predictors if k != 'const']
        Backward_result = backward(X=X,y=y,predictors = predictors)
        if Backward_result["AIC"] < Forward_result["AIC"]:
            Stepmodels.loc[i] = Backward_result
            predictors = Stepmodels.loc[i]["model"].model.exog_names
            Smodel_before = Stepmodels.loc[i]["AIC"]
            predictors = [k for k in predictors if k != "const"]
            print('backward')
        if Stepmodels.loc[i]["AIC"] > Smodel_before:
            break
        else :
            Smodel_before = Stepmodels.loc[i]["AIC"]
    toc = time.time()
    print("Total elapsed time:",(toc-tic),"seconds.")
    return (Stepmodels["model"][len(Stepmodels["model"])])
```

회귀분석

- 카페 (AIC 기준)

1) 전체 경우의 수

In [35]:
```
print("카페:", processSubset(X = train_x_cafe, y=train_y_cafe, feature_set = feature_columns_cafe))
```

```
카페: {'model': <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD2B2AC70>, 'AIC': 733240.8658506114}
```

In [17]:
```
# 카페 변수 선택에 따른 학습시간과 저장
models_cafe = pd.DataFrame(columns=['AIC', 'model'])
tic = time.time()
for i in range(1,6):
    models_cafe.loc[i] = getBest(X=train_x_cafe, y=train_y_cafe, k=i)
toc = time.time()
print('카페 Total elapsed time:', (toc-tic), 'seconds')
```

```
Prcessed 5 models on 1 predictors in 0.019896268844604492 seconds
Prcessed 10 models on 2 predictors in 0.05481457710266113 seconds
Prcessed 10 models on 3 predictors in 0.06179404258728027 seconds
Prcessed 5 models on 4 predictors in 0.026912212371826172 seconds
Prcessed 1 models on 5 predictors in 0.006976127624511719 seconds
카페 Total elapsed time: 0.19730210304260254 seconds
```

In [18]:
```
# 선택된 변수의 개수(1,2,3,4,5) 별 가장 낮음 AIC를 보유한 모델들이 들어있는 data frame
models_cafe
```

Out[18]:

	AIC	model
1	733959.481113	<statsmodels.regression.linear_model.Regressio...
2	733492.362409	<statsmodels.regression.linear_model.Regressio...
3	733241.197980	<statsmodels.regression.linear_model.Regressio...
4	733238.909347	<statsmodels.regression.linear_model.Regressio...
5	733240.865851	<statsmodels.regression.linear_model.Regressio...

In [19]:
```
# 가장 AIC가 낮은 4번째 모델의 OLS 결과 출력
models_cafe.loc[4, 'model'].summary()
```

Out[19]:

OLS Regression Results

Dep. Variable:	금액	**R-squared:**	0.980
Model:	OLS	**Adj. R-squared:**	0.980
Method:	Least Squares	**F-statistic:**	3.121e+05
Date:	Sat, 05 Dec 2020	**Prob (F-statistic):**	0.00
Time:	15:21:46	**Log-Likelihood:**	-3.6661e+05
No. Observations:	25249	**AIC:**	7.332e+05
Df Residuals:	25244	**BIC:**	7.333e+05
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
건수	1055.1582	66.001	15.987	0.000	925.792	1184.524
고객수	1.035e+04	106.308	97.403	0.000	1.01e+04	1.06e+04
생애주기	1.401e+04	6766.933	2.071	0.038	749.278	2.73e+04
연령대	3760.1956	654.764	5.743	0.000	2476.819	5043.572
const	-1.558e+05	9475.457	-16.441	0.000	-1.74e+05	-1.37e+05

Omnibus:	44922.648	**Durbin-Watson:**	2.009
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	366598044.731
Skew:	12.031	**Prob(JB):**	0.00
Kurtosis:	592.817	**Cond. No.**	1.85e+03

회귀분석

- 카페 (AIC 기준)

2) 전진 선택법

In [112]:
```
Forward_best_model = forward_model(X=train_x_cafe, y=train_y_cafe)
```

```
Processed  5 models on 1 predictors in 0.01993250846862793
Selected predictors: ['고객수', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3714D30>
Processed  4 models on 2 predictors in 0.018939733505249023
Selected predictors: ['고객수', '연령대', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3719940>
Processed  3 models on 3 predictors in 0.015945911407470703
Selected predictors: ['고객수', '연령대', '건수', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3714D00>
Processed  2 models on 4 predictors in 0.01295614242553711
Selected predictors: ['고객수', '연령대', '건수', '생애주기', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3714DC0>
Processed  1 models on 5 predictors in 0.007973909378051758
Selected predictors: ['고객수', '연령대', '건수', '생애주기', '성별', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3719460>
Total elapsed time: 0.09667515754699707 seconds.
```

In [113]:
```
# 전진 선택법 모델 최종 선택된 변수와 AIC
print(Forward_best_model.model.exog_names)
print(Forward_best_model.aic)
```

```
['고객수', '연령대', '건수', '생애주기', 'const']
733238.9093465175
```

3) 후진 소거법

In [114]:
```
Backward_best_model = backward_model(X=train_x_cafe, y= train_y_cafe)
```

```
Processed  5 models on 4 predictors in 0.031409502029418945
Selected predictors: ['건수', '고객수', '생애주기', '연령대', 'const']  AIC: <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD367F850>
Processed  4 models on 3 predictors in 0.019930124282836914
Selected predictors: ['건수', '고객수', '연령대', 'const']  AIC: <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3714BE0>
Total elapsed time: 0.0668189525604248 seconds.
```

In [115]:
```
# 후진 소거법 모델 최종 선택된 변수와 AIC
print(Backward_best_model.model.exog_names)
print(Backward_best_model.aic)
```

```
['건수', '고객수', '생애주기', '연령대', 'const']
733238.9093465174
```

회귀분석

- 카페 (AIC 기준)

단계적 선택법

```
In [116]: Stepwise_best_model = Stepwise_model(X=train_x_cafe, y=train_y_cafe)
```

```
Processed  5 models on 1 predictors in 0.017967939376831055
Selected predictors: ['고객수', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3714A
00>
forward
Processed  1 models on 0 predictors in 0.003985166549682617
Selected predictors: ['const']  AIC: <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD36D4D30>
Processed  4 models on 2 predictors in 0.018935680389404297
Selected predictors: ['고객수', '연령대', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000
026DD3719A00>
forward
Processed  2 models on 1 predictors in 0.007973909378051758
Selected predictors: ['고객수', 'const']  AIC: <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000026DD3719C
70>
Processed  3 models on 3 predictors in 0.014950990676879883
Selected predictors: ['고객수', '연령대', '건수', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrapper object a
t 0x0000026DD5727100>
forward
Processed  3 models on 2 predictors in 0.014950990676879883
Selected predictors: ['고객수', '연령대', 'const']  AIC: <statsmodels.regression.linear_model.RegressionResultsWrapper object at 0x0000
026DD36BC8E0>
Processed  2 models on 4 predictors in 0.013953208923339844
Selected predictors: ['고객수', '연령대', '건수', '생애주기', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionResultsWrap
per object at 0x0000026DD36E3BE0>
forward
Processed  4 models on 3 predictors in 0.03388619422912598
Selected predictors: ['고객수', '연령대', '건수', 'const']  AIC: <statsmodels.regression.linear_model.RegressionResultsWrapper object a
t 0x0000026DD54D2490>
Processed  1 models on 5 predictors in 0.010982990264892578
Selected predictors: ['고객수', '연령대', '건수', '생애주기', '성별', 'const'] AIC:  <statsmodels.regression.linear_model.RegressionRes
ultsWrapper object at 0x0000026DD3719BB0>
forward
Processed  5 models on 4 predictors in 0.027907371520996094
Selected predictors: ['고객수', '연령대', '건수', '생애주기', 'const']  AIC: <statsmodels.regression.linear_model.RegressionResultsWrap
per object at 0x0000026DD3714AF0>
backward
Total elapsed time: 0.20434309052062988 seconds.
```

```
In [117]: # 단계적 선택법 모델 최종 선택된 변수와 AIC
          print(Stepwise_best_model.model.exog_names)
          print(Stepwise_best_model.aic)
```

```
['고객수', '연령대', '건수', '생애주기', 'const']
733238.9093465175
```

AIC 기준으로 4가지 방법 모두
고객 수, 연령대, 건 수, 생애주기를
변수로 선택함을 알 수 있다.

회귀분석

- 카페 (변수 P-value 기준)

```
df1 = df1[["년월","업종명","연령대","성별","생애주기","고객수","건수","금액"]]
```

```
# 전진 선택법
variables = df1.columns[2:-1].tolist() # 설명변수 설정

y =df1["금액"] # 반응변수 설정
selected_variables=[] # 선택된 변수
sl_enter = 0.05

sv_per_step = [] # 단계별 선택된 변수
adjusted_r_squared = [] # 단계별 수정된 결정계수
steps = []
step = 0
while len(variables) > 0:
    remainder = list(set(variables) - set(selected_variables))
    pval = pd.Series(index = remainder)
    for col in remainder:
        X = df1[selected_variables+[col]]
        X = sm.add_constant(X)
        model = sm.OLS(y,X).fit()
        pval[col] = model.pvalues[col]

    min_pval = pval.min()
    if min_pval < sl_enter:
        selected_variables.append(pval.idxmin())

        step += 1
        steps.append(step)
        adj_r_squared = sm.OLS(y, sm.add_constant(df1[selected_variables]))
        adjusted_r_squared.append(adj_r_squared)
        sv_per_step.append(selected_variables.copy())
    else:
        break
```

```
<ipython-input-23-9dc2c6234f69>:14: DeprecationWarning: The default dtype for empty Serie
n a future version. Specify a dtype explicitly to silence this warning.
  pval = pd.Series(index = remainder)
```

```
selected_variables
```

```
['고객수', '연령대', '건수', '생애주기']
```

```
# 후진 제거법
variables = df1.columns[2:-1].tolist() # 설명변수 설정

y =df1["금액"] # 반응변수 설정
selected_variables= variables # 초기에는 모든 변수 선택
sl_remove = 0.05

sv_per_step = [] # 단계별 선택된 변수
adjusted_r_squared = [] # 단계별 수정된 결정계수
steps = []
step = 0
while len(variables) > 0:
    X = sm.add_constant(df1[selected_variables])
    p_vals = sm.OLS(y,X).fit().pvalues[1:]
    max_pval = p_vals.max()
    if max_pval >= sl_remove:
        remove_variable = p_vals.idxmax()
        selected_variables.remove(remove_variable)

        step += 1
        steps.append(step)
        adj_r_squared = sm.OLS(y, sm.add_constant(df1[selected_variables]))
        adjusted_r_squared.append(adj_r_squared)
        sv_per_step.append(selected_variables.copy())
    else:
        break
```

```
selected_variables
```

```
['연령대', '생애주기', '고객수', '건수']
```

회귀분석

- 카페 (변수 P-value 기준)

```
# 단계적 선택법
variables = df1.columns[2:-1].tolist() # 설명변수 설정

y =df1["금액"] # 반응변수 설정
selected_variables=[] # 선택된 변수
sl_enter = 0.05
sl_remove = 0.05

sv_per_step = [] # 단계별 선택된 변수
adjusted_r_squared = [] # 단계별 수정된 결정계수
steps = []
step = 0
while len(variables) > 0:
    remainder = list(set(variables) - set(selected_variables))
    pval = pd.Series(index = remainder)
    for col in remainder:
        X = df1[selected_variables+[col]]
        X = sm.add_constant(X)
        model = sm.OLS(y,X).fit()
        pval[col] = model.pvalues[col]

    min_pval = pval.min()
    if min_pval < sl_enter:
        selected_variables.append(pval.idxmin())
        while len(selected_variables) > 0:
            selected_X = df1[selected_variables]
            selected_X = sm.add_constant(selected_X)
            selected_pval = sm.OLS(y,selected_X).fit().pvalues[1:]
            max_pval = selected_pval.max()
            if max_pval >= sl_remove:
                remove_variable = selected_pval.idxmax()
                selected_variabels.remove(remove_variable)
            else:
                break

        step += 1
        steps.append(step)
        adj_r_squared = sm.OLS(y, sm.add_constant(df1[selected_variables]))
        adjusted_r_squared.append(adj_r_squared)
        sv_per_step.append(selected_variables.copy())
    else:
        break
```

```
model1 = smf.ols(formula = "금액 ~ 고객수+연령대+건수+생애주기", data = df1)
result1 = model1.fit()
result1.summary()
```

OLS Regression Results

Dep. Variable:	금액	**R-squared:**	0.982
Model:	OLS	**Adj. R-squared:**	0.982
Method:	Least Squares	**F-statistic:**	4.922e+05
Date:	Fri, 04 Dec 2020	**Prob (F-statistic):**	0.00
Time:	12:29:27	**Log-Likelihood:**	-5.2200e+05
No. Observations:	36070	**AIC:**	1.044e+06
Df Residuals:	36065	**BIC:**	1.044e+06
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.517e+05	7538.664	-20.129	0.000	-1.67e+05	-1.37e+05
고객수	1.074e+04	83.836	128.136	0.000	1.06e+04	1.09e+04
연령대	3605.9034	521.262	6.918	0.000	2584.214	4627.593
건수	777.5586	52.116	14.920	0.000	675.409	879.708
생애주기	1.494e+04	5395.953	2.769	0.006	4367.728	2.55e+04

Omnibus:	60759.658	**Durbin-Watson:**	1.588
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	472944375.127
Skew:	10.632	**Prob(JB):**	0.00
Kurtosis:	563.564	**Cond. No.**	1.86e+03

회귀분석

- 카페 AIC 기준, P-value 기준으로 선택한 모델이 동일

OLS Regression Results

Dep. Variable:	금액	**R-squared:**	0.982
Model:	OLS	**Adj. R-squared:**	0.982
Method:	Least Squares	**F-statistic:**	4.922e+05
Date:	Fri, 04 Dec 2020	**Prob (F-statistic):**	0.00
Time:	12:29:27	**Log-Likelihood:**	-5.2200e+05
No. Observations:	36070	**AIC:**	1.044e+06
Df Residuals:	36065	**BIC:**	1.044e+06
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.517e+05	7538.664	-20.129	0.000	-1.67e+05	-1.37e+05
고객수	1.074e+04	83.836	128.136	0.000	1.06e+04	1.09e+04
연령대	3605.9034	521.262	6.918	0.000	2584.214	4627.593
건수	777.5586	52.116	14.920	0.000	675.409	879.708
생애주기	1.494e+04	5395.953	2.769	0.006	4367.728	2.55e+04

Omnibus:	60759.658	**Durbin-Watson:**	1.588
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	472944375.127
Skew:	10.632	**Prob(JB):**	0.00
Kurtosis:	563.564	**Cond. No.**	1.86e+03

카페 업종에서의 회귀식은
Y= -151700+3606*연령대+14940*생애주기+10740*고객수 +777*건수 로 도출된다.

회귀분석

- **일반 음식점업** AIC 기준, P-value 기준으로 선택한 모델이 동일

```
model3 = smf.ols(formula = "금액 ~ 연령대+성별+생애주기+고객수+건수", data = df3)
result3 = model3.fit()
result3.summary()
```

OLS Regression Results

Dep. Variable:	금액	**R-squared:**	0.977
Model:	OLS	**Adj. R-squared:**	0.977
Method:	Least Squares	**F-statistic:**	1.412e+06
Date:	Fri, 04 Dec 2020	**Prob (F-statistic):**	0.00
Time:	12:39:56	**Log-Likelihood:**	-2.7409e+06
No. Observations:	165688	**AIC:**	5.482e+06
Df Residuals:	165682	**BIC:**	5.482e+06
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-4.843e+05	4.04e+04	-11.978	0.000	-5.64e+05	-4.05e+05
연령대	1.276e+04	1912.351	6.673	0.000	9012.480	1.65e+04
성별	-3.357e+05	1.89e+04	-17.733	0.000	-3.73e+05	-2.99e+05
생애주기	1.014e+05	1.97e+04	5.160	0.000	6.29e+04	1.4e+05
고객수	-8391.8081	221.430	-37.898	0.000	-8825.806	-7957.811
건수	3.844e+04	146.515	262.340	0.000	3.81e+04	3.87e+04

Omnibus:	141498.061	**Durbin-Watson:**	1.622
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	3236662438.967
Skew:	-2.296	**Prob(JB):**	0.00
Kurtosis:	687.698	**Cond. No.**	4.23e+03

일반 음식점 업종에서의 회귀식은

Y= -484300+12760*연령대-335700*성별+101400*생애주기

-8391*고객 수+38440*건수 로 도출된다.

- **스포츠업** AIC 기준, P-value 기준으로 선택한 모델이 동일

```
model2 = smf.ols(formula = "금액 ~ 연령대+생애주기+성별+고객수+건수", data = df2)
result2 = model2.fit()
result2.summary()
```

OLS Regression Results

Dep. Variable:	금액	**R-squared:**	0.937
Model:	OLS	**Adj. R-squared:**	0.937
Method:	Least Squares	**F-statistic:**	3.342e+04
Date:	Fri, 04 Dec 2020	**Prob (F-statistic):**	0.00
Time:	12:36:31	**Log-Likelihood:**	-1.8206e+05
No. Observations:	11301	**AIC:**	3.641e+05
Df Residuals:	11295	**BIC:**	3.642e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.773e+06	9.55e+04	-18.562	0.000	-1.96e+06	-1.59e+06
연령대	1.2e+04	4420.395	2.715	0.007	3336.932	2.07e+04
생애주기	3.035e+05	4.75e+04	6.391	0.000	2.1e+05	3.97e+05
성별	1.522e+05	4.6e+04	3.308	0.001	6.2e+04	2.42e+05
고객수	6.759e+04	2558.313	26.420	0.000	6.26e+04	7.26e+04
건수	4.801e+04	2048.189	23.441	0.000	4.4e+04	5.2e+04

Omnibus:	4540.931	**Durbin-Watson:**	1.443
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	646298.645
Skew:	-0.880	**Prob(JB):**	0.00
Kurtosis:	40.006	**Cond. No.**	575.

스포츠 업종에서의 회귀식은

Y= -1773000+12000*연령대+152200*성별+303500*생애주

+67590*고객 수+48010*건수 로 도출된다.

회귀분석

- **관광업** AIC 기준, P-value 기준으로 선택한 모델이 동일

```
# 세가지 변수를 이용한 회귀 모델 구현
model5 = smf.ols(formula = "금액 ~ 성별+고객수+건수", data = df5)
result5 = model5.fit()
result5.summary()
```

OLS Regression Results

Dep. Variable:	금액	**R-squared:**	0.328
Model:	OLS	**Adj. R-squared:**	0.328
Method:	Least Squares	**F-statistic:**	856.0
Date:	Fri, 04 Dec 2020	**Prob (F-statistic):**	0.00
Time:	12:46:13	**Log-Likelihood:**	-83873.
No. Observations:	5264	**AIC:**	1.678e+05
Df Residuals:	5260	**BIC:**	1.678e+05
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.251e+04	8.68e+04	-0.259	0.795	-1.93e+05	1.48e+05
성별	-1.526e+05	5.66e+04	-2.696	0.007	-2.64e+05	-4.16e+04
고객수	2.622e+05	9355.977	28.023	0.000	2.44e+05	2.81e+05
건수	-1.005e+05	5994.755	-16.764	0.000	-1.12e+05	-8.87e+04

Omnibus:	7691.317	**Durbin-Watson:**	1.312
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4253981.096
Skew:	8.623	**Prob(JB):**	0.00
Kurtosis:	141.194	**Cond. No.**	99.2

관광업에서의 회귀식은
Y= -22510-152600*성별 +262200*고객 수-100500*건수
로 도출된다.

- **전시 및 행사 대행업** AIC 기준, P-value 기준으로 선택한 모델이

```
model4 = smf.ols(formula = "금액 ~ 연령대+성별+고객수+건수", data = df4)
result4 = model4.fit()
result4.summary()
```

OLS Regression Results

Dep. Variable:	금액	**R-squared:**	0.810
Model:	OLS	**Adj. R-squared:**	0.810
Method:	Least Squares	**F-statistic:**	6630.
Date:	Fri, 04 Dec 2020	**Prob (F-statistic):**	0.00
Time:	12:44:36	**Log-Likelihood:**	-87582.
No. Observations:	6217	**AIC:**	1.752e+05
Df Residuals:	6212	**BIC:**	1.752e+05
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-4.017e+05	1.9e+04	-21.128	0.000	-4.39e+05	-3.64e+05
연령대	4646.8870	338.132	13.743	0.000	3984.032	5309.742
성별	5.291e+04	8179.572	6.468	0.000	3.69e+04	6.89e+04
고객수	6.094e+04	1540.066	39.573	0.000	5.79e+04	6.4e+04
건수	-1.171e+04	1328.259	-8.813	0.000	-1.43e+04	-9102.573

Omnibus:	7268.299	**Durbin-Watson:**	1.177
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	3078403.270
Skew:	5.602	**Prob(JB):**	0.00
Kurtosis:	111.436	**Cond. No.**	212.

전시 및 행사 대행업에서의 회귀식은
Y= -401700+4647*연령대+52910*성별+60940*고객 수 -11710*건수 로 도출된다.

회귀분석

- 카페

  Y= -151700+3606*연령대+14940*생애주기+10740*고객 수 +777*건수

- 관광업

  Y= -22510-152600*성별 +262200*고객 수-100500*건수

- 스포츠업

  Y= -1773000+12000*연령대+152200*성별+303500*생애주기 +67590*고객 수+48010*건수

- 일반 음식점업

  Y= -484300+12760*연령대-335700*성별+101400*생애주기 -8391*고객 수+38440*건수

- 전시 및 행사 대행업

  Y= -401700+4647*연령대+52910*성별+60940*고객 수 -11710*건수

결과 해석

성별, 생애주기, 연령대는 범주형 변수이기 때문에 단지 금액의 값이 변하는 것에만 영향을 주고, 연속형 변수인 **건수, 고객수**는 해당 값이 달라짐에 따라 매출변화에 많은 영향을 끼칠 것으로 예상된다.

(계수가 음수이면 매출에 부정적인 영향을 주고, 양수이면 매출변화에 긍정적인 영향을 준다)

감사합니다