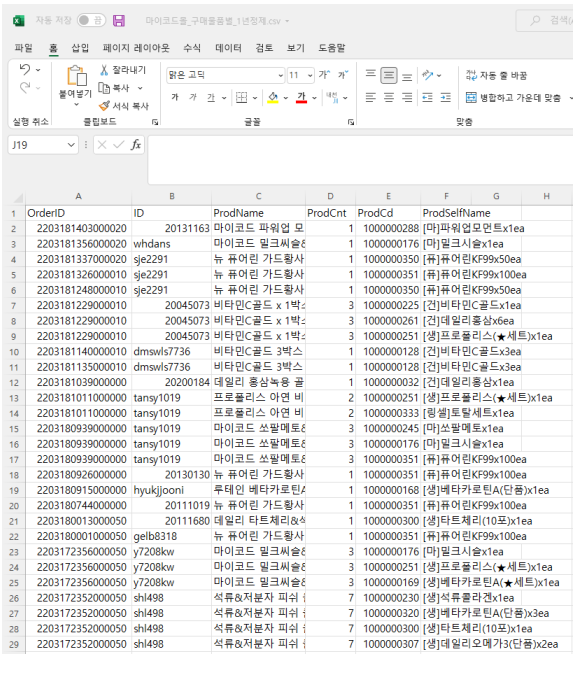
연관분석_아이코드몰 데이터

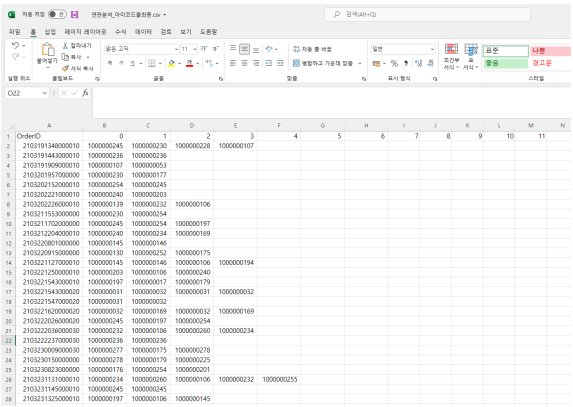
- **원본데이터**

  https://s3-us-west-2.amazonaws.com/secure.notion-static.com/073198bc-e7ba-42c4-ae5f-b4c88c69312f/가공전.csv

  - 아래 데이터의 OrderID에는 중복된 아이디가 존재. 아이디의 unique한 값으로 PeodCd를 가로 형태로 나열.
  - 연관분석을 하기 위해 한 고객(ID)가 구매한 제품의 패턴을 파악.



	A	B	C	D	E	F	G	H
1	OrderID	ID	ProdName	ProdCnt	ProdCd	ProdSelfName		
2	2203181403000020	20131163	마이코드 파워업 모	1	1000000288	[마]파워업모먼트x1ea		
3	2203181356000020	whdans	마이코드 밀크씨슬&	1	1000000176	[마]밀크시슬x1ea		
4	2203181337000020	sje2291	뉴 퓨어린 가드황사	1	1000000350	[퓨]퓨어린KF99x50ea		
5	2203181326000010	sje2291	뉴 퓨어린 가드황사	1	1000000351	[퓨]퓨어린KF99x100ea		
6	2203181248000010	sje2291	뉴 퓨어린 가드황사	1	1000000350	[퓨]퓨어린KF99x50ea		
7	2203181229000010	20045073	비타민C골드 x 1박스	3	1000000225	[건]비타민C골드x1ea		
8	2203181229000010	20045073	비타민C골드 x 1박스	3	1000000261	[건]데일리홍삼x6ea		
9	2203181229000010	20045073	비타민C골드 x 1박스	3	1000000251	[생]프로폴리스(★세트)x1ea		
10	2203181140000010	dmswls7736	비타민C골드 3박스	1	1000000128	[건]비타민C골드x3ea		
11	2203181135000010	dmswls7736	비타민C골드 3박스	1	1000000128	[건]비타민C골드x3ea		
12	2203181039000000	20200184	데일리 홍삼녹용 골	1	1000000032	[건]데일리홍삼x1ea		
13	2203181011000000	tansy1019	프로폴리스 아연 비	2	1000000251	[생]프로폴리스(★세트)x1ea		
14	2203181011000000	tansy1019	프로폴리스 아연 비	2	1000000333	[링셀]토탈세트x1ea		
15	2203180939000000	tansy1019	마이코드 쏘팔메토&	3	1000000245	[마]쏘팔메토x1ea		
16	2203180939000000	tansy1019	마이코드 쏘팔메토&	3	1000000176	[마]밀크시슬x1ea		
17	2203180939000000	tansy1019	마이코드 쏘팔메토&	3	1000000351	[퓨]퓨어린KF99x100ea		
18	2203180926000000	20130130	뉴 퓨어린 가드황사	1	1000000351	[퓨]퓨어린KF99x100ea		
19	2203180915000000	hyukjjooni	루테인 베타카로틴A	1	1000000168	[생]베타카로틴A(단품)x1ea		
20	2203180744000000	20111019	뉴 퓨어린 가드황사	1	1000000351	[퓨]퓨어린KF99x100ea		
21	2203180013000050	20111680	데일리 타트체리&쏘	1	1000000300	[생]타트체리(10포)x1ea		
22	2203180001000050	gelb8318	뉴 퓨어린 가드황사	1	1000000351	[퓨]퓨어린KF99x100ea		
23	2203172356000050	y7208kw	마이코드 밀크씨슬&	3	1000000176	[마]밀크시슬x1ea		
24	2203172356000050	y7208kw	마이코드 밀크씨슬&	3	1000000251	[생]프로폴리스(★세트)x1ea		
25	2203172356000050	y7208kw	마이코드 밀크씨슬&	3	1000000169	[생]베타카로틴A(★세트)x1ea		
26	2203172352000050	shl498	석류&저분자 피쉬	7	1000000230	[생]석류콜라겐x1ea		
27	2203172352000050	shl498	석류&저분자 피쉬	7	1000000320	[생]베타카로틴A(단품)x3ea		
28	2203172352000050	shl498	석류&저분자 피쉬	7	1000000300	[생]타트체리(10포)x1ea		
29	2203172352000050	shl498	석류&저분자 피쉬	7	1000000307	[생]데일리오메가3(단품)x2ea		



1. 데이터 불러오기

```
df=pd.read_csv('C:/Users/areum/Desktop/POC시연자료/가공전.csv')
```

```
df
```

	OrderID	ProdCd
0	2203181403000020	1000000288
1	2203181356000020	1000000176
2	2203181337000020	1000000350
3	2203181326000010	1000000351
4	2203181248000010	1000000350
...	...	...
18343	2103191212000000	1000000106
18344	2103191059000000	1000000254
18345	2103191055000000	1000000254
18346	2103191004000000	1000000259
18347	2103190822000000	1000000131

2. groupby

- groupby사용해서 orderid를 그룹으로 묶고 prodcd를 apply함수를 이용해 리스트 형태로 만듦 > reset_index를 통해 만들어진 리스트의 dataframe이름을new로 정해줌.

```
df3=df.groupby('OrderID')['ProdCd'].apply(list).reset_index(name='new')
```

```
df3
```

	OrderID	new
0	2103190822000000	[1000000131]
1	2103191004000000	[1000000259]
2	2103191055000000	[1000000254]
3	2103191059000000	[1000000254]
4	2103191212000000	[1000000106]
...	...	...
10121	2203181248000010	[1000000350]
10122	2203181326000010	[1000000351]
10123	2203181337000020	[1000000350]
10124	2203181356000020	[1000000176]
10125	2203181403000020	[1000000288]

10126 rows × 2 columns

여기까지 dataframe한개 > df3

1. 원본데이터에서 ordeid 개수 세기

- 원본데이터에서 orderid를 그룹으로 묶고 인덱스는 reset하기 위해 False사용. count()를 이용하여 id값이 몇번 사용되었는지 세줌.

```
df=df.groupby(by=['OrderID'], as_index=False).count()
```

```
df
```

	OrderID	ProdCd
0	2103190822000000	1
1	2103191004000000	1
2	2103191055000000	1
3	2103191059000000	1
4	2103191212000000	1
...	...	...
10121	2203181248000010	1
10122	2203181326000010	1
10123	2203181337000020	1
10124	2203181356000020	1
10125	2203181403000020	1

10126 rows × 2 columns

2. 중복 2개 이상 추출 작업

- a를 실행하면 false와 true형식으로 나옴 이걸 df에 넣어줘야 true인 값만 빼줌.

```
a=df.ProdCd>=2
```

```
df=df[a]
```

```
df
```

	OrderID	ProdCd
6	2103191348000010	4
11	2103191443000010	2
14	2103191909000010	2
19	2103201957000000	2
22	2103202152000010	2
...	...	...
10108	2203172352000050	7
10109	2203172356000050	3
10115	2203180939000000	3
10116	2203181011000000	2
10120	2203181229000010	3

4345 rows × 2 columns

여기까지 dataframe한개 > df

---

1. merge사용해서 dataframe 2개 합치기

- on은 통해 공통적인 것을 묶음

```
merge_outer = pd.merge(df,df3,on='OrderID')
```

```
merge_outer
```

	OrderID	ProdCd	new
0	2103191348000010	4	[1000000245, 1000000230, 1000000228, 1000000107]
1	2103191443000010	2	[1000000236, 1000000236]
2	2103191909000010	2	[1000000107, 1000000053]
3	2103201957000000	2	[1000000230, 1000000177]
4	2103202152000010	2	[1000000254, 1000000245]
...	...	...	...
4340	2203172352000050	7	[1000000230, 1000000320, 1000000300, 100000030...
4341	2203172356000050	3	[1000000176, 1000000251, 1000000169]
4342	2203180939000000	3	[1000000245, 1000000176, 1000000351]
4343	2203181011000000	2	[1000000251, 1000000333]
4344	2203181229000010	3	[1000000225, 1000000261, 1000000251]

4345 rows × 3 columns

2. new안에 있는 리스트를 한개씩 분리하는 작업

- new열을 apply함수를 사용해서

```
new=merge_outer['new'].apply(lambda x: pd.Series(x))
```

```
new
```

	0	1	2	3	4	5
0	1.000000e+09	1.000000e+09	1.000000e+09	1.000000e+09	NaN	NaN
1	1.000000e+09	1.000000e+09	NaN	NaN	NaN	NaN
2	1.000000e+09	1.000000e+09	NaN	NaN	NaN	NaN
3	1.000000e+09	1.000000e+09	NaN	NaN	NaN	NaN
4	1.000000e+09	1.000000e+09	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...
4340	1.000000e+09	1.000000e+09	1.000000e+09	1.000000e+09	1.000000e+09	1.000000e+09
4341	1.000000e+09	1.000000e+09	1.000000e+09	NaN	NaN	NaN
4342	1.000000e+09	1.000000e+09	1.000000e+09	NaN	NaN	NaN
4343	1.000000e+09	1.000000e+09	NaN	NaN	NaN	NaN
4344	1.000000e+09	1.000000e+09	1.000000e+09	NaN	NaN	NaN

3. 마지막 new dataframe과 merge_outer를 인덱스대로 합쳐주면 끝 !

	A	B	C	D	E	F	G	H	I	J
1	OrderID	0	1	2	3	4	5	6	7	8
2	2103191348000010	1000000245	1000000230	1000000228	1000000107					
3	2103191443000010	1000000236	1000000236							
4	2103191909000010	1000000107	1000000053							
5	2103201957000000	1000000230	1000000177							
6	2103202152000010	1000000254	1000000245							
7	2103202221000010	1000000240	1000000203							
8	2103202226000010	1000000139	1000000232	1000000106						
9	2103211553000000	1000000230	1000000254							
10	2103211702000000	1000000245	1000000254	1000000197						
11	2103212204000010	1000000240	1000000234	1000000169						
12	2103220801000000	1000000145	1000000146							
13	2103220915000000	1000000130	1000000252	1000000175						
14	2103221127000010	1000000145	1000000146	1000000106	1000000194					
15	2103221250000010	1000000203	1000000106	1000000240						
16	2103221543000010	1000000197	1000000017	1000000179						
17	2103221543000020	1000000031	1000000032	1000000031	1000000032					
18	2103221547000020	1000000031	1000000032							
19	2103221620000020	1000000032	1000000169	1000000032	1000000169					
20	2103222026000020	1000000245	1000000197	1000000254						
21	2103222036000030	1000000232	1000000106	1000000260	1000000234					
22	2103222237000030	1000000236	1000000236							
23	2103230009000030	1000000277	1000000175	1000000278						
24	2103230150000000	1000000278	1000000179	1000000225						
25	2103230823000000	1000000176	1000000254	1000000201						
26	2103231131000010	1000000234	1000000260	1000000106	1000000232	1000000255				
27	2103231145000010	1000000245	1000000245							
28	2103231325000010	1000000197	1000000106	1000000145						
29	2103231824000010	1000000168	1000000106							
30	2103232140000010	1000000032	1000000227	1000000143	1000000230	1000000236	1000000236	1000000251		
31	2103240840000000	1000000244	1000000140							

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b83d7308-0f55-4e02-96b4-b6f9d0fe11a4/연관분석_아이코드몰최종.csv