

- 제공된 m_question01_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하시오. (먼저 android_devices.csv의 'Retail Branding' 컬럼명을 'manufacturer'로 변경하여 출력하고, user_usage와 user_device 데이터프레임을 공통컬럼인 'use_id' 사용하여 merge 하여 출력한다.)

1. 컬럼명 변경후의 devices 정보

	manufacturer	Marketing Name	Device	Model
0	NaN	NaN	AD681H Smartfren Andromax	AD681H
1	NaN	NaN	FJL21	FJL21
2	NaN	NaN	T31	Panasonic T31
3	NaN	NaN	hws7721g	MediaPad 7 Youth 2
4	3Q	0C1020A	0C1020A	0C1020A

2. result 데이터프레임:

	outgoing_mins_per_month	outgoing_sms_per_month	...	platform	device
0	21.97	4.82	...	android	GT-I9505
1	1710.08	136.88	...	android	SM-G930F
2	1710.08	136.88	...	android	SM-G930F
3	94.46	35.17	...	android	D2303
4	71.59	79.26	...	android	SM-G361F

[5 rows x 6 columns]

3. user_usage 데이터프레임의 dimensions: (240, 4)

4. result 데이터프레임의 dimensions: (159, 6)

- 제공된 m_question02_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하시오. (앞의 m_question01_제공파일.py 의 merge작업은 user_usage 데이터프레임에서 누락된 데이터가 발생된다. 따라서 outer 작업으로 merge 하여 모든 user_usage 데이터를 출력하도록 한다.)

1. result 데이터프레임:

	outgoing_mins_per_month	outgoing_sms_per_month	...	platform	device
0	21.97	4.82	...	android	GT-I9505
1	1710.08	136.88	...	android	SM-G930F
2	1710.08	136.88	...	android	SM-G930F
3	94.46	35.17	...	android	D2303
4	71.59	79.26	...	android	SM-G361F

[5 rows x 6 columns]

2. user_usage 데이터프레임의 dimensions: (240, 4)

3. result 데이터프레임의 dimensions: (240, 6)

4. result 데이터프레임의 'result' 컬럼에서 NaN값의 총갯수:81

3. 제공된 m_question03_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하시오. (앞의 m_question01_제공파일.py 의 merge작업은 user_device 데이터프레임에서 누락된 데이터가 발생된다. 따라서 outer 작업으로 merge 하여 모든 **user_device** 데이터를 출력하도록 한다.)

```
1. result 데이터프레임:
   outgoing_mins_per_month  outgoing_sms_per_month  ...  platform  device
0                NaN                NaN  ...    ios  iPhone7,2
1                NaN                NaN  ...  android  Nexus 5
2                NaN                NaN  ...  android  SM-G903F
3                NaN                NaN  ...    ios  iPhone7,2
4                NaN                NaN  ...  android  ONE E1003

[5 rows x 6 columns]
2.user_device 데이터프레임의 dimensions: (272, 6)
3. result 데이터프레임의 dimensions: (272, 6)
4. result 데이터프레임의 'monthly_mb' 컬럼에서 NaN값의 총갯수:113
5. result 데이터프레임의 'platform' 컬럼에서 NaN값의 총갯수:0
```

4. 제공된 m_question04_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하시오. (**user_usage**컬럼과 **user_device**컬럼이 모두 출력되도록 **outer merge**를 실행하고 **null** 값을 포함하는 행의 개수 출력)

```
1. merge한 전체 dimensions: (353, 7)
2. null 값을 가진 행의 갯수: 194
```

5. 제공된 m_question05_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하시오. (**제공된 3개의 데이터프레임** 활용한 **outer merge** 실행)

```
1. result 데이터프레임:
   outgoing_mins_per_month  outgoing_sms_per_month  ...  manufacturer  Model
0                21.97                4.82  ...    Samsung  GT-I9505
1               1710.08               136.88  ...    Samsung  SM-G930F
2               1710.08               136.88  ...    Samsung  SM-G930F
3                94.46                35.17  ...      Sony  D2303
4                71.59                79.26  ...    Samsung  SM-G361F

[5 rows x 8 columns]
2. result 데이터프레임의 dimensions: (292, 8)
```

6. 제공된 m_question06_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하시오. (devices 데이터프레임에서 Model값이 'SM-G930F' 행 검색 및 Device명이 'GT'로 시작하는 행 검색 .)

1. devices에서 Model명이 'SM-G930F' 검색:

	Retail	Branding	Marketing	Name	Device	Model
10381	Samsung	Galaxy	S7	herolte	SM-G930F	

2. devices에서 Device명이 'GT' 로 시작하는 데이터 검색:

	Retail	Branding	Marketing	Name	Device	Model
1095	Bitmore	GTAB700	GTAB700	NID_7010		
1096	Bitmore	GTAB900	GTAB900	S952		
2402	Grundig	GTB1050	GTB1050	GTB 1050		
2403	Grundig	GTB850	GTB850	GTB 850		
2404	Grundig	TC69CA2	GTB801	GTB 801		

2. result 데이터프레임의 dimensions: (164, 4)

7. 제공된 m_question 07_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하시오. (m_question05_제공파일.py 활용하여 'manufacturer' 별 use_id의 개수 (count)와 나머지 컬럼은 평균(mean)값으로 설정)

	use_id	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb
manufacturer				
HTC	44	299.842955	93.059318	5144.077955
Huawei	3	81.526667	9.500000	1561.226667
LGE	2	111.530000	12.760000	1557.330000
Lava	2	60.650000	261.900000	12458.670000
Lenovo	2	215.920000	12.930000	1557.330000
Motorola	16	95.127500	65.666250	3946.500000
OnePlus	6	354.855000	48.330000	6575.410000
Samsung	108	191.010093	92.390463	4017.318889
Sony	16	177.315625	40.176250	3212.000625
Vodafone	1	42.750000	46.830000	5191.120000
ZTE	1	42.750000	46.830000	5191.120000

8. m_question05_제공파일.py 소스파일을 활용하여 다음과 같이 출력 되도록 코드를 작성하십시오. (manufacturer 컬럼값이 Samaung 또는 Sony이고 outgoing_sms_per_month 값이 90보다 큰 조건 만족)

1. result 데이터프레임:

	outgoing_mins_per_month	outgoing_sms_per_month	...	manufacturer	Model
1	1710.08	136.88	...	Samsung	SM-G930F
2	1710.08	136.88	...	Samsung	SM-G930F
10	554.41	150.06	...	Samsung	SM-G935F
13	324.34	92.52	...	Samsung	SM-G900F
27	164.10	192.64	...	Samsung	SM-G900F
28	208.26	91.76	...	Samsung	SM-G900F
30	324.27	91.50	...	Samsung	SM-G900F
32	244.88	105.95	...	Sony	D5803
33	244.88	105.95	...	Sony	D5803
183	42.93	124.33	...	Samsung	SM-G361F
193	92.52	162.39	...	Sony	C6603
202	198.59	90.49	...	Samsung	SM-G900F
203	198.59	90.49	...	Samsung	SM-G900F
204	198.59	90.49	...	Samsung	SM-G900F

[48 rows x 8 columns]

2. result 데이터프레임의 dimensions: (48, 8)