

では、顧客毎に月/曜日別に集計を行い、最大値が4以上の曜日が1ヶ月でもあったユーザーはフラグ1とします。

それではやってみましょう。

まずは、顧客毎に月/曜日別に集計を行います。

```
uselog["weekday"] = uselog["usedate"].dt.weekday
uselog_weekday = uselog.groupby(["customer_id", "年月", "weekday"], as_index=False).count()[["customer_id", "年月", "weekday", "log_id"]]
uselog_weekday.rename(columns={"log_id": "count"}, inplace=True)
uselog_weekday.head()
```

■図3-8：顧客毎の月/曜日別集計結果

ノック26：利用履歴データから定期利用フラグを作成しよう

```
In [85]: uselog["weekday"] = uselog["usedate"].dt.weekday
uselog_weekday = uselog.groupby(["customer_id", "年月", "weekday"], as_index=False).count()[["customer_id", "年月", "weekday", "log_id"]]
uselog_weekday.rename(columns={"log_id": "count"}, inplace=True)
uselog_weekday.head()
```

Out[85]:

	customer_id	年月	weekday	count
0	AS002855	201804	5	4
1	AS002855	201805	2	1
2	AS002855	201805	5	4
3	AS002855	201806	5	5
4	AS002855	201807	1	1

1行目は、曜日の計算を行っています。0から6が付与され、それぞれ月曜から日曜に相当します。2行目では顧客、年月、曜日毎にlog\_idを数えています。表示された結果を見ると、顧客AS002855は、2018年4月に、weekdayが5、つまり土曜日に4回来ています。2018年5月にも土曜日に4回来ているため、毎週土曜日をジムの日として考えているようです。

ここから、顧客毎の各月の最大値を取得し、その最大値が4以上の場合、フラグを立ててみましょう。

```
uselog_weekday = uselog_weekday.groupby("customer_id", as_index=False).max()[["customer_id", "count"]]
uselog_weekday["routine_flg"] = 0
uselog_weekday["routine_flg"] = uselog_weekday["routine_flg"].where(uselog_weekday["count"] < 4, 1)
uselog_weekday.head()
```

■図3-9：フラグ作成結果

```
In [112]: uselog_weekday = uselog_weekday.groupby("customer_id", as_index=False).max()[["customer_id", "count"]]
uselog_weekday["routine_flg"] = 0
uselog_weekday["routine_flg"] = uselog_weekday["routine_flg"].where(uselog_weekday["count"] < 4, 1)
uselog_weekday.head()
```

Out[112]:

	customer_id	count	routine_flg
0	AS002855	5	1
1	AS008805	4	1
2	AS009013	2	0
3	AS009373	5	1
4	AS015233	5	1

1行目で、顧客単位まで集計を行い、その際に最大値を取得しています。つまり、月内かつ特定の曜日で最も利用した回数です。ここが4もしくは5の人は、少なくともどこかの月では、毎週特定の曜日で来たユーザーとなります。2行目、3行目では、4未満の場合は元の値である0をそのまま、4以上の場合は1を代入しています。

実際に、先頭5行をみると、countが2の顧客は0が入っていることがわかります。

これで、フラグの作成が完了しました。

次に、顧客データと結合し、利用履歴も含んだ顧客データを整形しましょう。

## ノック27：顧客データと利用履歴データを結合しよう

ここでは、ノック25、ノック26で作成したuselog\_customer、uselog\_weekdayを、customer\_joinと結合しましょう。

これまで何度も結合はやってきているので、自分なりに考えて挑戦しましょう。

```
customer_join = pd.merge(customer_join, uselog_customer, on="customer_id", how="left")
customer_join = pd.merge(customer_join, uselog_weekday[["customer_id", "routine_flg"]], on="customer_id", how="left")
customer_join.head()
```