

■図3-10：顧客データと利用履歴データの結合結果

ノック27：顧客データと利用履歴データを結合しよう

```
In [21]: 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()
```

```
Out[21]:
```

	customer_id	name	class	gender	start_date	end_date	campaign_id	is_deleted	class_name	price	campaign_name	mean	median	max	min	rou
0	OAB32399	XXXX	C01	F	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	4.833333	5.0	6	2	
1	PL270116	XXXX	C01	M	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	5.083333	5.0	7	3	
2	QA974876	XXXX	C01	M	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	4.583333	5.0	6	3	
3	HD024127	XXXX	C01	F	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	4.833333	4.5	7	2	
4	HD061448	XXXX	C03	F	2015-05-01	NaT	CA1	0	ナイト	6000	通常	3.916667	4.0	6	1	

もうだいぶ慣れてきたのではないのでしょうか。

二回結合を行っていますが、結合に使用するジョインキーはcustomer_idで、結合方法はレフトジョインとなります。表示された結果を見ると、先ほど集計した、median等やroutine_flgが結合されていることがわかります。

2行目の結合の際には、結合するuselog_weekdayの列をジョインキーのcustomer_idと結合したいroutine_flgに絞って結合を行っています。

念の為、欠損値も確認しておきましょう。

```
customer_join.isnull().sum()
```

実行するとend_date以外に欠損値がないことがわかります。これで、結合が問題なくできていることが確認できました。

いよいよ、利用履歴も加味した形で顧客の分析となるのですが、その前に、せっかく利用履歴で時間的な変化のデータを追加することができたので、会員期間という軸をもう1つ追加しておきましょう。

ノック28： 会員期間を計算しよう

それでは、会員期間を計算して列に追加していきます。

会員期間は、単純にはstart_dateとend_dateの差になります。ただし、2019年3月までに退会していないユーザーに関しては、end_dateに欠損値が入っています。そのため、差の計算ができません。そこで、ここでは2019年4

月30日として会員期間を算出しましょう。いろんな計算方法がありますが、2019年3月31日で算出する場合、実際に2019年3月31日で退会した人(2月末までに退会申請をした人)と区別がつかなくなるためです。

それでは、期間を計算していきましょう。ここでは、月単位で集計を行いましょう。

```
from dateutil.relativedelta import relativedelta
customer_join["calc_date"] = customer_join["end_date"]
customer_join["calc_date"] = customer_join["calc_date"].fillna(pd.to_datetime("20190430"))
customer_join["membership_period"] = 0
for i in range(len(customer_join)):
    delta = relativedelta(customer_join["calc_date"].iloc[i], customer_join["start_date"].iloc[i])
    customer_join["membership_period"].iloc[i] = delta.years*12 + delta.months
customer_join.head()
```

■図3-11：会員期間の計算結果

ノック28：会員期間を計算しよう

```
In [24]: from dateutil.relativedelta import relativedelta
customer_join["calc_date"] = customer_join["end_date"]
customer_join["calc_date"] = customer_join["calc_date"].fillna(pd.to_datetime("20190430"))
customer_join["membership_period"] = 0
for i in range(len(customer_join)):
    delta = relativedelta(customer_join["calc_date"].iloc[i], customer_join["start_date"].iloc[i])
    customer_join["membership_period"].iloc[i] = delta.years*12 + delta.months
customer_join.head()
```

```
Out[24]:
```

	customer_id	name	class	gender	start_date	end_date	campaign_id	is_deleted	class_name	price	campaign_name	mean	median	max	min	rou
0	OAB32399	XXXX	C01	F	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	4.833333	5.0	6	2	
1	PL270116	XXXX	C01	M	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	5.083333	5.0	7	3	
2	QA974876	XXXX	C01	M	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	4.583333	5.0	6	3	
3	HD024127	XXXX	C01	F	2015-05-01	NaT	CA1	0	オールタイム	10500	通常	4.833333	4.5	7	2	
4	HD061448	XXXX	C03	F	2015-05-01	NaT	CA1	0	ナイト	6000	通常	3.916667	4.0	6	1	

1行目でrelativedeltaを使用するためにライブラリのインポートをしています。これは、日付の比較に使用します。2行目で、日付計算用の列をend_dateベースに作成し、3行目で、欠損値に2019年4月30日を代入しています。

その後、データフレームを上から順番に計算し、会員期間を月単位で算出しています。