

## ノック22： 顧客データを整形しよう

ここでは、顧客データの整形を行います。ノック21で読み込んだcustomerに、会員区分のclass\_masterとキャンペーン区分のcampaign\_masterを結合します。

顧客データを主にして横に結合するので、**レフトジョイン**となります。

**ジョインキー**は自分で探してみましょう。

また、ジョイン前後でデータ件数が変わらないことを確認しましょう。

```
customer_join = pd.merge(customer, class_master, on="class", how="left")
customer_join = pd.merge(customer_join, campaign_master, on="campaign_id",
how="left")
customer_join.head()
```

■図3-2：データユニオン

ノック22：顧客データを整形しよう

```
In [7]: customer_join = pd.merge(customer, class_master, on="class", how="left")
customer_join = pd.merge(customer_join, campaign_master, on="campaign_id", how="left")
customer_join.head()
```

	customer_id	name	class	gender	start_date	end_date	campaign_id	is_deleted	class_name	price	campaign_name
0	CA832399	XXXX	C01	F	2015-05-01 00:00:00	NaN	CA1	0	オールタイム	10500	通常
1	FL270116	XXXX	C01	M	2015-05-01 00:00:00	NaN	CA1	0	オールタイム	10500	通常
2	CA974876	XXXX	C01	M	2015-05-01 00:00:00	NaN	CA1	0	オールタイム	10500	通常
3	HD024127	XXXX	C01	F	2015-05-01 00:00:00	NaN	CA1	0	オールタイム	10500	通常
4	HD661448	XXXX	C03	F	2015-05-01 00:00:00	NaN	CA1	0	ナイト	6000	通常

```
In [8]: print(len(customer))
print(len(customer_join))
4192
4192
```

1行目で、会員区分のマスタデータであるclass\_masterと、2行目で、キャンペーン区分であるcampaign\_masterとそれぞれ結合しています。実際に先頭5行の出力結果を見ると、class\_name、price、campaign\_name列が追加され、会員区分や金額等が分かるようにデータを整形できました。データ件数も、ジョイン前後で変化がないことが確認できます。

ジョインする際、キーが見つからないなど、上手くジョインができないと、欠損値が自動で入ります。そのため、ジョイン後は欠損値の確認をするようにしましょう。

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

■図3-3：欠損値の確認

```
In [9]: customer_join.isnull().sum()
Out[9]: customer_id    0
name                0
class              0
gender            0
start_date         0
end_date        2842
campaign_id        0
is_deleted         0
class_name         0
price              0
campaign_name      0
dtype: int64
```

end\_date以外は欠損値が0となっており、今回ジョインで追加した、class\_name、price、campaign\_name列にしっかりデータが入っていることが確認できました。また、end\_dateに欠損値が入っていること以外は比較的綺麗なデータであることもわかります。

また、end\_dateに欠損が入っている理由としては、退会していないユーザーは、退会日であるend\_dateを保持していないため、欠損値となっていることが考えられます。

## ノック23： 顧客データの基礎集計をしよう

データ加工が完了したので、この顧客データを集計し、全体像をみていきましょう。

まずは、集計する項目を考えてみます。どの会員区分やキャンペーン区分が多いか、いつ入会/退会が多いのか、男女比率や退会するまでの期間等を集計することができることに気づくと思います。まずは、会員区分、キャンペーン区分、性別、既に退会済みかどうか(is\_deleted列)毎に全体の数を把握してみましょう。

```
customer_join.groupby("class_name").count()["customer_id"]
```