

■図1-12：pivot_tableによる集計

```
In [21]: pd.pivot_table(join_data, index='item_name', columns='payment_month', values=['price', 'quantity'], aggfunc='sum')
```

```
Out[21]:
```

item_name	price						quantity					
	payment_month	201902	201903	201904	201905	201906	201907	201902	201903	201904	201905	201906
PC-A	24150000	26000000	25900000	24850000	26000000	25250000	483	520	518	497	520	505
PC-B	25245000	25500000	23460000	25330000	23970000	28220000	297	300	276	298	282	332
PC-C	19800000	19080000	21960000	20520000	21840000	19440000	165	159	183	171	182	162
PC-D	31140000	25740000	24300000	25920000	28800000	26100000	173	143	135	144	160	145
PC-E	59850000	64050000	64890000	58800000	63420000	71610000	285	305	309	280	302	341

2行目のpivot_tableは、行と列を指定することができます。そのため、今回は行に商品名、列に年月データがくるように、indexとcolumnsで指定しています。valuesでは集計したい数値列(price、quantity)、aggfuncには集計方法(sum)を指定しています。

先ほどの表よりも、月別、商品別の推移が把握しやすいかと思います。

売上の合計値としては、最も高価格のPC-Eですが、やはり数量的には最も安い価格のPC-Aが多いです。先ほどの月別推移では5月の売上が低下し、6月、7月で増加していました。商品別に見ると、5月はPC-B、PC-Dは増加していますが、大きな売上を占めるPC-Eの売上が大きく低下している影響が大きそうです。6月、7月は、PC-Eが大きく伸びてきています。

商品別の月別推移を見ることができました。

最後に、これを簡単なグラフとして表現しましょう。

表形式のデータは、細かい数字が把握できる一方で、伸びているのか、落ちているのかが一目でわかりません。分析のゴールはあくまでも現場で適切に運用され施策を回していくことです。現場の人の中には、数字が苦手な方もいるので、いかにして伝えるかはとても重要です。

⑩ ノック10： 商品別の売上推移を可視化してみよう

今回は、これまで見てきた、月別、商品別の売上推移をグラフにしてみましょう。可視化の流れとしては、まずは集計済みのデータを作成し、そのデータを用いてグラフ描画となります。

まずは、pivot_tableを用いてデータ集計を行います。

■図1-13：グラフ用データ作成

```
ノック10：商品別の売上推移を可視化してみよう

In [22]: graph_data = pd.pivot_table(join_data, index='payment_month', columns='item_name', values='price', aggfunc='sum')
graph_data.head()
Out[22]:
```

item_name	PC-A	PC-B	PC-C	PC-D	PC-E
payment_month					
201902	24150000	25245000	19800000	31140000	59850000
201903	26000000	25500000	19080000	25740000	64050000
201904	25900000	23460000	21960000	24300000	64890000
201905	24850000	25330000	20520000	25920000	58800000
201906	26000000	23970000	21840000	28800000	63420000

先ほどと、index、columnsに指定するものを入れ替えてあります。その結果、payment_monthはデータフレーム型のindexとして、商品名は列として作成できます。グラフを作成する際には、横軸にpayment_monthを、縦軸にgraph_dataの該当商品名を渡すことで描画が可能となります。

それでは、matplotlibを用いて描画してみましょう。

```
import matplotlib.pyplot as plt

%matplotlib inline

plt.plot(list(graph_data.index), graph_data["PC-A"], label='PC-A')
plt.plot(list(graph_data.index), graph_data["PC-B"], label='PC-B')
plt.plot(list(graph_data.index), graph_data["PC-C"], label='PC-C')
plt.plot(list(graph_data.index), graph_data["PC-D"], label='PC-D')
plt.plot(list(graph_data.index), graph_data["PC-E"], label='PC-E')

plt.legend()
```