# Pivot Tables

February 2, 2018

## 1 Pivot Tables

```
In [1]: import pandas as pd
        import numpy as np
```

```
In [2]: data = pd.read_excel("sales-funnel.xlsx")
```

```
In [3]: data.head()
```

```
Out[3]:    Account                           Name            Rep       Manager  \
        0   714466                Trantow-Barrows   Craig Booker  Debra Henley
        1   714466                Trantow-Barrows   Craig Booker  Debra Henley
        2   714466                Trantow-Barrows   Craig Booker  Debra Henley
        3   737550  Fritsch, Russel and Anderson   Craig Booker  Debra Henley
        4   146832                   Kiehn-Spinka  Daniel Hilton  Debra Henley

              Product  Quantity  Price     Status
        0         CPU         1  30000  presented
        1    Software         1  10000  presented
        2  Maintenance         2   5000    pending
        3         CPU         1  35000   declined
        4         CPU         2  65000        won
```

```
In [4]: data.dtypes
```

```
Out[4]: Account      int64
        Name        object
        Rep         object
        Manager     object
        Product     object
        Quantity     int64
        Price        int64
        Status      object
        dtype: object
```

```
In [5]: # Make the Status variable a categorical variable

        data["Status"] = data["Status"].astype("category")
        data["Status"].cat.set_categories(["won", "pending", "presented", "declined"], \
                                          inplace = True)
```

```
In [6]: data.dtypes
```

```
Out[6]: Account        int64
        Name          object
        Rep           object
        Manager       object
        Product       object
        Quantity       int64
        Price          int64
        Status      category
        dtype: object
```

The simplest pivot table must have a dataframe and an index

```
In [7]: pd.pivot_table(data, index = "Name")
```

Out[7]:

| Name | Account | Price | Quantity |
|---|---|---|---|
| Barton LLC | 740150.0 | 35000.0 | 1.000000 |
| Fritsch, Russel and Anderson | 737550.0 | 35000.0 | 1.000000 |
| Herman LLC | 141962.0 | 65000.0 | 2.000000 |
| Jerde-Hilpert | 412290.0 | 5000.0 | 2.000000 |
| Kassulke, Ondricka and Metz | 307599.0 | 7000.0 | 3.000000 |
| Keeling LLC | 688981.0 | 100000.0 | 5.000000 |
| Kiehn-Spinka | 146832.0 | 65000.0 | 2.000000 |
| Koepp Ltd | 729833.0 | 35000.0 | 2.000000 |
| Kulas Inc | 218895.0 | 25000.0 | 1.500000 |
| Purdy-Kunde | 163416.0 | 30000.0 | 1.000000 |
| Stokes LLC | 239344.0 | 7500.0 | 1.000000 |
| Trantow-Barrows | 714466.0 | 15000.0 | 1.333333 |

Of course, you can have multiple indices.

```
In [8]: pd.pivot_table(data, index = ["Manager", "Rep"])
```

Out[8]:

| Manager | Rep | Account | Price | Quantity |
|---|---|---|---|---|
| Debra Henley | Craig Booker | 720237.0 | 20000.000000 | 1.250000 |
| | Daniel Hilton | 194874.0 | 38333.333333 | 1.666667 |
| | John Smith | 576220.0 | 20000.000000 | 1.500000 |
| Fred Anderson | Cedric Moss | 196016.5 | 27500.000000 | 1.250000 |
| | Wendy Yule | 614061.5 | 44250.000000 | 3.000000 |

### 1.0.1 Create a pivot table that shows you the total quantity for each rep

```
In [9]: # by default the aggregation is by average
        pd.pivot_table(data, index = "Rep", values = "Quantity")
```

```
Out[9]:                Quantity
        Rep
        Cedric Moss    1.250000
        Craig Booker   1.250000
        Daniel Hilton  1.666667
        John Smith     1.500000
        Wendy Yule     3.000000
```

```
In [10]: pd.pivot_table(data, index = "Rep", values = "Quantity", aggfunc = np.sum)
```

```
Out[10]:                Quantity
         Rep
         Cedric Moss           5
         Craig Booker          5
         Daniel Hilton         5
         John Smith            3
         Wendy Yule           12
```

### 1.0.2 Create a pivot table that shows you the total quantity for each rep, and the max number of units per transaction

```
In [11]: pd.pivot_table(data, index = "Rep", values = "Quantity", \
                        aggfunc = [np.sum, max])
```

```
Out[11]:                    sum      max
                        Quantity Quantity
         Rep
         Cedric Moss           5        2
         Craig Booker          5        2
         Daniel Hilton         5        2
         John Smith            3        2
         Wendy Yule           12        5
```

### 1.0.3 Create a pivot table that shows you the number of units sold by each rep for each product

```
In [12]: pd.pivot_table(data, index = "Rep", values = "Quantity", \
                        columns = "Product", aggfunc = np.sum)
```

```
Out[12]: Product        CPU  Maintenance  Monitor  Software
         Rep
         Cedric Moss    3.0          1.0      NaN       1.0
         Craig Booker   2.0          2.0      NaN       1.0
         Daniel Hilton  4.0          NaN      NaN       1.0
         John Smith     1.0          2.0      NaN       NaN
         Wendy Yule     7.0          3.0      2.0       NaN
```

```
In [13]: # replace NaN's with zeros

         pd.pivot_table(data, index = "Rep", values = "Quantity", \
                        columns = "Product", aggfunc = np.sum, fill_value = 0)
```

Out[13]:

| Product | CPU | Maintenance | Monitor | Software |
|---|---|---|---|---|
| Rep | | | | |
| Cedric Moss | 3 | 1 | 0 | 1 |
| Craig Booker | 2 | 2 | 0 | 1 |
| Daniel Hilton | 4 | 0 | 0 | 1 |
| John Smith | 1 | 2 | 0 | 0 |
| Wendy Yule | 7 | 3 | 2 | 0 |

#### 1.0.4 Create a pivot table that shows you the number of units sold by each rep for each product with the margins of the table showing

```
In [14]: pd.pivot_table(data, index = "Rep", values = "Quantity", \
                        columns = "Product", aggfunc = np.sum, fill_value = 0, margins = True)
```

Out[14]:

| Product | CPU | Maintenance | Monitor | Software | All |
|---|---|---|---|---|---|
| Rep | | | | | |
| Cedric Moss | 3.0 | 1.0 | 0.0 | 1.0 | 5.0 |
| Craig Booker | 2.0 | 2.0 | 0.0 | 1.0 | 5.0 |
| Daniel Hilton | 4.0 | 0.0 | 0.0 | 1.0 | 5.0 |
| John Smith | 1.0 | 2.0 | 0.0 | 0.0 | 3.0 |
| Wendy Yule | 7.0 | 3.0 | 2.0 | 0.0 | 12.0 |
| All | 17.0 | 8.0 | 2.0 | 3.0 | 30.0 |