Topic for the class-Pivot tables

Unit _3 : Title-Descriptive statistics

Date & Time : 2.9.24 11.00 AM – 11.50 AM

Dr. Bhramaramba Ravi

Professor

Department of Computer Science and Engineering

GITAM School of Technology (GST)

Visakhapatnam – 530045

Email: **bravi@gitam.edu**

Unit3-syllabus

- **UNIT 3 Descriptive statistics 9 hours, P - 2 hours**
- Measures of Central Tendency – Measures of Variation – Quartiles and Percentiles –

  Moments – Skewness and Kurtosis. Exploratory Data Analytics Descriptive Statistics – Mean,

  Standard Deviation, Skewness and Kurtosis – Box Plots – Pivot Table – Heat Map – Correlation Statistics – ANOVA, Random variable, Variance, covariance, and correlation- Linear transformations of random variables, Regression.
- https://www.coursera.org/learn/data-visualization-r

Pivot table

- A pivot table is a **summary tool** that wraps up or summarizes information sourced from bigger tables.
- These bigger tables could be a database, an Excel spreadsheet, or any data that is or could be converted in a table-like form.
- The data summarized in a pivot table might include sums, averages, or other statistics which the pivot table groups together in a meaningful way.
- pivot tables enable data analysts to summarize large datasets into a concise and meaningful table which can be consumed at a glance.

Pivot table

- GroupBy abstraction lets us explore relationships within a dataset.
- A *pivot table* is a similar operation that is commonly seen in spreadsheets and other programs that operate on tabular data.
- The pivot table takes simple columnwise data as input, and groups the entries into a two-dimensional table that provides a multidimensional summarization of the data.
- The difference between pivot tables and GroupBy can sometimes cause confusion; we should think of pivot tables as essentially a *multidimensional* version of GroupBy aggregation.
- That is, you splitapply-combine, but both the split and the combine happen across not a onedimensional index, but across a two-dimensional grid.

Motivating pivot tables

- we'll use the database of passengers on the *Titanic*, available through the Seaborn library
- In[1]: **import numpy as np**
- **import pandas as pd**
- **import seaborn as sns**
- titanic = sns.load_dataset('titanic')

```
In[2]: titanic.head()

Out[2]:
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \\
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

      who adult_male deck  embark_town alive  alone
0     man        True  NaN  Southampton    no  False
1   woman       False    C    Cherbourg   yes  False
2   woman       False  NaN  Southampton   yes   True
3   woman       False    C  Southampton   yes  False
4     man        True  NaN  Southampton    no   True
```

This contains a wealth of information on each passenger of that ill-fated voyage, including gender, age, class, fare paid, and much more.

Pivot tables by hand

- To start learning more about this data, we might begin by grouping it according to gender, survival status, or some combination thereof. If you be tempted to apply a GroupBy operation—for example, let's look at survival rate by gender:

```
In[3]: titanic.groupby('sex')[['survived']].mean()

Out[3]:          survived
         sex
         female  0.742038
         male    0.188908
```

Pivot tables by hand contd.

- This immediately gives us some insight: overall, three of every four females on board survived, while only one in five males survived!
- This is useful, but we might like to go one step deeper and look at survival by both sex and, say, class. Using the vocabulary of GroupBy, we might proceed using something like this: we *group by* class and gender, *select* survival, *apply* a mean aggregate, *combine* the resulting groups, and then *unstack* the hierarchical index to reveal the hidden multidimensionality. In code

```
In[4]: titanic.groupby(['sex', 'class'])['survived'].aggregate('mean').unstack()

Out[4]: class      First    Second     Third
        sex
        female  0.968085  0.921053  0.500000
        male    0.368852  0.157407  0.135447
```

Pivot tables by hand contd.

- This gives us a better idea of how both gender and class affected survival.
- This two-dimensional GroupBy is common enough that Pandas includes a convenience routine, pivot_table, which succinctly handles this type of multidimensional aggregation.

Pivot table syntax

Here is the equivalent to the preceding operation using the `pivot_table` method of `DataFrames`:

```
In[5]: titanic.pivot_table('survived', index='sex', columns='class')

Out[5]: class      First    Second     Third
        sex
        female  0.968085  0.921053  0.500000
        male    0.368852  0.157407  0.135447
```

This is eminently more readable than the `GroupBy` approach, and produces the same result. As you might expect of an early 20th-century transatlantic cruise, the survival

Pivot tables syntax

- gradient favors both women and higher classes.
- First-class women survived with near certainty (hi, Rose!), while only one in ten third-class men survived (sorry, Jack!).

Multilevel pivot tables

- Just as in the GroupBy, the grouping in pivot tables can be specified with multiple levels,
- and via a number of options. For example, we might be interested in looking at
- age as a third dimension.
- We'll bin the age using the pd.cut function:

```
In[6]: age = pd.cut(titanic['age'], [0, 18, 80])
       titanic.pivot_table('survived', ['sex', age], 'class')
```

Out[6]:

	class	First	Second	Third
sex	age			
female	(0, 18]	0.909091	1.000000	0.511628
	(18, 80]	0.972973	0.900000	0.423729
male	(0, 18]	0.800000	0.600000	0.215686
	(18, 80]	0.375000	0.071429	0.133663

THANK YOU