Interactive Pivot Tables in R

Pivot Tables are probably the most powerful feature of Microsoft Excel. Pivot tables allow cross tabulation of different variables and display values as either sums, counts, averages, etc. Moreover, user has the ability to add and remove variables and modify results accordingly. This functionality proves to be extremely useful and can actually work as a basic dashboard.

Interestingly, you can generate such Pivot tables in R! Package `rpivotTable` allows building Pivot tables and dynamic slicing and dicing of the data. Lets see how its done.

```
install.packages("rpivotTable")
library(rpivotTable)
```

We will work with R's built-in dataset "Titanic". The dataset contains information on the fate of passengers on fatal maiden voyage of ocean liner "Titanic". It is a four dimensional array generated by cross tabulating 2201 observations and 4 variables. The variables are: Class (1st, 2nd, 3rd, Crew), Sex (Male, Female), Age (Child, Adult), Survived (No, Yes). Fifth column Freq shows number of entries for each of the combinations of earlier four variables.

```
data("Titanic")
```

Generate a simple Pivot table:

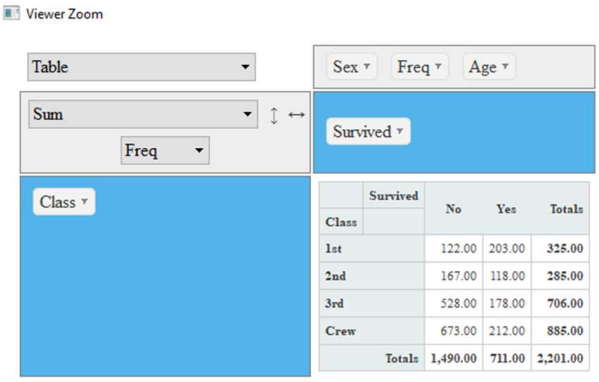
```
rpivotTable(Titanic)
```

The default `rpivotTable()` function converts data into a table. It can be viewed in RStudio's 'Viewer' pane. Let's understand basic structure of the output.

Left corner of the table indicates how the data is presented. It is 'Table' by default. If you click on the dropdown menu, a list of other output formats would be visible.
All column headers are placed in the top horizontal box.
Two blue patches represent row and column dimensions of the table, which are empty by default.
The table gives `Counts` of rows in the data.

Change output interactively:

Interactivity is the key feature of a Pivot table, let's see how to do it in `rpivotTable` . If you hover over the column headers, you will realise they are movable. You can drag and place them in the row-column blue patches to generate bivariate / multivariate tables.
Cell values can be controlled by clicking on the dropdown menu showing 'Counts' in the left corner. It can display sum, average, unique values, minimum, maximum, first, last, proportions, etc.

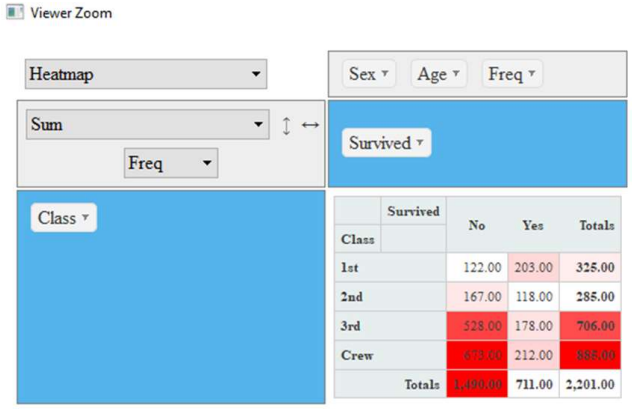


In this table, we are displaying 'Sum' of column 'Freq' based on columns 'Survived' and 'Class'. The table essentially shows how many people from which class survived or didn't survive the accident.

Output formats:

Biggest advantage of `rpivotTable()` is that it can show output not just as a simple table. Different output formats available are:

- Barcharts
- Heatmaps
- Treemaps
- Stacked Bar Chart
- Area Chart
- Line Chart
- Scatter Chart



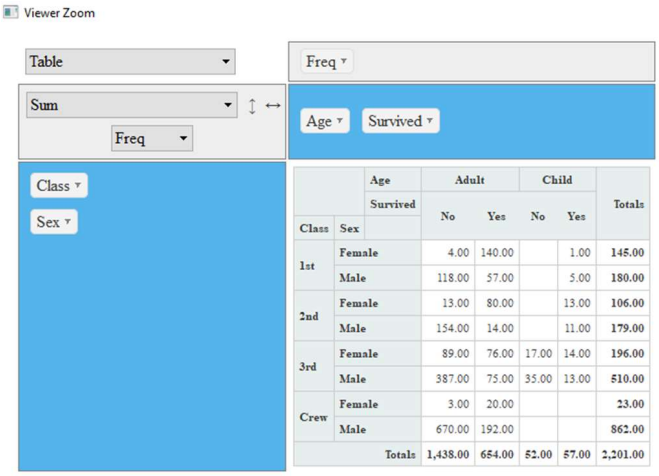
Here, we have changed the output format to 'Heatmap'. The distribution of deaths is now distinctly clear.

Pre-populate rpivotTable:

Rows and columns can be pre-populated in the output by adding the following arguments.

```
rpivotTable(Titanic, rows = c("Class","Sex"),
            cols = c("Age","Survived"), aggregatorName = "Sum", vals = "Freq")
```

`rows=` and `cols=` specify which variables should be placed where.
`aggregatorName=` gives how the values should be summarised.
`vals=` gives which variable should be considered for aggregation.



		Age	Adult		Child		Totals
		Survived	No	Yes	No	Yes	
Class	Sex						
1st	Female		4.00	140.00		1.00	145.00
	Male		118.00	57.00		5.00	180.00
2nd	Female		13.00	80.00		13.00	106.00
	Male		154.00	14.00		11.00	179.00
3rd	Female		89.00	76.00	17.00	14.00	196.00
	Male		387.00	75.00	35.00	13.00	510.00
Crew	Female		3.00	20.00			23.00
	Male		670.00	192.00			862.00
Totals			1,438.00	654.00	52.00	57.00	2,201.00