1010DATA

Basic Retail Analysis in 1010data

(212) 405.1010 | info@1010data.com | Follow: @1010data | www.1010data.com

Contents

Retail Sales Analysis

The 1010data Quick Start Guide breaks down basic operations in 1010data and how each can individually be used to perform a basic analysis. If you've read that guide you should have a basic idea on how to do things like select row, create metrics with computed columns, and summarize data sets with a tabulation or cross-tabulation. This guide will walk you through how to take those atomized operations and combine them to produce meaningful insights and analysis. The first project we'll look at is how to perform a basic market basket analysis.

Market basket analysis is broad term for the kinds of analyses used to better understand sales patterns and shopper behavior. Like any good analysis, this one starts with a simple questions: *How much, in terms of dollars, am I selling on each day of a given month?* In this instance, we want to look at a month's worth of data, and then summarize the whole month with a day by day breakdown of sales in dollars. The first step is to specify the actual month we are interested in. But first, let's take a quick look at the data we're going to analyze.

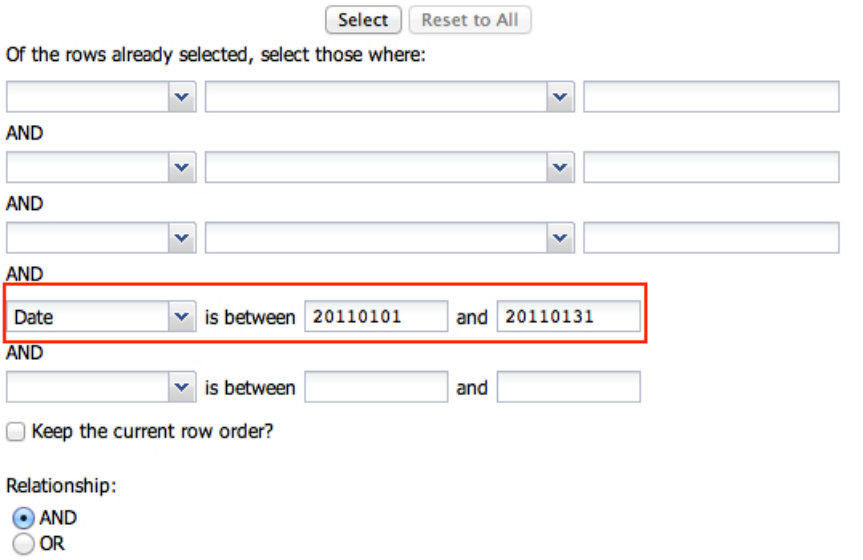
Sales Detail

Columns 1-16 of 16, Rows 1-24 of 3,314,753,767

Trans ID	Date	Time	Store	SKU	Extended Sales	Qty/ Wgt	Promo	Cost	Customer	Department	Group	Division	Sub-Division	Primary Segment	Secondary Segment
-1746773251	12/10/11	00:00:00	199	211611	0.9	1.00	0	0.65	e2c34159	35	311	4	3	high value	perimeter
-1746773251	12/10/11	00:00:00	199	92025	0.1	1.00	0	0.1	e2c34159	35	311	4	3	high value	perimeter
-1746773251	12/10/11	00:00:00	199	49061	1.65	1.00	0	0.92	e2c34159	35	384	4	3	high value	perimeter
-1746770521	12/10/11	00:00:00	70	100226	12.81	1.00	0	10.58		42	23	4	3		
-1746770521	12/10/11	00:00:00	70	335799	0.82	1.00	0	0.82		42	23	4	3		
-1746767474	12/10/11	00:00:00	164	55890	4.52	1.00	0	4.14	2a6d5f19	28	399	4	5	review	
-1746767474	12/10/11	00:00:00	164	398440	1.07	1.00	0	0.8	2a6d5f19	35	86	4	5	review	
-1746767474	12/10/11	00:00:00	164	305295	1.07	1.00	0	0.8	2a6d5f19	35	86	4	5	review	
-1746767474	12/10/11	00:00:00	164	118695	3.59	1.00	0	2.63	2a6d5f19	55	272	4	5	review	
-1746762972	12/10/11	00:00:00	181	217462	11.83	2.00	0	14.74		49	364	4	4		
-1746739362	12/10/11	00:00:00	65	303410	12.51	1.00	0	11.3		4	349	4	4		
-1746739361	12/10/11	00:00:00	65	239320	16.77	2.00	0	16.77	31d7cd04	10	136	4	4	review	
-1746739360	12/10/11	00:00:00	65	499092	1.22	1.00	0	0.99		20	448	4	4		
-1746739360	12/10/11	00:00:00	65	140199	0.08	1.00	0	0.08		20	448	4	4		
-1746736249	12/10/11	00:00:00	138	342130	1.01	2.00	0	0.51	a80e6a5e	35	311	4	4	convenience	
-1746736249	12/10/11	00:00:00	138	92025	0.2	2.00	0	0.2	a80e6a5e	35	311	4	4	convenience	
-1746736249	12/10/11	00:00:00	138	462501	2.28	1.00	0	1.28	a80e6a5e	42	23	4	4	convenience	
-1746736249	12/10/11	00:00:00	138	279697	0.09	1.00	0	0.09	a80e6a5e	42	23	4	4	convenience	
-1746733187	12/10/11	00:00:00	167	244196	3.56	1.00	0	2.36	f2a66aa9	35	494	4	5		
-1746733187	12/10/11	00:00:00	167	74749	1.93	1.00	0	1.02	f2a66aa9	35	351	4	5		
-1746733187	12/10/11	00:00:00	167	277667	1.65	0.79	0	0.56	f2a66aa9	36	65	4	5		
-1746733187	12/10/11	00:00:00	167	247560	1.72	2.39	0	0.97	f2a66aa9	36	468	4	5		
-1746733187	12/10/11	00:00:00	167	15198	0.83	1.00	0	0.2	f2a66aa9	36	65	4	5		
-1746733187	12/10/11	00:00:00	167	104786	3.2	1.00	0	2.43	f2a66aa9	35	86	4	5		

If you read the 1010data Quick Start Guide this might look familiar to you. This data contains all the same columns as the very small (35 rows) data set that was used in that tutorial. However, in this version, you can see that the number of rows is much higher (3.3 billion!). Regardless of the size of the table, both these data sets share basic information. They contain the transaction number, date of the transaction, the SKU (product identification number) for each item purchased, and the date of the transaction, among others. Now that we understand the data a little better, we're going to revisit out central question: How much in sales is my organization doing, day by day, in a month?

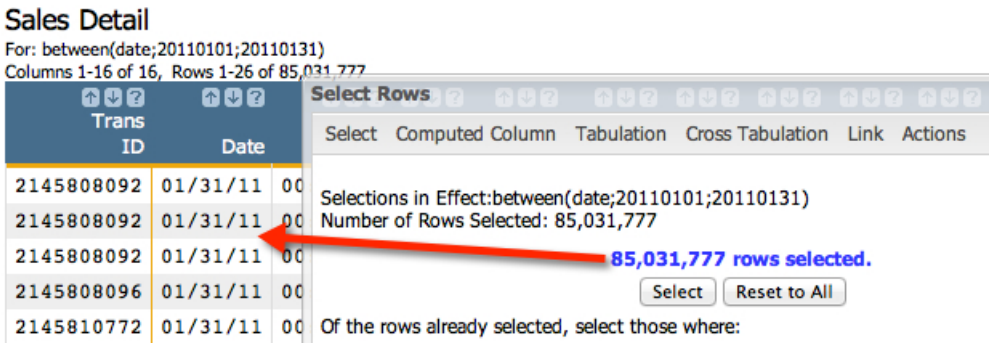
Since we are primarily interested in a single month's worth of data, the first step is to narrow down the rows in the table to only those that take place during the month we're interested in. If you'd like to use the GUI for this, feel free. But part of this guide is helping you get comfortable with the 1010data Macro Language. So we'll provide both the GUI and Macro Language way of doing things for the first few tutorials. But don't be shocked if at some point we're only working with the Macro Language. Here's our date selection in the GUI:



As you can see, we're going to look at sales for the organization in the month of January, 2011. Here's what the Macro Language code looks like:

```
<sel value="between(date;20110101;20110131)"/>
```

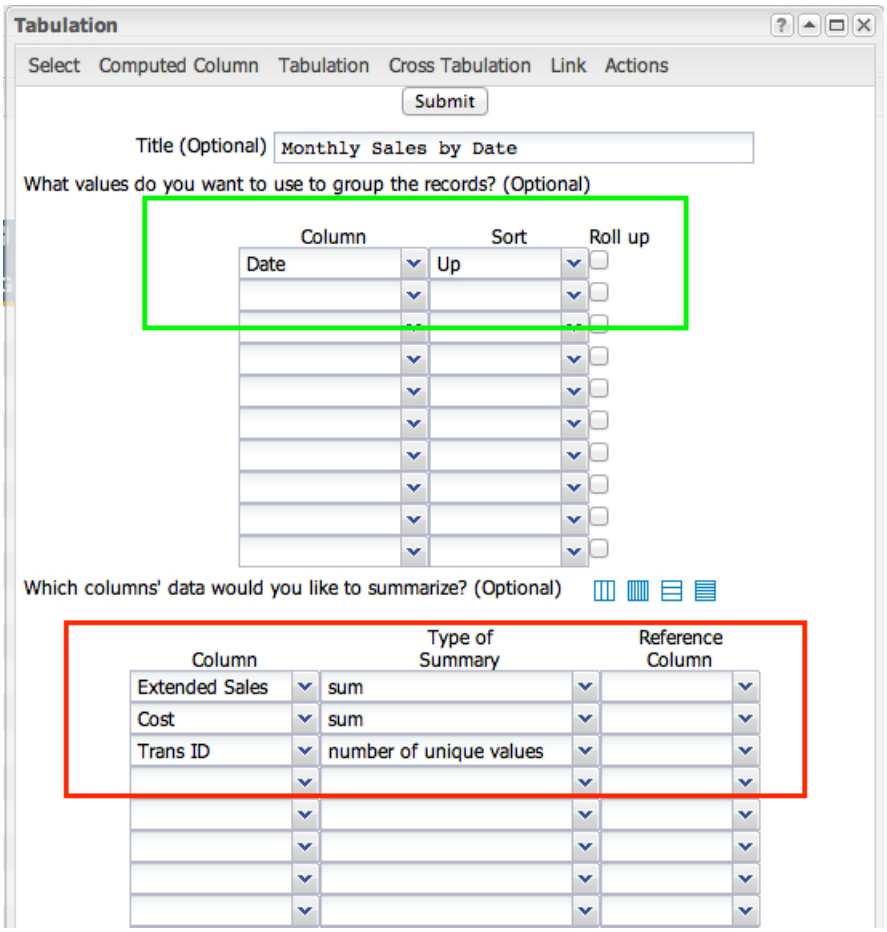
As you may have noticed, we're using the `between(X;Y;Z)` function to select our date range. `between(X;Y;Z)` is great for date selections, and has many other useful purposes. Once the selection is made, we'll go from our initial 3.3 billion rows of data to a more manageable 85 million, as shown below:



After the row selection executes, you should see that every remaining row (more than 85 million) falls within the date range specified. Now we can see every item sold in every basket for the entire month. But remember, we're interested in total sales dollars for each day of the month, and don't particularly care about specific items or transactions (for now). If you remember the basics of performing a tabulation, then it should be fairly clear that we want to group by the date and view the sum of sales for each day. Let's take a look at the tabulation settings. Go to **Analysis** > **Tabulation...**. Here are our requirements for the tabulation:

- Group by the date
- View the sum of sales
- View the total costs for the items sold on each day
- View the number of transactions on each day

Let's take a look at the **Tabulation...** dialog to understand how to create the aggregations we're interested in. Then, we'll quickly run through the Macro Language code. Here's the dialog:



Now that we've defined our grouping metric and the summarizations we're interested in, click **Submit** to get the results:

Monthly Sales by Date
For: between(date;20110101;20110131)
Columns 1-4 of 4, Rows 1-22 of 31

Date	Sum of Extended Sales	Sum of Cost	Num of Unique Values in Trans ID
	308,040,557.45	222,121,208.61	
01/01/11	7,987,628.49	5,703,583.20	292,140
01/02/11	11,563,261.67	8,570,742.30	333,831
01/03/11	11,956,458.05	8,928,963.74	361,694
01/04/11	10,681,792.31	7,913,031.68	354,896
01/05/11	9,943,805.77	7,282,138.37	328,924
01/06/11	9,510,073.53	6,872,652.59	323,327
01/07/11	10,575,250.49	7,707,715.24	346,191
01/08/11	13,088,470.51	9,635,585.41	366,944
01/09/11	12,735,734.12	9,192,675.66	349,558
01/10/11	10,267,361.11	7,257,613.98	337,633
01/11/11	8,750,032.17	6,144,318.22	322,882
01/12/11	8,408,308.69	5,923,799.06	318,777
01/13/11	8,204,657.34	5,786,467.21	313,038
01/14/11	9,580,069.17	6,857,852.36	345,009
01/15/11	11,959,562.17	8,770,719.68	361,589
01/16/11	11,541,183.20	8,468,829.95	339,847
01/17/11	9,337,652.99	6,492,288.22	332,411
01/18/11	8,414,554.55	5,902,993.68	324,426
01/19/11	7,908,934.83	5,535,431.46	314,785
01/20/11	7,918,868.30	5,569,143.26	311,714
01/21/11	9,460,493.39	6,843,492.29	350,622
01/22/11	12,043,503.23	8,873,236.84	372,355

As you can see, we've produced a tidy summarization that clearly lists sales totals for each date within January, 2011. And just in case you're interested, here's a nicely commented Macro Language version of this analysis:

```
<note>First step is to select the month (or other time period) of interest</
note>
<sel value="between(date;20110101;20110131)"/>

<note>Next, create a tabulation with the tabu element. The breaks attribute
 specifies the column to group by.</note>
<tabu label="Monthly Sales by Date" breaks="date">

    <note>The break element tells the system how to sort the values in the
 group-by column</note>
    <break col="date" sort="up"/>

    <note>This tcol element is the first summarization in the tabulation. It
 produces the sum of extended sales</note>
    <tcol name="sumsales" source="xsales" fun="sum" label="Sum
 of`Extended`Sales"/>


    <note>This tcol creates a sum of the total cost for each date</note>
    <tcol name="sumcost" source="cost" fun="sum" label="Sum of`Cost"/>

    <note>This tcol counts the number of unique transaction IDs for each
 date</note>
```

```
    <tcol name="transucnt" source="transid" fun="ucnt" label="Num
 of`Unique`Values`in`Trans`ID"/>
</tabu>
```

And just in case you don't care about the comments, here's the code the system produces (or which a savvy 1010data user might write):

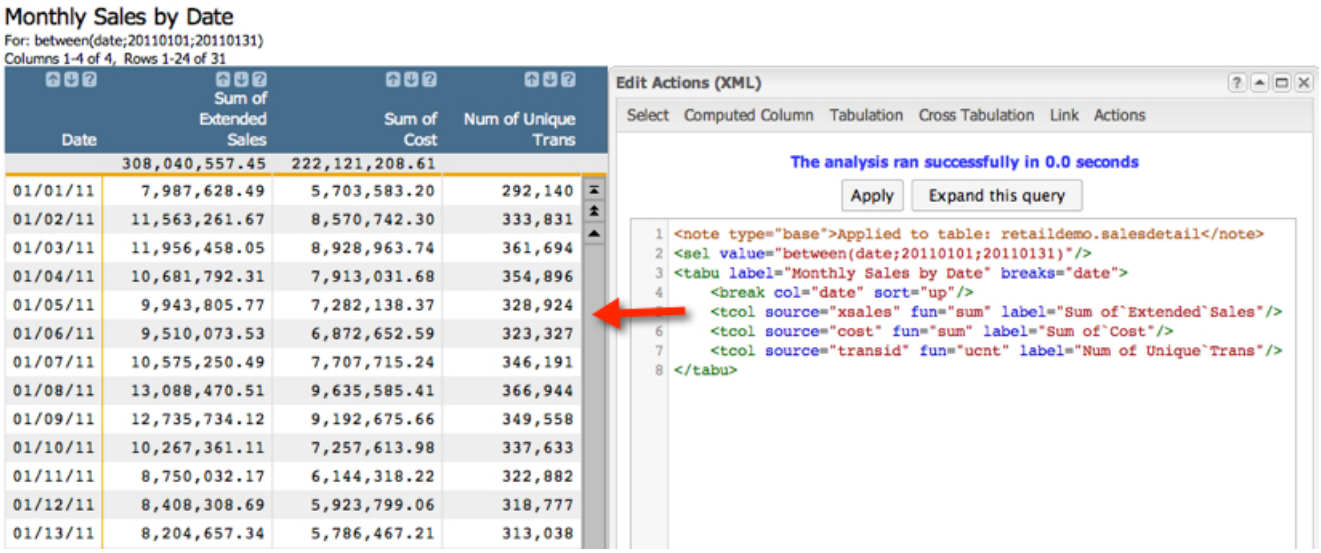
```
<sel value="between(date;20110101;20110131)"/>
<tabu label="Monthly Sales by Date" breaks="date">
    <break col="date" sort="up"/>
    <tcol name="sumsales" source="xsales" fun="sum" label="Sum
 of`Extended`Sales"/>
    <tcol name="sumcost" source="cost" fun="sum" label="Sum of`Cost"/>
    <tcol name="transucnt" source="transid" fun="ucnt" label="Num
 of`Unique`Values`in`Trans`ID"/>
</tabu>
```

Producing a basic summary of sales by date for a given time period is a very common way of measuring retail sales performance. In 1010data it can be performed on truly massive data sets in a matter for a couple minutes. Try your hand and playing with these values and altering the analysis to answer your own questions. For instance, as a follow up, try to determine the margin for each day of the month.

Next up, we'll take this analysis one step further by determining what the most profitable weekday of the month is.

Most Profitable Weekday in Month

The next step in our basic analysis of sales data is to determine what the most profitable weekday was for a given month (or whatever period of time you prefer to evaluate). We'll start with the tabulation we have already produced, which give us totals for sales and cost for each date in January, 2011, as follows:



Our next objective is to determine what week day (Sunday - Saturday) was the most profitable overall. To do this we only need to perform three basic steps:

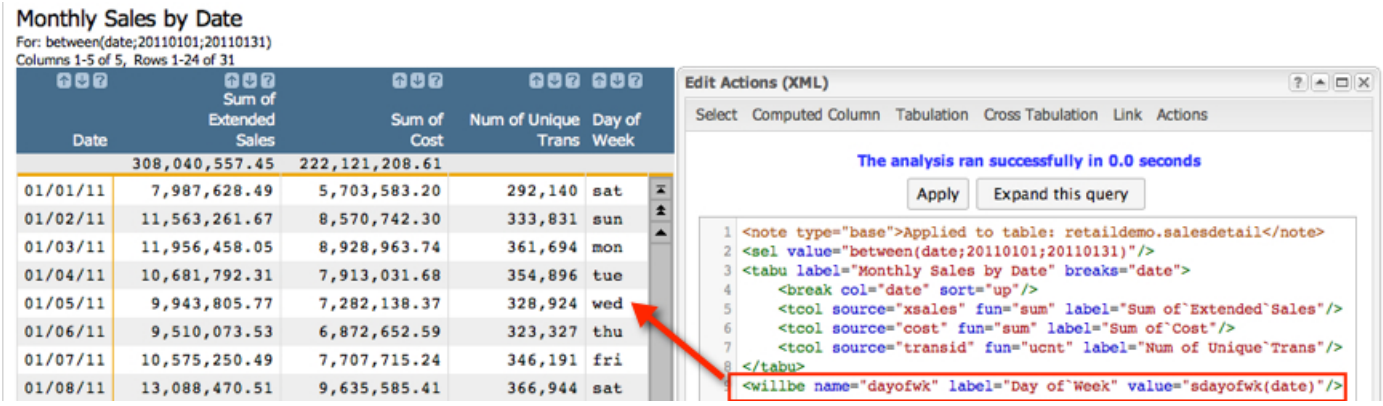
1. Create a computed column that assigns a week day for each date in the table
2. Create a tabulation that groups by week day and summarizes sales and cost for each one
3. Create a computed column that calculates margin (aka profit!)

Step 1: Assign a week day to each date. While we could do this by going to **Columns** > **Create Computed Column...**, we're actually going to do everything for this part of the analysis in the Macro Language. Start with the code we generated last time and then add the last line in bold:

```
<sel value="between(date;20110101;20110131)"/>
<tabu label="Monthly Sales by Date" breaks="date">
    <break col="date" sort="up"/>
    <tcol name="sumsales" source="xsales" fun="sum" label="Sum
 of`Extended`Sales"/>
    <tcol name="sumcost" source="cost" fun="sum" label="Sum of`Cost"/>
    <tcol name="transucnt" source="transid" fun="ucnt" label="Num
 of`Unique`Values`in`Trans`ID"/>
</tabu>

<note>Create a computed column that assigns a week day to each date using the
 sdayofwk(X) function</note>
<willbe name="dayofwk" label="Day of`Week" value="sdayofwk(date)"/>
```

The line above produces the following results:



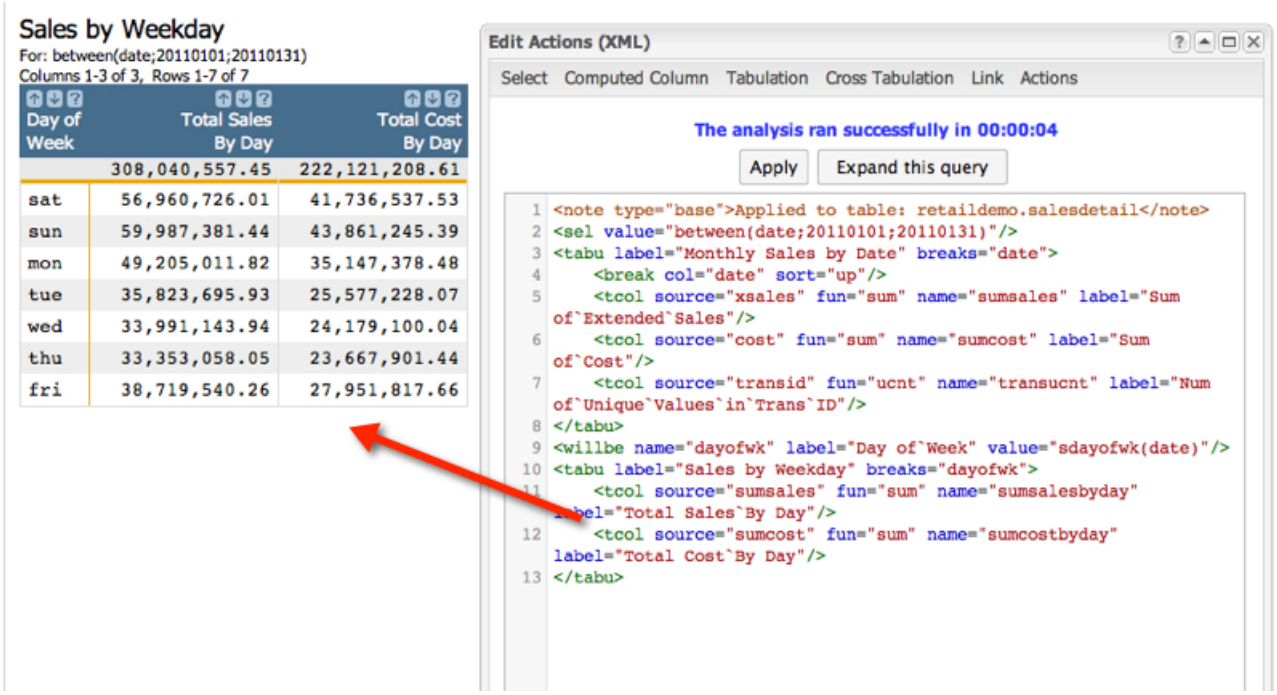
Step 2, perform a tabulation and use the new column we just created as the group by column:

```
<sel value="between(date;20110101;20110131)"/>
<tabu label="Monthly Sales by Date" breaks="date">
    <break col="date" sort="up"/>
    <tcol name="sumsales" source="xsales" fun="sum" label="Sum
 of`Extended`Sales"/>
    <tcol name="sumcost" source="cost" fun="sum" label="Sum of`Cost"/>
    <tcol name="transucnt" source="transid" fun="ucnt" label="Num
 of`Unique`Values`in`Trans`ID"/>
</tabu>

<note>Create a computed column that assigns a week day to each date using the
 sdayofwk(X) function</note>
<willbe name="dayofwk" label="Day of`Week" value="sdayofwk(date)"/>

<note>Tabulate using sdayofwk as the grouping column. Summarize both sales and
 cost</note>
<tabu label="Sales by Weekday" breaks="dayofwk">
    <tcol name="sumsalesbyday"source="sumsales" fun="sum" label="Total
 Sales`By Day"/>
    <tcol name="sumcostbyday" source="sumcost" fun="sum" label="Total Cost`By
 Day"/>
</tabu>
```

Notice here that we are tabulating a tabulation. No problem. Tabulations are just regular old 1010data worksheets, and can be manipulated the exact same ways as any other table or worksheet. Here are the results of the last tabulation:



Step 3, calculate margin (profits rule!!). To do this, create another computed column. Again, feel free to do this in the GUI, but we'll stick to the Macro Language for this example:

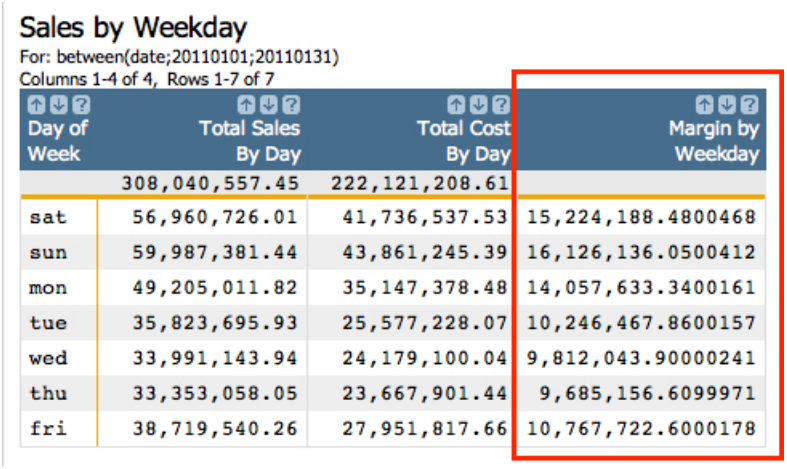
```
<sel value="between(date;20110101;20110131)"/>
<tabu label="Monthly Sales by Date" breaks="date">
    <break col="date" sort="up"/>
    <tcol name="sumsales" source="xsales" fun="sum" label="Sum
 of`Extended`Sales"/>
    <tcol name="sumcost" source="cost" fun="sum" label="Sum of`Cost"/>
    <tcol name="transucnt" source="transid" fun="ucnt" label="Num
 of`Unique`Values`in`Trans`ID"/>
</tabu>

<note>Create a computed column that assigns a week day to each date using the
 sdayofwk(X) function</note>
<willbe name="dayofwk" label="Day of`Week" value="sdayofwk(date)"/>

<note>Tabulate using sdayofwk as the grouping column. Summarize both sales and
 cost</note>
<tabu label="Sales by Weekday" breaks="dayofwk">
    <tcol name="sumsalesbyday"source="sumsales" fun="sum" label="Total
 Sales`By Day"/>
    <tcol name="sumcostbyday" source="sumcost" fun="sum" label="Total Cost`By
 Day"/>
</tabu>

<note>Create a computed column to calculate margin by weekday</note>
<willbe name="marginbyweekday" label="Margin by`Weekday" value="sumsalesbyday
 - sumcostbyday"/>
```

Run the code and you should get the following results:



Sales by Weekday

For: between(date;20110101;20110131)

Columns 1-4 of 4, Rows 1-7 of 7

Day of Week	Total Sales By Day	Total Cost By Day	Margin by Weekday
	308,040,557.45	222,121,208.61	
sat	56,960,726.01	41,736,537.53	15,224,188.4800468
sun	59,987,381.44	43,861,245.39	16,126,136.0500412
mon	49,205,011.82	35,147,378.48	14,057,633.3400161
tue	35,823,695.93	25,577,228.07	10,246,467.8600157
wed	33,991,143.94	24,179,100.04	9,812,043.90000241
thu	33,353,058.05	23,667,901.44	9,685,156.6099971
fri	38,719,540.26	27,951,817.66	10,767,722.6000178

Our results tell us that Saturday and Sunday are by far our most profitable days of the week. This makes a lot of sense, since that's when most people actually have time to go grocery shopping. Still, it's nice to not only know for sure, but be able to attach hard numbers to the phenomenon. But it does raise another question. It is intuitive that our highest margins, in terms of total dollars, are on the same days we have the highest sales, also in terms of total dollars. However, if we want to think about this same relationship in terms of percentages, does the same dynamic hold true?

Just for fun, let's create one last computed column. This time we're going to calculate margin as a percent of cost, to understand if we're profiting by a higher or lower percentage of our cost on a given day of the week. Remember, this will only apply to January 2011, but we could easily use this same process to understand these relationships across any time period we like.

```
<sel value="between(date;20110101;20110131)"/>
<tabu label="Monthly Sales by Date" breaks="date">
    <break col="date" sort="up"/>
    <tcol source="xsales" fun="sum" name="sumsales" label="Sum
 of`Extended`Sales"/>
    <tcol source="cost" fun="sum" name="sumcost" label="Sum of`Cost"/>
    <tcol source="transid" fun="ucnt" name="transucnt" label="Num
 of`Unique`Values`in`Trans`ID"/>
</tabu>
<willbe name="dayofwk" label="Day of`Week" value="sdayofwk(date)"/>
<tabu label="Sales by Weekday" breaks="dayofwk">
    <tcol source="sumsales" fun="sum" name="sumsalesbyday" label="Total
 Sales`By Day"/>
    <tcol source="sumcost" fun="sum" name="sumcostbyday" label="Total Cost`By
 Day"/>
</tabu>
<willbe name="marginbyweekday" label="Margin by`Weekday" format="dec:2"
 value="sumsalesbyday - sumcostbyday"/>

<note>Create a computed column to calculate margin as a percentage of total
 cost</note>
<willbe name="percentmargin" label="Margin as Percent`of Cost"
 value="(marginbyweekday/sumcostbyday)*100" format="dec:2"/>
```

Click the **Submit** button to run the code. Let's take a close look at the results:

Sales by Weekday

For: between(date;20110101;20110131)

Columns 1-5 of 5, Rows 1-7 of 7

Day of Week	Total Sales By Day	Total Cost By Day	Margin by Weekday	Margin as Percent of Cost
	308,040,557.45	222,121,208.61		
sat	56,960,726.01	41,736,537.53	15,224,188.48	36.48
sun	59,987,381.44	43,861,245.39	16,126,136.05	36.77
mon	49,205,011.82	35,147,378.48	14,057,633.34	40.00
tue	35,823,695.93	25,577,228.07	10,246,467.86	40.06
wed	33,991,143.94	24,179,100.04	9,812,043.90	40.58
thu	33,353,058.05	23,667,901.44	9,685,156.61	40.92
fri	38,719,540.26	27,951,817.66	10,767,722.60	38.52

Now this is interesting. While Saturday and Sunday are our two highest margin days in terms of total dollars, they actually represent the two days where our margin represents the least profit as a percentage of total cost. What could account for this apparent inversion of the relationship?

If you're interested in taking this process one step further to discover what drives these results, here are some questions you might ask:

- Are higher margin items sold in a higher percentage of the total volume of items during week days, as opposed to weekends?
- Are lower margin items sold in a higher percentage of the total volume of items during weekend days, as opposed to regular weekdays?
- Do these results reflect a seasonal result or do they hold their integrity throughout the entire year?

You can answer all the questions above with the basics building blocks we've already covered in this tutorial. However, as always, the most interesting questions are the ones you invent yourself.