Paper 327-2012

Speeding Up Discovery

John Sall, SAS Institute, Cary, NC, USA

ABSTRACT

With fast hardware and multithreaded software, interactive exploration becomes feasible in larger problems. But that also means that it will take some resourceful techniques, combined with human judgment, to look across thousands of variables to find the relationships that are worth looking at.

INTRODUCTION

You know the frustration. You are used to working with small to medium sized data tables, and all the answers seem to be instant. Then you get a large table, and it seems to take forever. Where you used to be in "flow" – pursuing the analysis without restraint, one thing leading smoothly to the next – now you are stalled, waiting for the next step. You feel guilty about spending time waiting and decide to start another activity while you are waiting for the computer. Then when you return later, you've lost the thread of thought and have to work hard to recover it.

In years past, we used to tolerate and even plan for the wait because we knew that computers took a while to do their work. But now, many iterations of Moore's Law later, we have higher expectations. Our machines are not only fast, but they also have multiple cores, and we expect our software to put them all to work, through multithreading. Our computers have huge memories, now, typically 2-4 GB, but many have much more. Many of us have SSDs, semiconductor storage devices, disk drives that aren't disks anymore, but store data in flash memory, with very little latency and very high transfer rates.

Now we should be more demanding, so that we can have that "flow" experience with much larger tables.

Speed is also important to the user interface. When you drag-and-drop variables into Graph Builder in JMP, you don't want to wait for something to happen. When you make a selection in a histogram or scatterplot, you want it to select quickly.

JMP 10 makes these interactions with your data much faster. A number of routines were multithreaded in the last two versions of JMP, and now we pursued performance much more. We found many bottlenecks that were slowing things down, and we worked on them.

So how well does JMP 10 preserve the "flow" experience? I tested JMP 10 with three data tables, which I refer to as *small*, *tall* and *wide*. You can access the data and try it yourself — the data files are all publicly data accessible online.

UPGRADE YOUR HARDWARE

The first thing to do before tackling large jobs is to get a bigger machine. Hardware and memory are incredibly cheap. Your investment of $2,000 or more in a higher-end laptop will repay its investment very quickly in terms of increased range and productivity. I got a machine with one of the newer Intel® Core™ i7 processors (2920XM), with four cores (hyperthreaded to perform like eight) and 16 GB of memory. Be sure to get the 64-bit edition of the operating system so that it can put all that memory to good use. I know that most people can get along just fine with a $600 laptop, but you are an analytics professional. Your work deserves a bigger machine, and the bigger machine is still incredibly cheap compared to prices of much smaller machines just a few years ago.

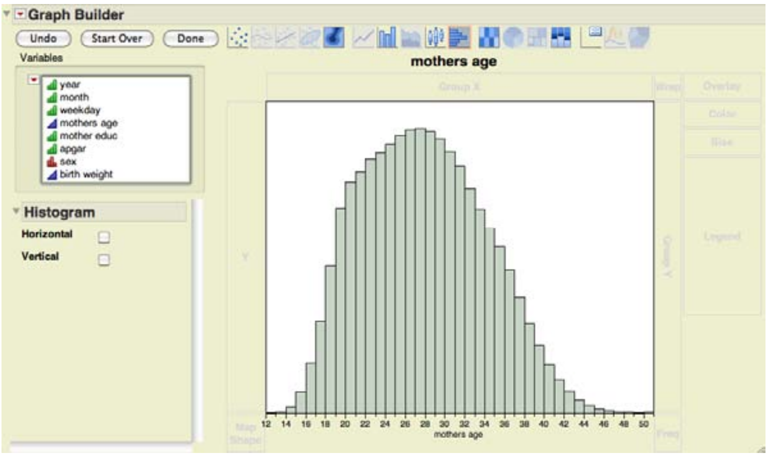
THE 'SMALL' DATA IS BIRTH RECORDS

The *small* table is all the birth records from the CDC over three years, 12.8 million records, small only in comparison to the other examples. Just go to http://www.cdc.gov/nchs/data_access/Vitalstatsonline.htm and start downloading. You will find the "Users Guide to the 2008 Natality Public Use File" containing the descriptions of all the fields. This code book is consistent across a number of years. Looking at the code book, open one of the data files under Text with Preview and start identifying fields you are interested in. I picked out 14 fields. After reading the data, a script is stored in the "source" table property. Copy the script and paste into a new script window, changing the details so that you can use this script for the next few files, each file holding one year. I edited the script a little, and you are welcome to use it to avoid the preview step. (See APPENDIX for script.) Then concatenate the three tables into one and compress columns.

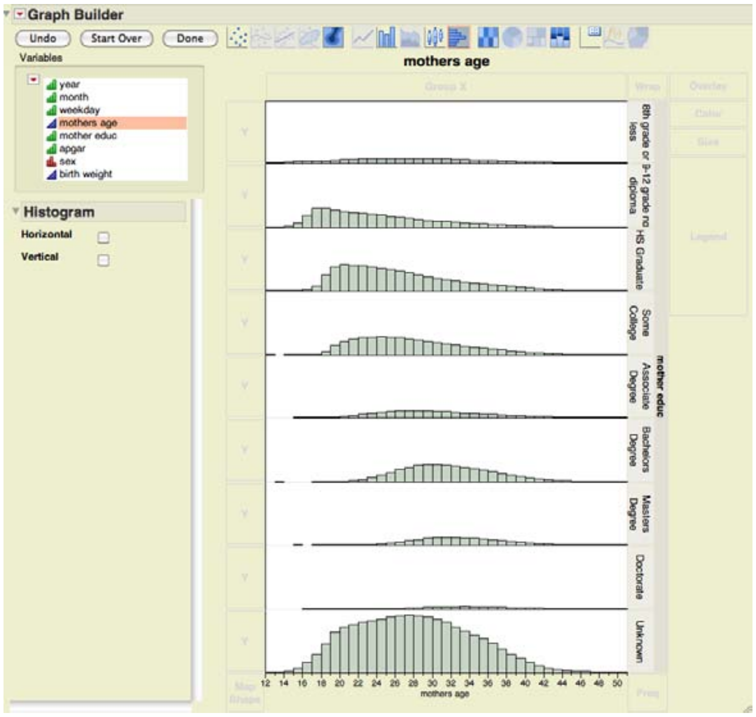
ANALYZE THE ‘SMALL’ DATA

The birth records *small* data, while 12.8 million records, is still small enough to get near-instant analysis. So you might as well do it the fun way, using drag-and-drop on Graph Builder.

So I dragged “mothers age” to the X-axis drop zone and changed the graph to “Histogram.” Actually, I prefer Histogram as the default, so I have switched the Graph Builder preferences to that. The data is not particularly normal, rising more sharply at the low end than the high end.

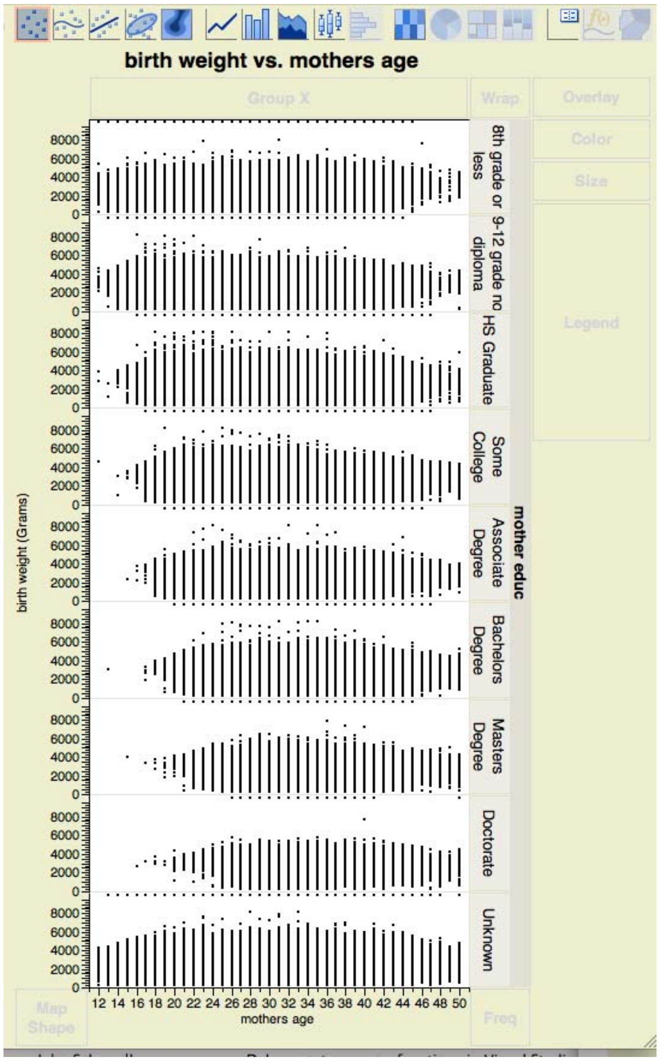


Drag “mother educ” to the Group Y drop zone, and change the graph to Histogram. Drag the resize corner to give more room for the group labels.

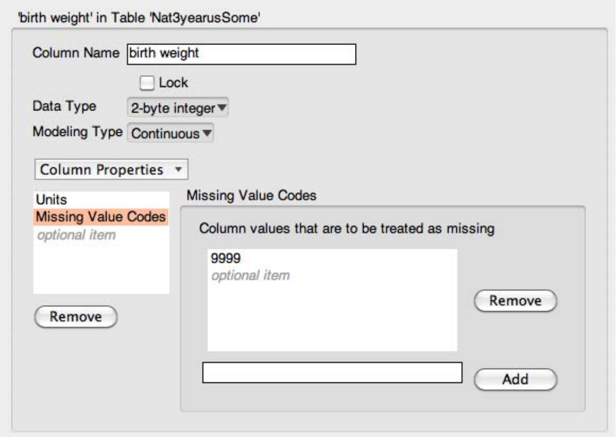


You start to see that most mothers’ education is “unknown.”

Now drag “birth weight” to the Y-axis, click in the Points icon and deselect the Smoother, resulting in this scatterplot.



Here I got suspicious, because there was a long row of points with identical values at the top of the scale. I clicked on one of these to select it, went to the data table and chose Rows->Next Selected to find it as row 24663 where birth weight happened to be 9999. Now babies just don’t come that heavy (10 kg, 22 pounds), and anything that is all nines is likely to be a code for missing. So I brought up the Column Info dialog for birth weight, added a Missing Value Codes property and entered 9999 as a missing value code. Then I returned to Graph Builder and selected Redo Analysis.


'birth weight' in Table 'Nat3yearusSome'

Column Name: birth weight

☐ Lock

Data Type: 2-byte integer

Modeling Type: Continuous

Column Properties

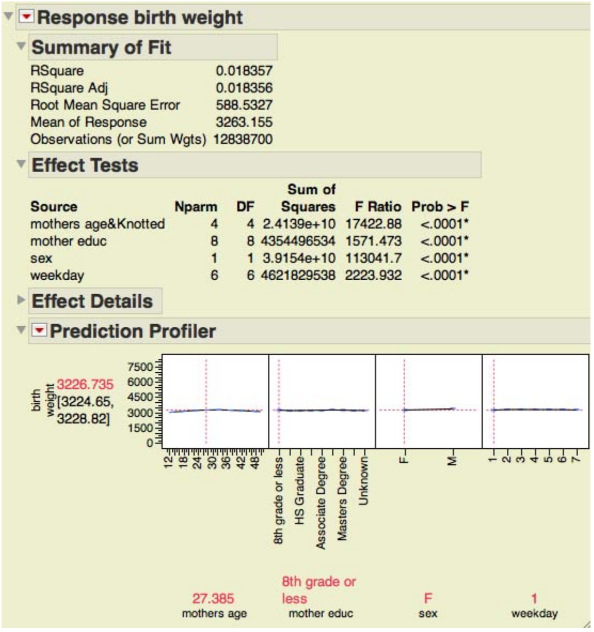
Units
Missing Value Codes
optional item

Missing Value Codes

Column values that are to be treated as missing

9999
optional item

Now I want to fit a model, so I chose Fit Model, giving mother’s age a knotted spline, and adding mother’s education, sex of baby and weekday.



Response birth weight

Summary of Fit

```
RSquare                        0.018357
RSquare Adj                    0.018356
Root Mean Square Error         588.5327
Mean of Response               3263.155
Observations (or Sum Wgts)     12838700
```

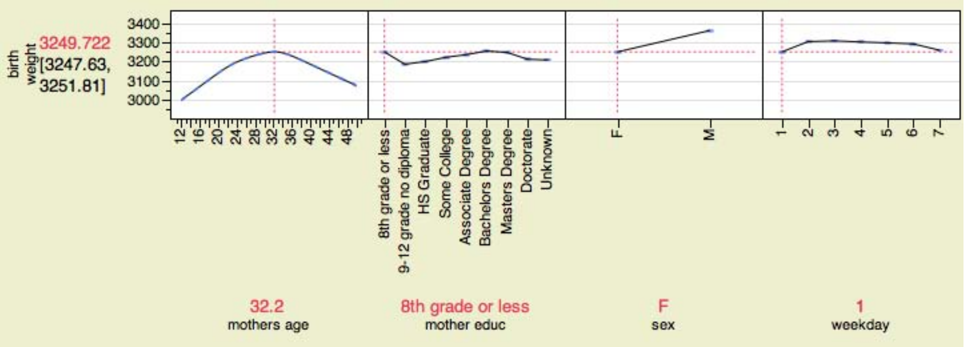
Effect Tests

```
                                          Sum of
Source                 Nparm   DF        Squares    F Ratio   Prob > F
mothers age&Knotted        4    4     2.4139e+10   17422.88    <.0001*
mother educ                8    8     4354496534   1571.473    <.0001*
sex                        1    1     3.9154e+10   113041.7    <.0001*
weekday                    6    6     4621829538   2223.932    <.0001*
```

Effect Details

Prediction Profiler

This fit for 20 parameters on 12.8 million rows took only 11 seconds. We are still in “flow.” Very little of the variation was explained by the model, but all the terms are significant. However, all the effect sizes are small, as you can see in the Profiler. They are significant because we have a huge sample size, so very small effects can still be very significant. Though the effect sizes are small, we can expand the scale in the Profiler to see the small effects, just using the cursor to drag directly on the y scale.



We see that mothers aged 32 tend to have the largest babies. High-school non-graduates have the smallest. Males are generally 110 grams heavier. And babies come smaller on weekends by around 40-50 grams. While all these effects are small, they could be interesting. Why are weekend babies smaller? I would guess that it has nothing to do with the babies themselves, but more a function of planned deliveries avoiding weekends, thus making emergency deliveries relatively more common on weekends.

We conclude that 12.8 million rows can be analyzed in “flow,” with very little wait.

THE ‘TALL’ DATA IS AIRLINE FLIGHT INFORMATION

The *tall* table is all airline flights from 1987 (third quarter) through 2008, 123 million records, the ASA Challenge problem from 2009. The download site for the data is http://stat-computing.org/dataexpo/2009/ Preparing the data will take a few hours because there is a separate file for each year, and each file is big and needs decompressing and some preparation. The files are particularly easy to read because the format is very simple, csv (comma-separated variables) with a header containing the names; so just open with the “Best Guess” option. Then you use Tables->Concatenate to obtain one big file. There are many things you can do to improve the file. Some fields that should be numeric are character because of missing-value codes, like “NA” – so change them. You can make formula columns to obtain timestamp data from the various date and time fields. You can update-merge in latitude and longitude for the airports by code. You can convert the times to UTC so that they are comparable.

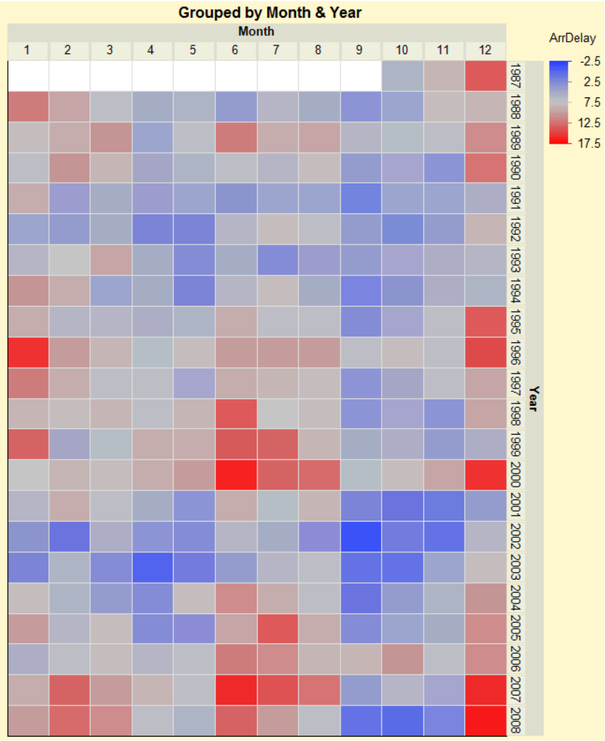
More importantly, you should compress the columns to save on memory and disk space and to improve load and save times. Just select all the variables and do Cols->Compress Selected Columns. This will take a while, but the time will be repaid many times over when you are analyzing the file. If you are short on memory, do this before you concatenate all the files.

I have a compressed version of this table with all 123,534,969 rows, but just eight of the columns, and it takes only 1.48 GB on disk and opens in 1.6 seconds. A wider version takes longer, 13-23 seconds. Making list-check columns helps a lot. When you have more than 255 categories, you can make a numeric-coded version using an add-in feature “Compress to Labeled Code” that I wrote that you can find on the JMP File Exchange via the JMP website, jmp.com.

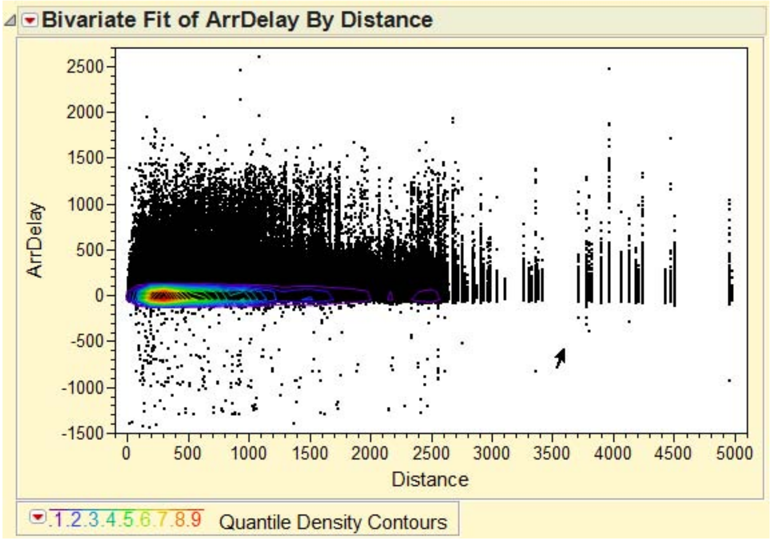
ANALYZE THE ‘TALL’ DATA

Opening the “tall” airline flight data takes just 1.7 seconds, pretty fast for 123 million rows. Another time, it took 4.6 seconds. It varies depending on the current state of paged memory condition and SSD disk cache. But even 4.6 seconds is fast.

I want to know about arrival delay. First, I use Graph Builder, dragging Year to the Group Y axis, Month to the Group X axis, switching to Heatmap, and then dragging ArrDelay to the Color role. Each drag takes a few seconds. If you save a script and do it in one step, the whole thing takes 12.4 seconds. The result is a very nice display showing in red all those months plagued by long delays. Some months in the spring and fall have very few delays, but the summer months and winter months have lots of dark red delays, probably due to bad weather. Some years are relatively delay-free, such as 1991-1994 and 2002-2003. Some years are bad, including 2000 and 2007-2008.

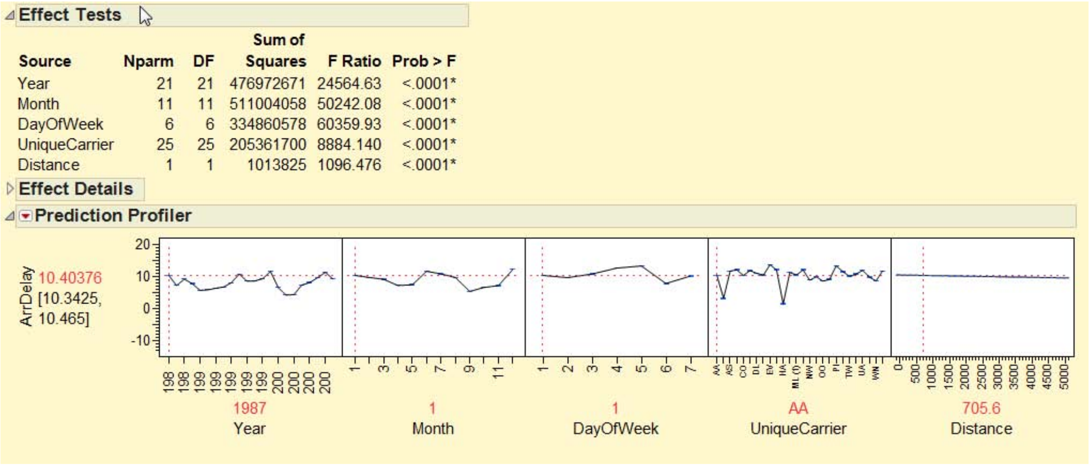


Are delays related to the distance of the flight? I did a Bivariate Fit of ArrDelay by Distance, resulting in this, below. Notice that longer-distance flights come in only a few trip lengths. Notice that there seem to be lots of values that are very negative. If you drag a selection box around all the points below -700, you will find that it selects only 146 points. This is very few compared to the 123 million points in the plot. These points must have validity issues on the shorter flights because it is not possible to be 700 minutes early on a very short flight.



Though there are points all around, we know most flights do not have extreme delays, so we need a better perspective on where most of the data is. The Nonpar Density can do this, putting density contours around where most of the data is. Most of the data is in a narrow band. Only 10% is outside the outermost purple contours.

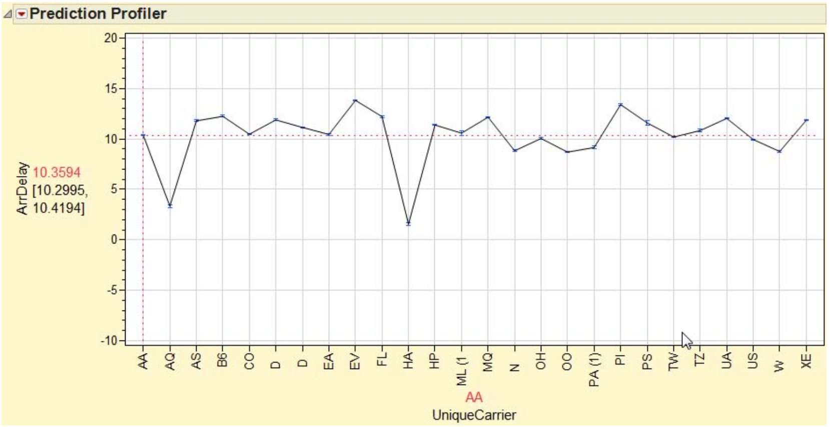
We fit a model of arrival delay by year, month, day of week, carrier and distance, with very little predictiveness, and effects only showing up on the Profiler when you expand the y scale.



Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Year	21	21	476972671	24564.63	<.0001*
Month	11	11	511004058	50242.08	<.0001*
DayOfWeek	6	6	334860578	60359.93	<.0001*
UniqueCarrier	25	25	205361700	8884.140	<.0001*
Distance	1	1	1013825	1096.476	<.0001*

Expanding the carrier factor, we see that the two Hawaiian airlines AQ (Aloha) and HA (Hawaiian) are relatively less late.



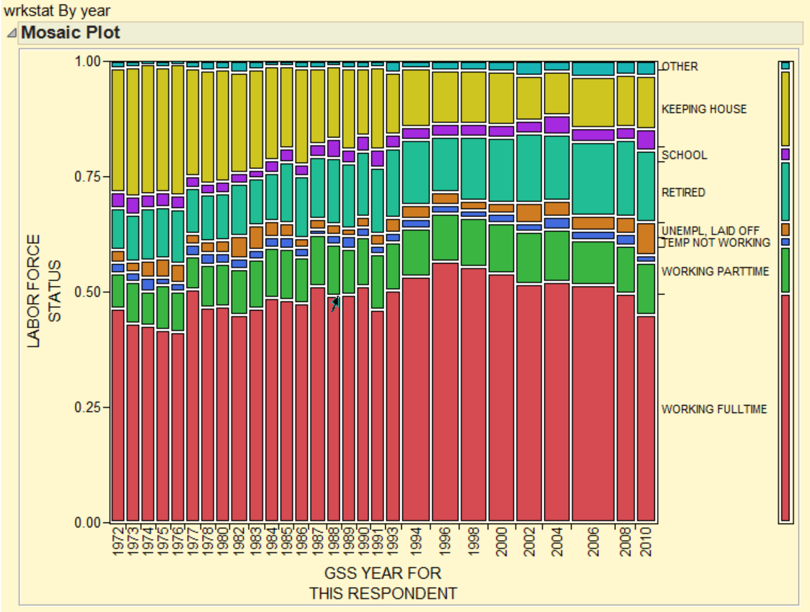
So was this all in “flow”? The Bivariate Fit takes 3 seconds, the kernel density 9 seconds. Not bad. Dragging a rectangle to select points was not instant, though — taking 15 seconds to catch up to the mouse. Fitting the 65-parameter model on 123 million rows took 107 seconds, a reasonable time for this size problem.

THE ‘WIDE’ DATA IS A SOCIAL SURVEY

The *wide* table is the General Social Survey from NORC, only 55,000 rows, but 5,416 columns. You can download it from http://www3.norc.org/GSS+Website/ I picked “Quick Downloads” and the SPSS format. The SPSS format is very nice because it has all the value labels and missing value codes already prepared, saving a huge amount of work. You have a choice of using either the short names or the long names when you import, but actually you can change your mind later, after you import using add-in menu items “Use Long Names” and “Use Short Names.” I always recommend long names, though sometimes mass-edit them when there is a systematic nuisance phrase in them.

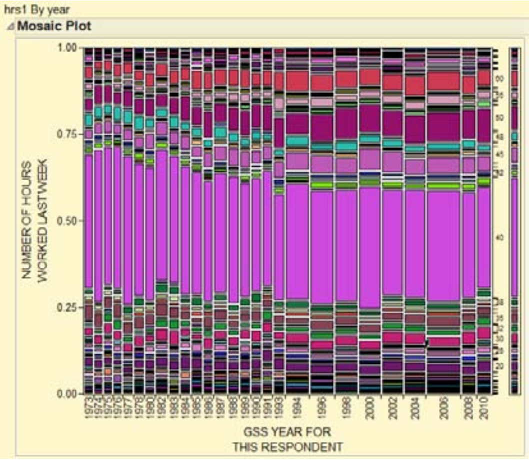
ANALYZE THE ‘WIDE’ DATA

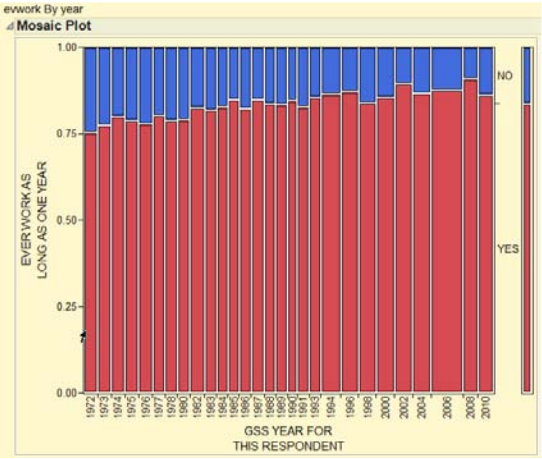
After working with the *tall* data table, we switch to the *wide* data table, which has only 55,087 rows — everything is much more instant again. This is survey data, and we are interested in which questions changed a lot over the 28 time periods of this survey. So we start with the first question, LABOR FORCE STATUS, with the labels IAP, WORKING FULLTIME, WORKING PARTTIME, TEMP NOT WORKING, UNEMPL, LAID OFF, RETIRED, SCHOOL, KEEPING HOUSE, OTHER, NA. We run the Fit Y by X Contingency platform with the following results:

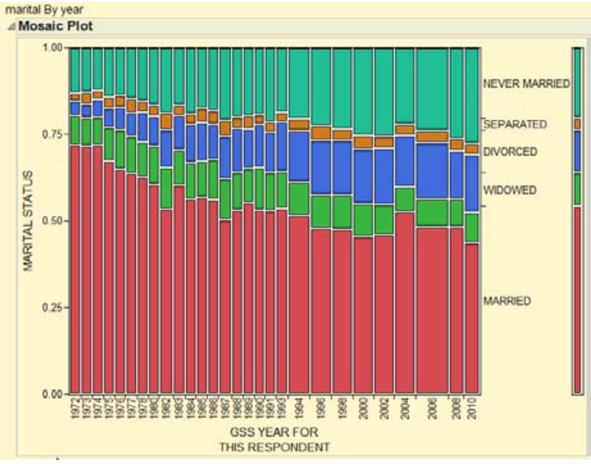


There are changes of interest — full time work peaked in 1996, and the most noticeable change was KEEPING HOUSE, which declined steadily until leveling off in the mid-1990s.

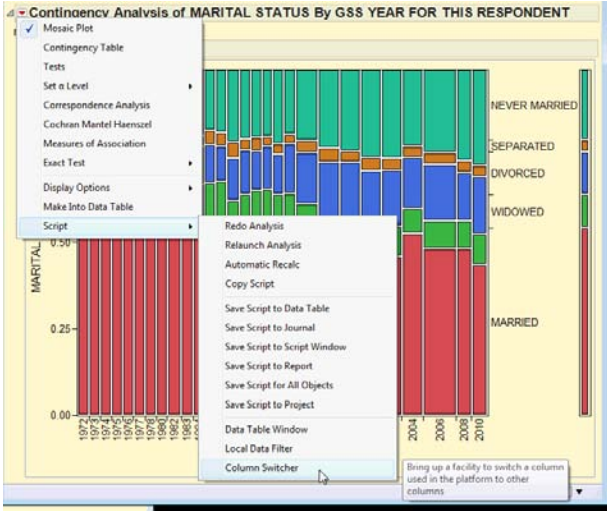
We move on to the next question, which is NUMBER OF HOURS WORKED LAST WEEK. Rather than relaunch the Contingency platform, we can just drag the variable name from the data table columns panel, to the Y-axis, which highlights as a landing zone when you start the drag into that window. In this case, there are lots of categories, but it is dominated by the 40-hour workweek. Next is EVER WORK AS LONG AS ONE YEAR, which rises. We do a few more.





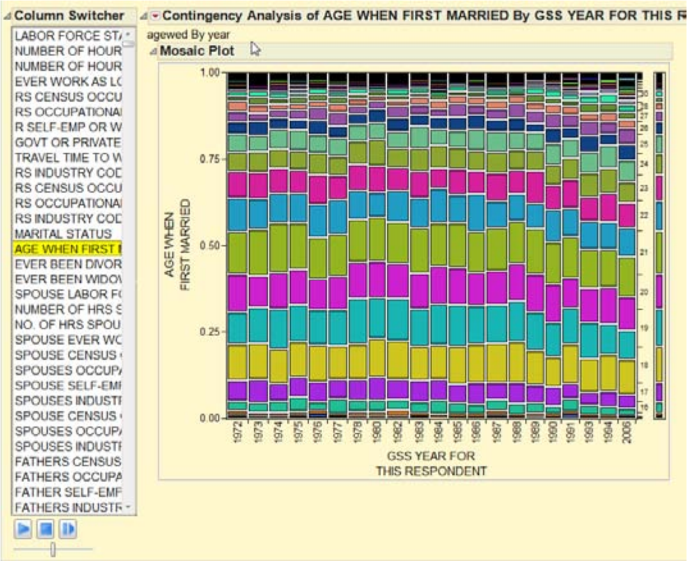


We have done four and have only 5,410 more questions to go. We are going to have to speed this up.

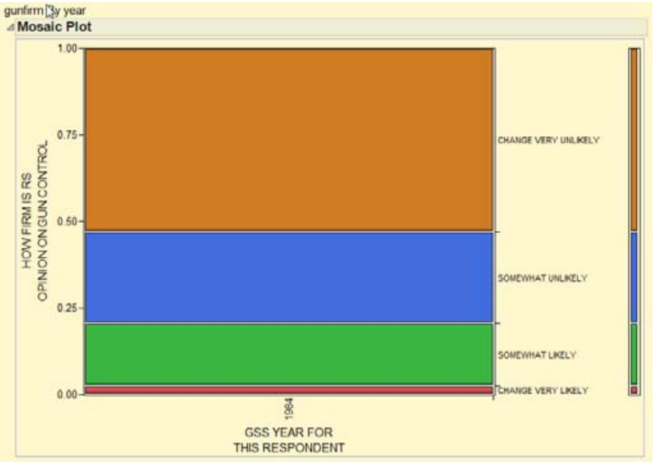


Bring up a facility to switch a column used in the platform to other columns

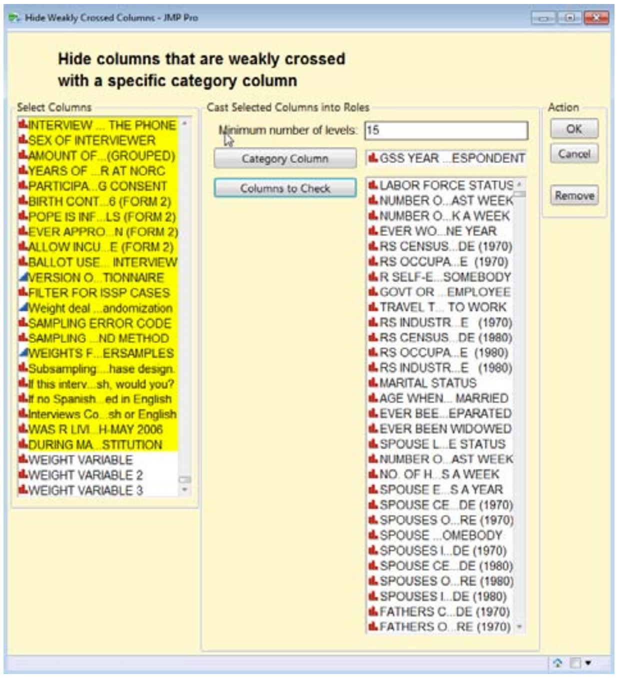
There is a new command in JMP 10 called “Column Switcher” in most platforms that allows you to rerun the platform, switching one column to another in a list of columns. Doing that for the Y-axis column gives you a way to easily switch just by clicking a column, or even just pressing the arrow key to go to the next. You can even have it sequence automatically by clicking the “play” button. Now we can speedily skim through, looking for interesting questions that change their response pattern over the years.



As you skim through the columns, you will see many of them that only ask a particular question for one or two years. These are obviously of no use to you in finding interesting questions that changed over many years.



We can use an add-in called “Exclude weakly-crossed columns” that is precisely designed for this situation, allowing you to exclude all the columns that don’t have enough levels in a category column (Year, in this case). So we ask for 15-category columns.



Hide Weakly Crossed Columns - JMP Pro

Hide columns that are weakly crossed with a specific category column

Select Columns

INTERVIEW ... THE PHONE
SEX OF INTERVIEWER
AMOUNT OF...(GROUPED)
YEARS OF ...R AT NORC
PARTICIPA...G CONSENT
BIRTH CONT...6 (FORM 2)
POPE IS INF...LS (FORM 2)
EVER APPRO...N (FORM 2)
ALLOW INCU...E (FORM 2)
BALLOT USE... INTERVIEW
VERSION O...TIONNAIRE
FILTER FOR ISSP CASES
Weight deal ...andomization
SAMPLING ERROR CODE
SAMPLING ...ND METHOD
WEIGHTS F...ERSAMPLES
Subsampling...hase design
If this interv...sh, would you?
If no Spanish...ed in English
Interviews Co...sh or English
WAS R LIVI...H-MAY 2006
DURING MA...STITUTION
WEIGHT VARIABLE
WEIGHT VARIABLE 2
WEIGHT VARIABLE 3

Cast Selected Columns into Roles

Minimum number of levels: 15

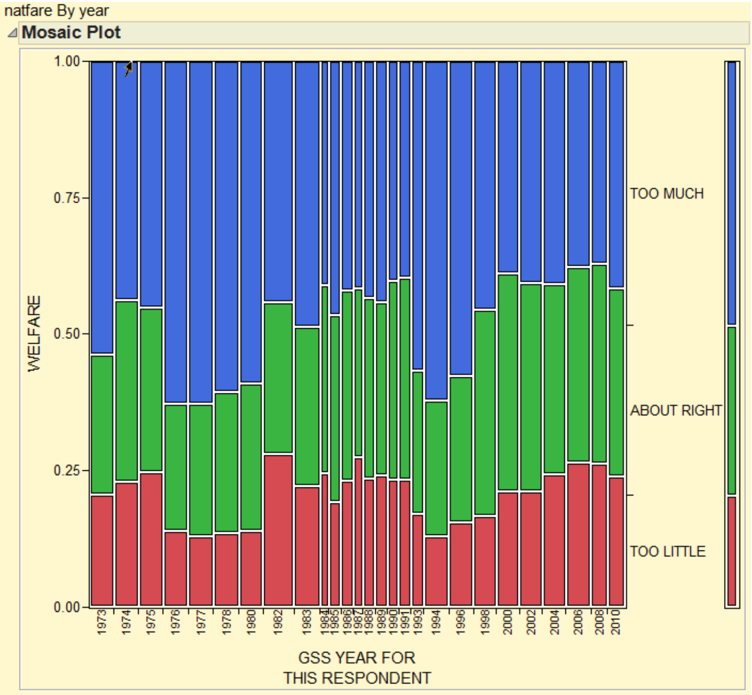
Category Column: GSS YEAR ...ESPONDENT

Columns to Check:

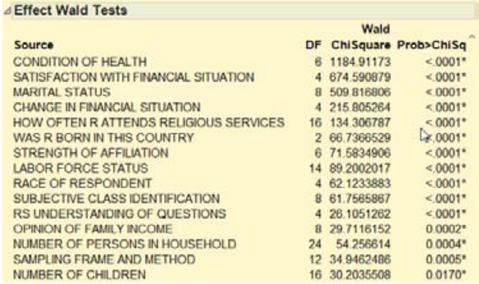
LABOR FORCE STATUS
NUMBER O...AST WEEK
NUMBER O...K A WEEK
EVER WO...NE YEAR
RS CENSUS...DE (1970)
RS OCCUPA...E (1970)
R SELF-E...SOMEBODY
GOVT OR ...EMPLOYEE
TRAVEL T... TO WORK
RS INDUSTR...E (1970)
RS CENSUS...DE (1980)
RS OCCUPA...E (1980)
RS INDUSTR...E (1980)
MARITAL STATUS
AGE WHEN... MARRIED
EVER BEE...EPARATED
EVER BEEN WIDOWED
SPOUSE L...E STATUS
NUMBER O...AST WEEK
NO. OF H...S A WEEK
SPOUSE E...S A YEAR
SPOUSE CE...DE (1970)
SPOUSES O...RE (1970)
SPOUSE ...OMEBODY
SPOUSES I...DE (1970)
SPOUSE CE...DE (1980)
SPOUSES O...RE (1980)
SPOUSES I...DE (1980)
FATHERS C...DE (1970)
FATHERS O...RE (1970)

Now we can recreate the Contingency Platform, add the Column Switcher and hold down the arrow key. The time to skim through all the 15-year questions among the 5,416 columns is only 32 seconds.

Of course, we do stop at the interesting cases. For example, the opinion on whether there is too much welfare spending seems to be related to whether there is a Republican or Democrat in office, not to the actual level of welfare spending.



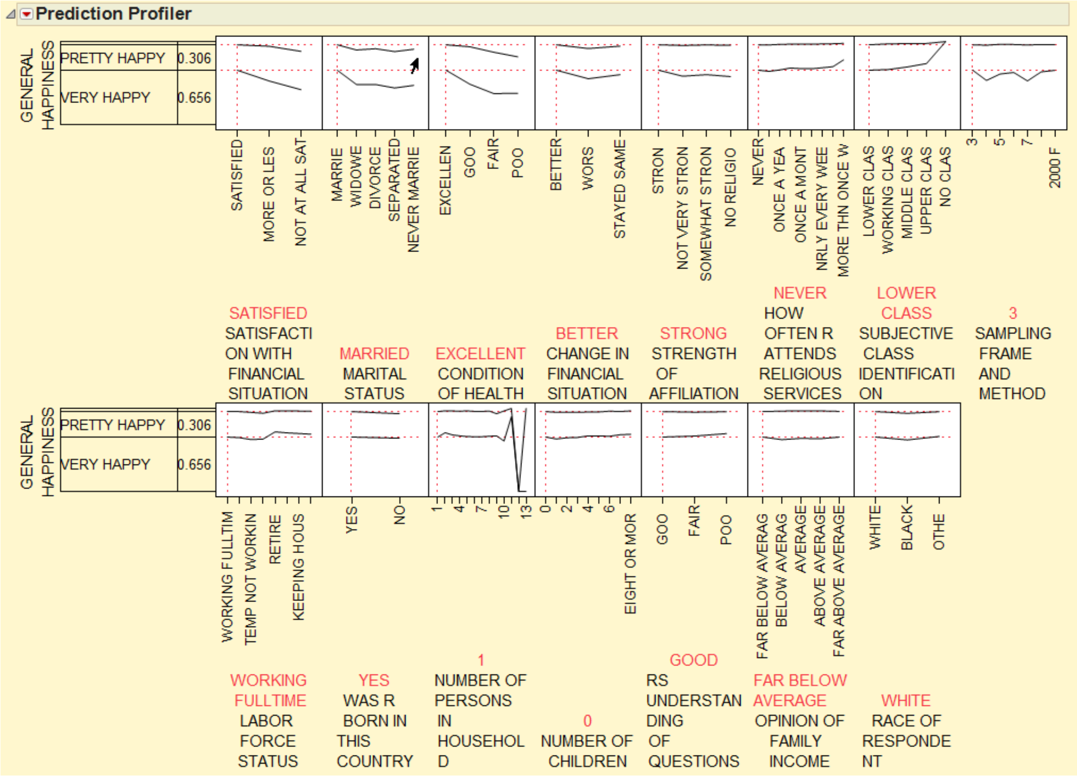
There is one column that I want to pursue further: GENERAL HAPPINESS. It might be interesting to see which other columns predict happiness. We can't use all 5,416 other columns. We don't want values that don't cover many years, so we use the add-in to exclude those. We don't want values that have a large number of missing values, so we use an add-in to exclude those. We still have too many. So we use a data mining tree model to make a large predictor, and then we look at column contributions to the fit and use the stronger columns to then make a logistic regression model (Williams College professor Dick De Veaux calls this approach "shaking the tree"). Here we see what seems to matter most.



Effect Wald Tests

Source	DF	Wald ChiSquare	Prob>ChiSq
CONDITION OF HEALTH	6	1184.91173	<.0001*
SATISFACTION WITH FINANCIAL SITUATION	4	674.590879	<.0001*
MARITAL STATUS	8	509.816806	<.0001*
CHANGE IN FINANCIAL SITUATION	4	215.805264	<.0001*
HOW OFTEN R ATTENDS RELIGIOUS SERVICES	16	134.306787	<.0001*
WAS R BORN IN THIS COUNTRY	2	66.7366529	<.0001*
STRENGTH OF AFFILIATION	6	71.5834906	<.0001*
LABOR FORCE STATUS	14	89.2002017	<.0001*
RACE OF RESPONDENT	4	62.1233883	<.0001*
SUBJECTIVE CLASS IDENTIFICATION	8	61.7565867	<.0001*
RS UNDERSTANDING OF QUESTIONS	4	26.1051262	<.0001*
OPINION OF FAMILY INCOME	8	29.7116152	0.0002*
NUMBER OF PERSONS IN HOUSEHOLD	24	54.256614	0.0004*
SAMPLING FRAME AND METHOD	12	34.9462486	0.0005*
NUMBER OF CHILDREN	16	30.2035508	0.0170*

The Profiler shows which levels associate with happiness. Apparently, having 12 persons in a household makes you very unhappy. Having good health and good financial situation are associated with being happy.



CONCLUSION: FAST IS GOOD

Being able to analyze big data with a fast computer and fast software is a very good thing. It keeps you focused on the data and your models of it, rather than drifting off while you wait for the results. It encourages highly interactive exploration to pursue clues, make discoveries.

However, some problems are too big for a single computer. Here is when you need to switch to the SAS® high-performance products. SAS has made a huge investment in supporting massively parallel compute appliances, distributing across many nodes, each node with many gigabytes of main memory to hold the data. How long does it take to calculate a logistic regression with a billion rows? Much less than a minute.

We have entered the Era of Big Data. We have big data tables. When they are not too big, we can analyze them interactively and graphically with JMP®. When they are bigger, we can still analyze them rapidly, using SAS.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

John Sall
SAS Institute
SAS Campus Drive
Cary, NC 27513 USA
Web: jmp.com and sas.com

APPENDIX

Script from data open preview table property “Source” edited slightly:

```
dir = "path to CDC Birth Files";
Open(
        dir||"Nat2008us.dat",
        columns(
                xxx=omit, year=Numeric, month=Numeric,
                xxx=omit, age=Numeric,
                xxx=omit,Mother Age=Numeric,
                xxx=omit,Mother Race=Numeric,
                xxx=omit,Marital=Numeric,
                xxx=omit,Educ=Numeric,
                xxx=omit,Father Age=Numeric,
                xxx=omit,Father Race=Numeric,
                xxx=omit,Birth Order=Numeric,
                xxx=omit,Delivery Method=Numeric,
                xxx=omit,Apgar Code=Numeric,
                xxx=omit,Sex Code=Numeric,
                xxx=omit,Downs=Numeric,
                xxx=omit),
   Import Settings(
    Fixed Column Widths(
                14,4,2,// 21
                68,2, // 91
                52,1, //144
                8,1,//153
                5,1,//159
                17,2,//188
                3,1, //192
                25,1, //218
                184,1, //403
                12,1, //416
                20,1, //437
                274,1, //712
                63,0       //775
   ),
        Strip Quotes(0), Use Apostrophe as Quotation Mark(0),
        Scan Whole File(1), Treat empty columns as numeric(0),
        Labels(0), Column Names Start(1), Data Starts(1),
        Lines To Read(All), Year Rule("10-90")
  )
);
```