Paper 173-30

A System to Calculate Market Value-at-Risk using Monte Carlo Simulation II: Reporting

SUGI George Rezek, GMAC Enterprise Risk Management, Minneapolis, MN

ABSTRACT

The non-additive nature of VaR results presents special problems in reporting. This paper describes the mechanics of VaR calculation and the maintenance of elemental correlations across all levels of aggregation, such as enterprise-wide, business division and instrument type. Examples of graphs used in reporting are also given.

INTRODUCTION

Three years ago upper management at GMAC Mortgage Group decided that it was important to establish policies and programs designed to monitor the market, credit, and operational risks the three companies of the Group faced. My NESUG paper last year gave an overall view of the VaR model used at GMAC and aspects of the system. This paper focuses on the reporting of the results.

WHAT IS VALUE-AT-RISK?

Value-at-Risk is the maximum amount of money that may be lost on a portfolio on average over a given period of time, with a given confidence interval *under normal circumstances*. For traders this period of time, or holding period, can be a day or less. For our model the holding period is one month. When we say that our VaR is $47m for a given company at the 95th confidence interval, we mean that, under normal markets conditions, we expect to lose more than $47m, 5% of the time. Note that this does not consider catastrophic events.

VALUE-AT-RISK at GMAC

While there are various ways of calculating Value-at-Risk, we use a two factor, interest rate and spread, correlation model. The help explain the mechanics of the model, I've illustrated an example involving three risk factors, three products, running 10 simulations in Appendix A.

VAR CALCULATION

As the elements of SAS/IML® matrices are either all number or all character, we have to prepare the data in base SAS such that we can keep track of the variable names, etc. The names of the organization levels have to be converted into numbers. If this is a static list you can use proc format and use the number assignment to force the order of the reports or read the data into data step as in the example below:

```
data desktemp1 ;
  set sasdb.desk(keep=desk desk_sort_order process_group) ;
  where process_group = 'MVAR' ;
proc sort data = desktemp1 ;    by desk ;
data desktemp2(drop=process_group) ;
  set desktemp1 ;  by desk ;
  if last.desk ;
proc sort data = desktemp2 ;
  by desk_sort_order ;
data desk(rename=(desk=start) drop=process_group) ;
  retain fmtname '$deskfmt' ;
  set desktemp2 ;
  by desk notsorted ;
  if first.desk then do ; label + 1; output ;  end ;
proc format cntlin = desk fmtlib ;
```

The advantage of the having the data create the format is that the code doesn't have to be adjusted with each change in the database. Next we denormalized various database tables such that one matrix (mark_sql) holds risk factor and product id's and profiles, any information we'll need in to alter VaR values (currency ratio, liquidity adjustment), plus information we'll need for aggregation (business unit, asset class, etc.).

We want the report to appear as a tree structure, aggregating VaR results in a total, then at the company group level (GMAC, GM), then see each of these branch off into the various company in the groups, etc. We construct our reporting structure by merging a dataset with all the (numerical) permutations of the organization (business_group, company, etc.) with the desks in the mark_sql dataset, adding those numerical representations to create mark_val.

```
data primenum ;
  set sasdb.desk ;
  where process_group = 'MVAR' ;
      groupnum = input(put(company_group,$grpfmt.),best32.) ;
      conum = input(put(company,$compfmt.),best32.) ;
      busnum = input(put(business,$busfmt.),best32.) ;
      busunit = input(put(business_unit,$bnitfmt.),best32.) ;
      desknum = input(put(desk,$deskfmt.),best32.) ;
* proc print data = primenum ; * title 'primenum' ;
proc sort data = primenum out = prime_match ;
  by desk_id ;
data prime_mat(keep=desk_id groupnum conum busnum busunit desknum) ;
  set prime_match ;
  by desk_id ;
  if last.desk_id ;
proc sort data = sasdb.mark_sql ;
  by desk_id ;
data mark_val ;
  merge sasdb.mark_sql(in=a)
        prime_mat(in=b) ;
  by desk_id ;
  if a ;
```

We create a macro variable of product id's to use as a label for product level profit and loss matrix.

```
proc sort data = mark_val ;
  by product_id ;
data _null_ ;
%global prodcols ;
proc sql ;
select compress( tranwrd(tranwrd(tranwrd(tranwrd(upcase(tranwrd(tranwrd
       ( translate(left(trim(p.product_name)),'________','()&/-+ ')
          || '_' ||
         translate(left(trim(p.product_term)),'________','()&/-+ ')
          || '_' ||
         translate(left(trim(p.product_rating)),'________','()&/-+ ')
         || '_' ||
       put(p.product_id,z3.0),'__','_' ),'__','_' )),'_1YR_1YR','_1YR'),
         '_3YR_3YR','_3YR'),
         'RESIDUAL_FINANCING','RES_FIN'),
         'FIXED_RATE_LOANS','FIXRATLNS'))
  into :prodcols separated by ','
  from mark_val mv
      ,sasdb.product p
  where p.product_id = mv.product_id ;
quit ;
```

To flatten out the tree reporting structure into something we could use in a spread sheet we remove duplicates from five files, each representing a level of the organization, retaining the numbers that will appear in the mark_val matrix and concatonating the names for the report

```
data primerow0 ;
  length primecut $ 200 ;
  set primenum(keep=groupnum company_group desk_id) ;
  primecut = left(trim(company_group)) ;

data primerow1 ;
  length primecut $ 200 ;
  set primenum(keep=groupnum company_group conum company desk_id) ;
  primecut =  left(trim(company_group)) || '_' || left(trim(company)) ;
     ...
```

And merge the results:

```
data primerow4 ;
  length primecut $ 200 ;
  set primenum(keep=groupnum company_group conum company busnum business busunit business_unit
                      desknum desk desk_id) ;
  primecut = left(trim(company_group)) || '_' || left(trim(company))  || '_' ||
             left(trim(business)) || '_' || left(trim(business_unit)) || '_' || left(trim(desk)) ;

data sasdb.primerow ;
  length primecut $ 200 ;
  set primerowt ;
  by groupnum conum busnum busunit desknum ;
  if last.desknum ;

proc contents data = sasdb.primerow out = list(keep=nobs) noprint ;
data _null_ ;
  %global coln ;
  set list ;
  nobs = nobs + 1 ;
 call symput('coln','col' || left(trim(nobs))) ;  stop;
```

The interpolation module was originally written using nested loops. The interpolation process is the most time intensive of the entire program; substituting matrix notation is some parts of the module reduced the time of the program by one third. Not only is matrix notation elegant and easy to follow, it often proves to be more efficient, but it consumes more memory.

We read the file that contains the primary business categories into a matrix (primerow), dimension the matrix that will hold the totaled profit and loss aggregations by these categories:

```
use sasdb.primerow ;
read all var{groupnum conum busnum busunit desknum}  into primenum ;
GMAC = j(&sims,nrow(primenum)+1,0) ;
GMAC_IR = j(&sims,nrow(primenum)+1,0) ;
GMAC_SP = j(&sims,nrow(primenum)+1,0) ;
```

The liquidity adjustment is calculated by dividing the sum of the base values of the products times the liquidity factor by the sum of the base values within each primary business category.

```
* group offset ;

    do j = 1 to nrow(primenum) ;
       if (primenum[j,1] = mark_val[i,22]
         & primenum[j,2] = .
         & primenum[j,3] = .
         & primenum[j,4] = .
         & primenum[j,5] = .)
       then do ;
         GMAC[,j+1] = GMAC[,j+1] + PL[,i] ;
         GMAC_IR[,j+1] = GMAC_IR[,j+1] + PL_IR[,i] ;
         GMAC_SP[,j+1] = GMAC_SP[,j+1] + PL_SP[,i]
       end ;
    end ;

* company offset ;
    do j = 1 to nrow(primenum) ;
       if (primenum[j,1] = mark_val[i,22]
         & primenum[j,2] = mark_val[i,23]
         & primenum[j,3] = .
         & primenum[j,4] = .
         & primenum[j,5] = .)
       then do ;
         GMAC[,j+1] = GMAC[,j+1] + PL[,i] ;
         GMAC_IR[,j+1] = GMAC_IR[,j+1] + PL_IR[,i] ;
         GMAC_SP[,j+1] = GMAC_SP[,j+1] + PL_SP[,i] ;
       end ;
     end ;
```

The profit and loss numbers are aggregated to the relevant business category by comparing the individual business level numbers in mark_val with the numbers in primenum.

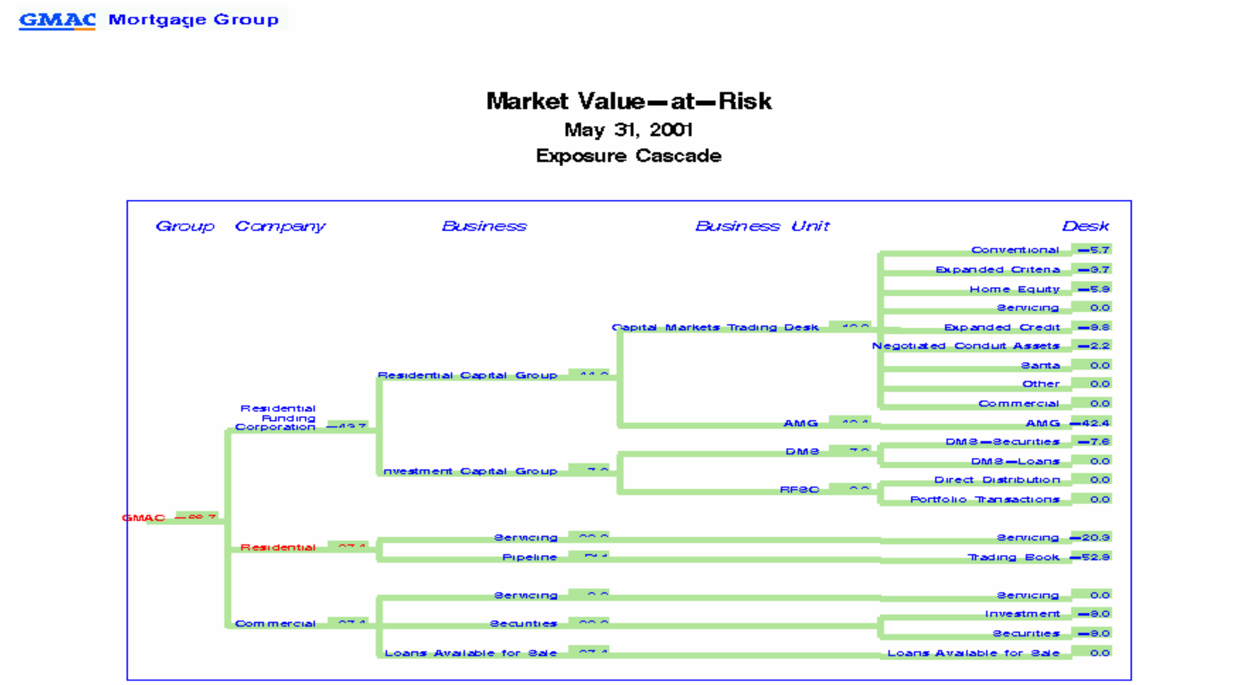
Separate row matrices can be made for each confidence level using an IML module. These matrices can adjusted for liquidity by dividing multiplying by the liquidity ratio:

```
fifha = fifth(GMAC) ;
gmac9995 = fifha[4,] ;
gmac999 = fifha[3,] ;
gmac990 = fifha[2,] ;
gmac950 = fifha[1,] ;
```

REPORTING OF VAR RESULTS

The process is repeated for each month specified and the results appended to SAS data sets. Typically we want a 12 month run to evaluate VaR over time. The results are read in IML, transposed and the relevant names added as row names. All numbers in the following charts and graphs are representative.

Reports were designed primarily for two different groups: upper management and analysts. Those for upper management were mostly overview and graphical, those for analysts were more detailed and largely on an ad Hoc basis. The following graph was one of the first to appear in the monthly reports. The VaR numbers appear in the cascade of business entities. The VaR numbers in red indicate a violation and in this theoretical case, the violation of one company clearly causes the violation of the group and it is precisely this emphasis that we intended. The elements of this graph were layed out by hand and given the frequency of changes and additions to the hierarchy it became tedious to maintain. Although we did not have SAS/OR at the time, Proc Netdraw in SAS/OR® can produce a similar chart.



The VaR numbers were distributed to analysts on spread sheet, on separate worksheets with different confidence intervals.

AGMAC_950	MAY2001
GMAC	130,929,802
GMAC/RFC	-60,866,649
GMAC/RFC-Residential Capital Group	-61,141,541
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk	-9,573,930
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Conventional	-4,210,821
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Expanded Criteria	-3,419,253
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Home Equity	-6,450,804
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Servicing	3,333,330
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Expanded Credit	-9,451,757
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Negotiated Conduit Assets	-3,346,201
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Santa	44,440
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Other	0
GMAC/RFC-Residential Capital Group-Capital Markets Trading Desk-Commercial	0
GMAC/RFC-Residential Capital Group-AMG	-75,463,134
GMAC/RFC-Residential Capital Group-AMG-Servicing	-40,550,169
GMAC/RFC-Residential Capital Group-AMG-IO Book	-70,303,098
GMAC/RFC-Residential Capital Group-AMG-Residual	-16,630,296
GMAC/RFC-Investment Capital Group	-3,001,432
GMAC/RFC-Investment Capital Group-DMS	-1,821,444
GMAC/RFC-Investment Capital Group-DMS-DMS - Securities	-1,735,534
GMAC/RFC-Investment Capital Group-DMS-DMS - Loans	-396,434
GMAC/RFC-Investment Capital Group-RFSC	-2,287,872

As the number of portfolios were constantly being added to the database, it was almost impossible to maintain comparable VaR numbers. Typically we ran 12-month at a time, so that we could see trends, keeping the definitions of VaR aggregation and methodology constant. The problem with these revisions is that it was impossible to reproduce previous VaR numbers. We created a file in Oracle in which we appended the current month's results in order to maintain a permanent history. Note that we have two dates: valuation_date, the date of the data and a processing date. This would permit storing multiple runs concurrently, so that you could save results from alternate methodologies for the same period.

Processing Date	Valuation Date	COMPANY_GROUP	COMPANY	BUSINESS	BUSINESS_UNIT	DESK	RESULT_TYPE	CI	VAR_RESULT
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	Total	Total	Total	95	-1513162.095
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	CORE	Total	Total	95	-894078.84
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	CORE	CORE	Total	95	-894078.84
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	REG	Total	Total	95	-68816.0242
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	REG	REG	Total	95	-683786.0242
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	MHF	Total	Total	95	-1383986.5676
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	MHF	MHF	Total	95	-1383986.5676
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	HC	Total	Total	95	0
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	HC	HC	Total	95	0
6/30/2003	7/31/2002	GMAC Mortgage Group	RFC	BCG	RF	Total	Total	95	0

These results were limited, as there were requests for aggregation that cut across business units. While the model assumed every product as an asset, the products were divided into “Asset” and “Hedge” designations. In order to produce such aggregations I would have to hard code conditions in the final loop in IML. As this loop was an integral part of the model, the VaR model would be rerun, a lengthy process and error prone. While the prospect of maintaining the simulated profit and loss matrices (one for interest rate, one for spread) seemed impractical, it became more attractive over time. Running a query against the product table and P&L tables could produce VaR numbers for requests for all residual I/O or prepayment sensitive assets without interfering with the main program.

The var_results table, where monthly results of every business level was stored, is useful in reproducing previously reported VaR numbers, as this table is updated with current month results only. We retain P&L figures for the entire 12- or 24-month periods to do “what-if” scenarios on non-standard business levels or to evaluate special case situations. In fact, when two of the companies officially merged into one entity, management not only required that VaR numbers be produced for both companies separately and combined, these numbers became one of the most anticipated. In this case there was no alternative but to create and retain the P&L table, as the database would only accept the companies be calculated either separately or together.

At his point it becomes obvious that an interactive screen to drive these requests is what the analysts would really want. Although each company already has the VaR code and generates there own results, most of the requests that are referred back to us concerns such queries.

The P&L results for each month are appended into two tables (one for interest rate and one for spread):

date_mon	PRODUCT_ID	PL_VALUE	SIM_NO	sims
5/31/2004	12	-131598.9235	1	15000
5/31/2004	12	584850.5111	2	15000
5/31/2004	12	-850804.3203	3	15000
5/31/2004	12	148108.1298	4	15000
5/31/2004	12	135896.8923	5	15000
5/31/2004	12	478860.5237	6	15000
5/31/2004	12	1889896.57	7	15000
…				

Another common use for maintaining the P&L matrices is to recalculate diversification benefit, when one set of assets is excluded from the enterprise total. For example, what would our VaR be, if we excluded one company? Developing queries from .html screens using SAS/IntrNet® proved less daunting than I’d originally imagined. The problem of lack of familiarity was greatly aided by adapting existing examples. Fortunately, what I needed was relatively simple: I need the screens to record the choices from pull-down menus and pass them to SAS code. The SAS code

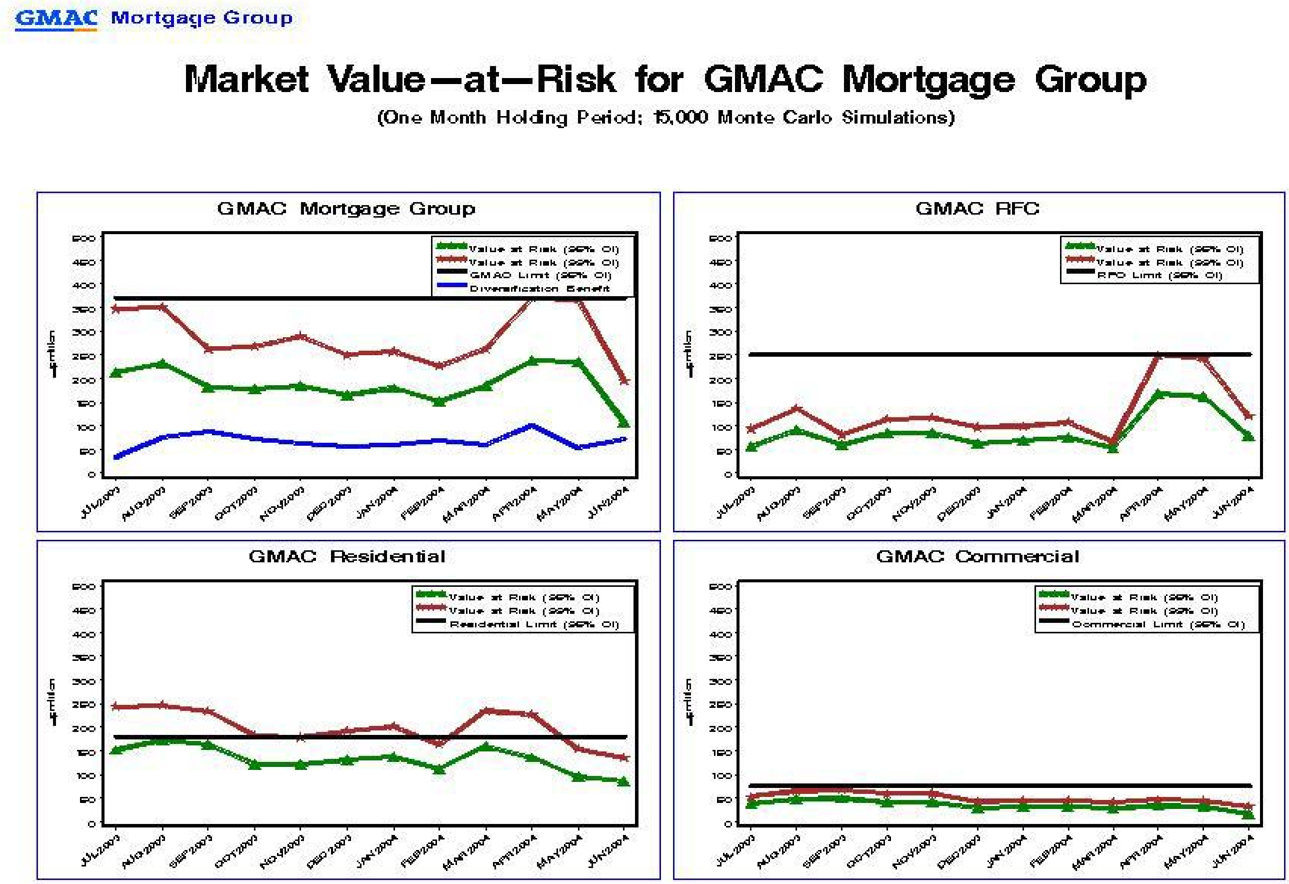
```
proc sort data=sasdbin.company(keep=company) nodupkey out=sort; by descending
company ;

    data _null_;
      set sort end=last;
      by descending company;
      file _webout;
      if _n_= 1 then do;
        put '<td align="left"><select name="company" size="3" multiple>';
      end;
      if first.company then put "<option>" company ;
      if last then do;
        put '</select></td>';
      end;
      run;
```

generates html code that creates a pull-down menu of the SAS dataset , company, showing three rows at a time allowing multiple rows to be selected. The selections are passed to a SAS query as macro variables, &company1, &company2,…plus &company0, which holds the number of rows selected. These macro variables can be inserted into a query to subset the P&L table.

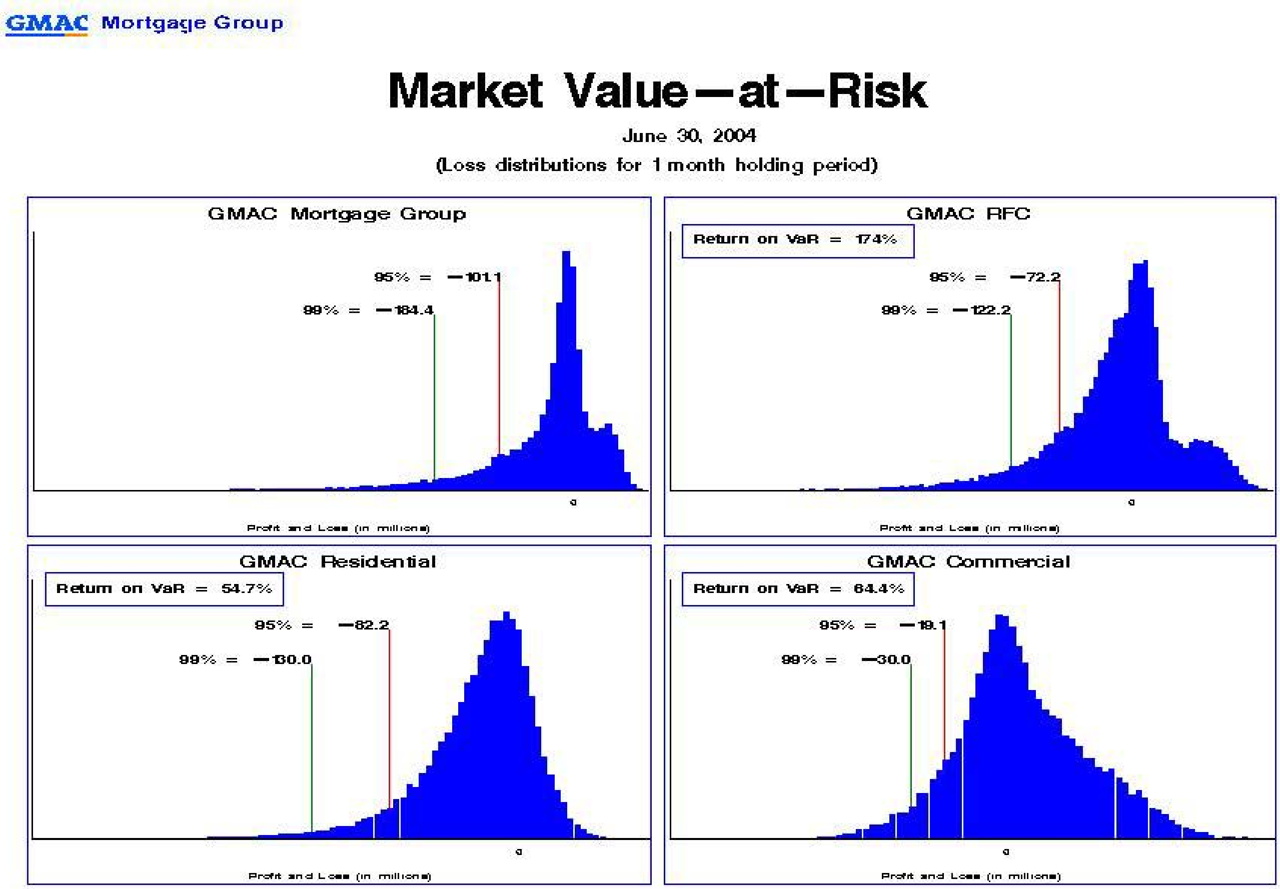
While time in producing a chart or a graph doesn't seem to have much importance, the time involved in producing results in an interactive system becomes extremely important. Storing the P&L matrix after each monthly run, reloading the appropriate monthly run and managing the process in IML saved time in an interactive session. This is perhaps because the process is running in memory. The caution here is that it is bound by the memory limits of the operating system, in our case Windows XP 32-bit.

GRAPHS

The two most important graphs were the 12-month history and current P&L distribution. Few people have an intuitive sense for what a VaR number represents, so the 12-month history shows month over month change. Also, the diversification benefit is included on this Mortgage Group graph (in blue).



The other heavily used graph is the graph of the distribution of the P&L. We assume a normal distribution and if we see a distribution that isn't normal, we want to take a closer look. Each month we run the graph of the distribution for the current period. What we need is to see the underlying distributions associated with each point on the 12-month history graph history graph.



CONCLUSION

The SAS system offers a full array of products to display statistical results that are accessible to the average SAS user.

REFERENCES

Philippe Jorion, *Value at Risk*, 2nd edition.

J.P. Morgan's *Risk Metrics - Technical Document* available on the internet at http://www.jpmorgan.com/RiskManagement/RiskMetrics/RiskMetrics.html

Rezek, G., (2003) "A System for Calculating Value-at-Risk using Oracle® and Monte Carlo Simulation in SAS/IML®. *Proceedings of the Sixteenth Annual Conference of the Northeastern SAS Users Group*

ACKNOWLEDGMENTS

Thanks to Michael Davis, Jonah Turner, Bernd Imken and the others at SUGI 29 who presented excellent papers at SUGI 29 on the subject of SAS/IntrNet, SAS/AF® and AppDevStudio™.

Special thanks to Brent Juusola of GMAC-RFC for his patient explanations of the SAS/IntrNet installation and walk through of current applications.

CONTACT INFORMATION

George Rezek
GMAC Enterprise Risk Management
8400 Normandale Lake Blvd
Bloomington, MN 55437
Work phone: (952) 857-6042
Fax: (952) 857-6688
email: george.rezek@gmacerm.com

Appendix A

GMAC Two-Factor Value-at-Risk Model

History of Risk Factors

Year-Month	Gov Treas 5yr	AA Loan	10yr Swap
2002M04	4.405	113.500	54.000
2002M03	4.838	124.300	66.500
2002M02	4.189	130.600	66.500
2002M01	4.369	136.900	72.000
2001M12	4.300	153.800	76.500
2001M11	4.062	148.700	69.650
2001M10	3.473	146.700	66.670
2001M09	3.802	146.700	66.670
2001M08	4.378	141.000	81.710
2001M07	4.525	141.600	80.160
2001M06	4.949	151.000	90.180
2001M05	4.949	141.300	77.710
2001M04	4.949	148.800	81.370
2001M03	4.949	174.900	93.140
2001M02	4.949	154.200	89.330
2001M01	4.949	143.400	79.390
2000M12	4.949	163.000	102.140
2000M11	4.949	171.100	118.100
2000M10	4.949	169.000	112.100
2000M09	4.949	164.000	111.450
2000M08	4.949	182.000	127.250
2000M07	4.949	175.000	122.500
2000M06	4.949	179.500	123.300
2000M05	4.949	193.000	135.900
2000M04	4.949	179.000	119.000
2000M03	4.949	175.000	125.750
2000M02	4.949	163.000	102.100
2000M01	4.949	140.000	85.550
1999M12	4.949	144.000	79.000
1999M11	4.949	143.000	79.000
1999M10	4.949	155.000	87.000
1999M09	4.949	163.000	97.000
1999M08	4.949	179.000	108.000
1999M07	4.949	167.000	100.000
1999M06	4.949	144.000	83.500
1999M05	4.949	138.000	79.500
1999M04	4.949	134.000	71.000
1999M03	4.949	137.000	79.000
1999M02	4.949	137.000	74.000
1999M01	4.949	145.000	75.000
1998M12	4.949	161.000	76.000
1998M11	4.949	155.000	80.500
1998M10	4.949	210.000	77.000
1998M09	4.949	165.000	84.500
1998M08	4.949	123.000	85.500
1998M07	4.949	98.000	55.500
1998M06	4.949	96.000	55.500
1998M05	4.949	90.000	53.500
1998M04	4.949	90.000	49.500
1998M03	4.949	95.000	51.500
1998M02	4.949	90.000	48.500
1998M01	4.949	90.000	48.500
1997M12	4.949	90.000	47.500
1997M11	4.949	90.000	45.500
1997M10	4.949	88.000	53.500
1997M09	4.949	70.000	44.500
1997M08	4.949	67.000	42.500
1997M07	4.949	67.000	40.500
1997M06	4.949	67.000	38.500
1997M05	4.949	70.000	35.500

Correlation of Risk Factors

Risk Factor	Gov Treas 5yr	AA Loan	10yr Swap
Gov Treas 5yr	1.000	-0.145	0.194
AA Loan	-0.145	1.000	0.572
10yr Swap	0.194	0.572	1.000

Covariance of Risk Factors

Risk Factor	Gov Treas 5yr	AA Loan	10yr Swap
Gov Treas 5yr	0.005	-0.136	0.130
AA Loan	-0.136	192.415	78.837
10yr Swap	0.130	78.837	98.713

Singular Value Decomposition

Simulated Risk Factor Values

RISK FACTOR	Gov Treas 5yr	COM AA	10yr Swap
Simulation 1	0.03	16.20	8.35
Simulation 2	-0.04	2.69	0.72
Simulation 3	0.07	26.82	6.43
Simulation 4	0.00	5.68	13.59
Simulation 5	-0.07	0.29	-4.02
Simulation 6	-0.08	-2.66	-4.21
Simulation 7	-0.04	-10.22	8.94
Simulation 8	0.08	-1.99	-1.51
Simulation 9	0.03	21.57	13.07
Simulation 10	0.01	-7.32	-3.21

Standardized Risk Factor Values

Risk Factor	Gov Treas 5yr	COM AA	10yr Swap
Simulation 1	0.54	1.17	0.84
Simulation 2	-0.53	0.19	0.07
Simulation 3	1.20	1.93	0.65
Simulation 4	0.11	0.41	1.37
Simulation 5	-0.98	0.02	-0.40
Simulation 6	-1.02	-0.19	-0.42
Simulation 7	-0.46	-0.74	0.90
Simulation 8	1.31	0.14	-0.15
Simulation 9	0.57	1.56	1.32
Simulation 10	0.20	-0.53	-0.32

Compute 's for Interest Rate and Spread

	Current Month	+1 σ	+2 σ	+3 σ
Gov Treas 5yr	4.405	29.674	59.348	89.023
COM AA	113.500	13.871	27.743	41.614
Swap 10yr	54.000	9.936	19.871	29.807

Interest Rate → Gov Treas 5yr; Spread → COM AA, Swap 10yr

Traders use volatility reports to predict market values of their portfolios corresponding to -3, -2, -1, 0, +1, +2 & +3 σ movement of risk factors

Interest Rate Shocks

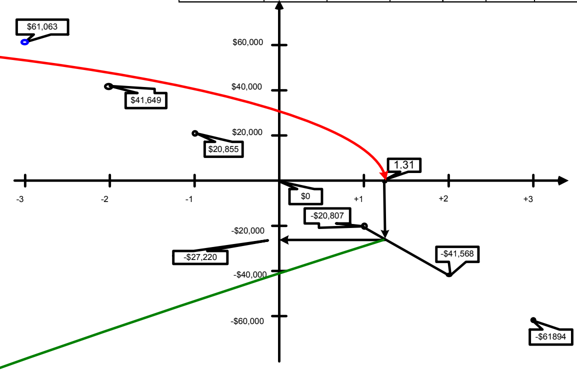
Product	-3σ	-2σ	-1σ	Base Value	+1σ	+2σ	+3σ
LAS FRL AA	61,063	41,649	20,855	0	-20,807	-41,568	-61,894
LAS ARML AA	46	31	16	0	-17	-34	-53
Sec Agency	-2,011,065	-1,340,709	-670,354	0	670,356	1,340,712	2,011,067

Spread Shocks

Product	-3σ	-2σ	-1σ	Base Value	+1σ	+2σ	+3σ
LAS FRL AA	61,980	41,376	20,711	0	-20,829	-41,668	-62,387
LAS ARML AA	-13,123	-8,744	-4,369	0	-11,381	-7,012	-2,649
Sec Agency	-858,826	-572,550	-286,275	0	286,276	572,552	858,827

Example of Straight-Line Interpolation

Product	-3 σ	-2 σ	-1σ	Base Value	+1σ	+2 σ	+3 σ
LAS FRL AA	61,063	41,649	20,855	0	-20,807	-41,568	-61,894



Interest Rate	LAS FRL AA	LAS ARML AA	Sec Agency
Simulation 1	-11,182	-9	360,270
Simulation 2	11,012	8	-353,964
Simulation 3	-25,026	-20	806,582
Simulation 4	-2,251	-2	72,531
Simulation 5	20,341	16	-653,818
Simulation 6	21,287	16	-684,284
Simulation 7	9,508	7	-305,632
Simulation 8	-27,220	-22	877,428
Simulation 9	-11,866	-10	382,307
Simulation 10	-4,201	-3	135,341

+

Spread	LAS FRL AA	LAS ARML AA	Sec Agency
Simulation 1	-24,323	-10,648	240,527
Simulation 2	-4,041	-2,208	20,793
Simulation 3	-40,280	-7,303	185,304
Simulation 4	-8,530	-4,661	391,543
Simulation 5	-430	-235	-115,826
Simulation 6	3,966	-837	-121,256
Simulation 7	15,257	-3,219	257,699
Simulation 8	2,964	-625	-43,636
Simulation 9	-32,400	-8,955	376,694
Simulation 10	10,931	-2,306	-92,622

=

Interest Rate & Spread	LAS FRL AA	LAS ARML AA	Sec Agency
Simulation 1	-35,505	-10,657	600,797
Simulation 2	6,971	-2,199	-333,170
Simulation 3	-65,306	-7,324	991,886
Simulation 4	-10,781	-4,662	464,074
Simulation 5	19,911	-219	-769,644
Simulation 6	25,253	-820	-805,539
Simulation 7	24,765	-3,211	-47,932
Simulation 8	-24,256	-647	833,791
Simulation 9	-44,266	-8,965	759,002
Simulation 10	6,730	-2,309	42,719

Sum the simulations across all products

Interest Rate & Spread	LAS FRL AA		LAS ARML AA		Sec Agency		Sum of All Products
Simulation 1	-35,505	+	-10,657	+	600,797	=	554,634
Simulation 2	6,971	+	-2,199	+	-333,170	=	-328,398
Simulation 3	-65,306	+	-7,324	+	991,886	=	919,257
Simulation 4	-10,781	+	-4,662	+	464,074	=	448,630
Simulation 5	19,911	+	-219	+	-769,644	=	-749,952
Simulation 6	25,253	+	-820	+	-805,539	=	-781,106
Simulation 7	24,765	+	-3,211	+	-47,932	=	-26,378
Simulation 8	-24,256	+	-647	+	833,791	=	808,888
Simulation 9	-44,266	+	-8,965	+	759,002	=	705,771
Simulation 10	6,730	+	-2,309	+	42,719	=	47,140

Sort Values

Sum of All Products (Sorted)
919,257
808,888
705,771
554,634
448,630
47,140
-26,378
-328,398
-749,952
-781,106

Determine VaR at 95% Confidence Interval

George Rezek
george.rezek@gmacerm.com