A QUADRATIC PROGRAM FOR DETERMINING EFFICIENT FRONTIER PORTFOLIO COMPOSITIONS USING THE SAS LANGUAGE

ALEX ANCKONI III, THE GEORGE WASHINGTON UNIVERSITY

PURPOSE

Since the appearance of the seminal works of Markowitz and Mark and Newbold, methods for computing the composition of efficient frontier portfolios have been described by a number of authors including Martin and Francis and Archer. The purpose of this paper is to present two versions of a quadratic program for determining the compositions of efficient frontier portfolios written in the SAS language which is widely available at computing facilities in the United States. The two versions of the program which are presented cover the case where assets can be held in long and short positions without constraint and the case where assets can only be held in long positions (only positive asset weights). Either program can be operated using the SAS correlation procedure to calculate the correlation matrix of asset total yields or using a variance-covariance matrix and a yield vector (where these are determined from either ex-post or ex-ante considerations). A number of examples are presented as are a number of suggestions for research use of these programs. The next section of this paper describes the basic structure of these two programs. This is followed by a section demonstrating basic use of the two final section of this paper will briefly indicate how these programs might be employed in research work of current interest to financial economists. A major goal of this paper is to make an easy-to-use program more accessible to students, research scholars and to practicing financial managers.

THE PROGRAMS

The SAS computer language has normally been used for the statistical analysis of data through application of user-friendly subroutines (called Procedures in the SAS documentation) which cover many familiar analysis. For those desired analyses for which SAS pre-programmed procedures are not available, the SAS software system language provides two facilities which simplify the preparation of custom programs with minimal coding requirements. The first of these is the capability to write code directly in a matrix format with a powerful set of supporting matrix operations such as matrix multiplication, matrix transposition, matrix inversion, matrix row/column deletion, matrix concatenation and various element by element matrix operations. This facility is well documented as PROC MATRIX in the SAS Statistics manual cited above. The heart of the programs written using the SAS matrix facility with input to this facility and output from this facility provided conveniently by other integrated SAS procedures.

The second SAS facility of importance to this paper is a macro facility which allows repetitive computation of a series of SAS data and/or computational steps. This macro facility will be discussed further in the section of this paper dealing with extended application of these programs.

For purposes of this paper, program one will be used to indicate the unconstrained program and program two will be used to indicate the program for which only positive asset weights are allowed (constrained to long positions only). Both programs use the same input sections as will be discussed below. There are two input sections which may be used interchangeably with either of the two computation sections depending upon the nature of the input data as will be discussed below. The two computation sections are unique due to the more complex computations associated with the solution of the constrained quadratic programming problem. The two programs use the same output sections as will also be discussed below.

Input Section

The purpose of the input section of the program is to take information provided and deliver it to the computation section as a variance-covariance matrix of asset total yields and a vector of mean asset yields.

If the input data are a set of total yield time series, these data are used to form the input data set as shown in the input section of program one (see below program). Then, the variance-covariance matrix of these total yields and the vector of mean asset yields are calculated using the SAS correlation procedure.

If the input data are a historic time series of asset prices and dividends, these are first converted to a total yield data set using SAS data step computations prior to computation of the variance-covariance matrix of the total yields and the vector of mean asset yields using the SAS correlation procedure.

The correlation procedure used in this first form of the input section prints the variance-covariance matrix of the asset yields, a yield correlation matrix and other simple statistics.

If the input data are a variance-covariance matrix and a mean yield vector (perhaps separately determined from historic information or perhaps determined from separate ex-ante analysis), the normal input section of the program can be shortened to the simple specification of these data in the input section as shown in the input section of sample program number two shown below.
Finally, both versions of the input section print the variance-covariance matrix ($\Sigma \Omega$) and the asset yield vector (ER) for verification purposes.

Computation Section

The computation sections of these two programs differ in that program one is a straightforward unconstrained solution for the asset weights which minimize the variance of a portfolio having a given yield. To describe the efficient frontier, the yield value is stepped over a number of values which cover the region from the yield of the asset having the minimum average yield to a yield value which exceeds the yield by an amount equal to two times the difference between the lowest and the highest individual asset mean yields. The "two times" parameter can be adjusted by modifying the value of the variable MUL in the fourth line of executable code in the program one computation section (see program below). The comment section at the beginning of program one contains additional suggestions regarding the setting of the MUL parameter.

In program two, the computation section solves for the asset weights as described above except the positive asset weight constraint is introduced using the normal method of iteratively deleting from the Jacobian matrix the row and column corresponding to negative asset weights. To describe the efficient frontier, the yield value is stepped over ten values which are centered in the region covering ninety percent of the range between the minimum and the maximum individual asset mean yields (since the yields and their standard deviation of the yields of all assets are available as output from the correlation procedure output or directly from the input data when matrix/vector input is used, twelve data points on this efficient frontier are available to the analyst for this case).

The results of either computation section is a description of each of the efficient frontier portfolios where each description is a row of output containing the weight of each asset, the portfolio yield and the standard deviation of the portfolio yield.

Output Section

Finally, the output section (identical for both programs) writes a table of the solution portfolio descriptions and then provides two plots of the solutions in yield-standard deviation space. One plot includes zero-zero reference coordinates, the other is expanded to show as large a plot of the solutions in yield-standard deviation space as possible. This section uses normal SAS printing and plotting procedures.

PROGRAM APPLICATIONS

This section provides brief background for the one application of each of the two subject programs shown in the programs below.

The application of program one shown below was one portion of an exploratory study of the optimal asset composition of a portfolio of United States dollar denominated money market instruments. The input data were the monthly average of daily yields of ninety-day U.S. Treasury bills. Since the Bulletin quotes commercial paper, banker's acceptances, certificates of deposit, Euro-certificates of deposit and Treasury bills. Since the Bulletin quotes commercial paper, banker's acceptances and Treasury bills in an annual banker's discount rate format, three computation lines were added to the data step in the input section of the program to provide true annual percentage interest rates for these assets. The output of this program shows the very high correlation of yields that is typical of assets in the subject market. The output table demonstrates that the efficient frontier, in this unconstrained case, is traced by portfolios containing short positions in commercial paper and banker's acceptances and containing long positions in the other three assets. Further discussion of these results are contained in the cited paper.

The application of program two shown below was one element of a larger research effort seeking to develop methods for estimating efficient portfolio composition from factor time series determined through use of the Arbitrage Pricing Theory. The results of this research analysis, which formed the input data for program two, were an estimated asset yield variance-covariance matrix ($\Sigma \Omega$) and an estimated asset yield vector (ER). The basic data in the cited work were the monthly yield of 30 day Eurocurrency deposits denominated in US dollars, Swiss francs, W. German marks, Netherlands guilders, French francs, Japanese yen, British pound sterling and Canadian dollars over the period from January 1974 to April 1983. The problem required solving for efficient frontier portfolios containing only positive asset weights. The output of the program shows the negative elements which are typical of a foreign exchange variance-covariance matrix and which are indicative of the benefits to be gained from portfolio diversification. The output table shows that the (upper limit) efficient frontier portfolios were dominated by US dollar, Canadian dollar and British pound sterling investments for this data period.

RECOMMENDATIONS FOR EXTENDED APPLICATION OF THESE PROGRAMS

The programs described in this paper could be used as they are for a number of important applications in the fields of operational investment management and empirical portfolio management research.
However, a major advantage of having these SAS language programs is in the ease with which much more complex issues can be explored. For instance, suppose one is concerned with determining the statistical distribution of the elements of the asset weight vector (and/or other elements of the solution parameters) as a function of the input data distribution functions near a specified "operating point" in the multivariate space defined by the elements of a given variance-covariance matrix and yield vector. Analytical solution of this problem for all but the most elementary distributions is indeed difficult. Using the programs presented in this paper and the SAS MACRO facility, numerical solutions to the above proposed problem may be addressed by introducing appropriate perturbations into the individual elements of the variance-covariance matrix and the yield vector and then executing the appropriate program a number of times in a SAS MACRO do-loop. The result of this simulation exercise would be a numerical distribution function for the output data element(s) of interest.

A second broad area where the above programs may be applied (especially program one) is the area of studying the simultaneous performance of corporate asset and liability "portfolios" to better understand those attributes which lead to optimal performance. The study of commercial bank operations using portfolio methods has been initiated by a number of authors. Further useful work in the banking sector and in the non-banking corporate sector could be facilitated by the programs presented in this paper.

The final potential application of the presented programs to be suggested in this paper is in the area of multinational asset management. The past decade has been a period of increasing utilization of international short-term financial asset portfolio management methods by multinational corporations seeking to optimize the performance of their multicurrency portfolio of (inter alia) accounts payable and accounts receivable. The attached programs could be used as operational tools to assist in the solution of this form of management problem or they could be used as research tools in the study of imposing methods for the management of such multicurrency short-term combined asset and liability portfolios.

ENDNOTES

6. A mathematical statement of the problem solved by each program is contained in appendix one.
7. All examples used in this paper will contain local data for expository purposes. These programs have also been used extensively with on-line disk-based data sets. SAS users at any computer facility are able to assist in providing the facility-dependent programming necessary for introducing on-line data sets into SAS programs.
8. The previously cited Martin paper describes this process in detail.
9. Note: Technically, the efficient frontier constitutes only those asset portfolios on the upper limb of the quadratic solution. Solutions on both limbs are included in all discussions in this paper.
13. The SAS language has a number of probability distribution functions which could be helpful in modelling the desired perturbation distributions.
14. Chapter 13 of the previously cited Francis and Archer book has an excellent review of the literature in this area.

Mathematical and Program Appendices follow this page.

Authors address:
Alex Anckonie III
Dept. of Business Administration
George Washington University
Washington, D.C. 20052
APPENDIX ONE
Mathematical Form of the Program Problem

Given that a portfolio's mean yield (R) and standard deviation of the yield (SIDI) in matrix notation is given by:

\[ \text{ER} = E(R) \times X \]

and

\[ \text{SIDI} = \text{SQRT}(X^\top \times \text{SIGMAH} \times X) \]

where:

- \( X \) is an \( n \times 1 \) column vector of asset weights.
- \( \text{ER} \) is a \( 1 \times n \) row vector of mean asset yields.
- \( \text{SIGMAH} \) is an \( n \times n \) variance-covariance matrix of the asset yields.
- \( \text{SQRT} \) is the square root operator,
- \( \times \) is the matrix multiplication operator, and
- \( \times \) is the matrix transposition operator.

Computation Section One solves the following problem:

Given: \( \text{ER} \) and \( \text{SIGMAH} \)

Find: The \( X \) which Minimizes:

\[ S = \text{SQRT}(X^\top \times \text{SIGMAH} \times X) \]

Subject to: \( R = \) Given values selected and stepped by the program.

Computation Section Two solves the same problem with the additional constraint:

Subject to: \( X(i) \) greater than or equal to zero for all \( i = 1, \ldots, n \)

Where \( X(i) \) is the \( i \) element of the \( X \) vector.


APPENDIX TWO
Program One

```
// EXECC SAS
*******************************************************************************
* THE INPUT SECTION OF THE PROGRAM BEGINS HERE *
* YIELD DATA INPUT CASE - *
*******************************************************************************;
OPTIONS ls=73; *SAS OUTPUT FOR TERMINAL ;
* THIS JOB CONTAINS SHORT TERM INSTRUMENT AVERAGE *
* 30 DAY RETURNS TAKEN FROM THE MONTHLY FEDERAL *
* RESERVE BULLETIN COVERING THE PERIOD 12/79-5/82. *
* NOTE: THE VARIABLE MULI SHOULD BE SET TO 1.0 *
* FOR AN INITIAL RUN AND THEN INCREASED TO 1.5 *
* AND THEN 2.0 TO EXTEND THE EFFICIENT FRONTIER *
* UPTWARD AND TO THE RIGHT. *
*******************************************************************************;
DATA MININST;
  INPUT CP BA CD EC TBLIL;
* THIS SECTION CORRECTS THE "BANKER'S DISCOUNT *
  *RATE" DATA FOR CP, BA AND TILLS TO NORMAL *
*ANNUAL INTEREST (RETURN) RATES ;
CPCTR=(CP/4)/(1-CP/4)*4;
BACTR=(BA/4)/(1-BA/4)*4;
TRILL=(TBLIL/4)/(1-TBLIL/4)*4;
CARDS;
  .1324 .1331 .1343 .1451 .1204
  .1304 .1315 .1339 .1433 .1200
  .1378 .1402 .1430 .1533 .1286
  .1681 .1710 .1757 .1872 .1520
  .1578 .1563 .1614 .1791 .1320
  .0949 .0960 .0979 .1120 .0658
  .0827 .0831 .0849 .0941 .0707
  .0841 .0858 .0865 .0933 .0806
  .0957 .0985 .1001 .1062 .0913
  .1097 .1113 .1129 .1207 .1027
  .1252 .1269 .1294 .1355 .1162
  .1518 .1534 .1568 .1646 .1373
  .1807 .1796 .1865 .1947 .1549
  .1658 .1662 .1719 .1807 .1502
  .1549 .1554 .1614 .1718 .1479
  .1394 .1388 .1443 .1536 .1336
  .1456 .1465 .1508 .1595 .1369
  .1756 .1756 .1827 .2036 .1630
  .1632 .1627 .1690 .1786 .1473
  .1700 .1710 .1776 .1849 .1495
  .1723 .1722 .1796 .1898 .1551
  .1609 .1611 .1684 .1780 .1470
  .1485 .1478 .1539 .1634 .1354
  .1216 .1200 .1248 .1333 .1086
  .1212 .1213 .1249 .1324 .1085
  .1309 .1306 .1351 .1429 .1228
  .1453 .1447 .1500 .1579 .1348
  .1380 .1373 .1421 .1490 .1268
  .1406 .1395 .1444 .1518 .1270
  .1342 .1329 .1380 .1453 .1209
PROC CMPR CON NOMISS OUT=MININST2;
VAR CPCTR BACTR CD EC TBLIL;
*THE FOLLOWING CODE FORMS DATA SETS CONTAINING *
*THE COVARIANCE MATRIX AND THE VECTOR OF MEAN *
*RETURNS; *
DATA MININST3 MININST4;
SET MININST2;
IF TYPE_="CON" THEN OUTPUT MININST3;
ELSE IF TYPE_="MEAN" THEN OUTPUT MININST4;
PROC MATRIX;
FETCH SIGMAH DATA=MININST3;
FETCH ER DATA=MININST4;
PRINT SIGMAH;
PRINT ER;
*******************************************************************************;
* COMPUTATION SECTION BEGINS HERE *
*******************************************************************************;
* CURVED CASE *
*******************************************************************************;
SIGMA2=SIGMAH+SIGMAH;
N=NCOL(ER);
MINN-MIN(ER);
MULI=2.0;
MAXX=(NAX2(MULI-MULI-MULI)+MINN)
STEP=(MAXX-MINN)/(MULI*10);
NULL4=0.0;
0.0;
ER=ER-ER;
AR=ER\*ER;
A=ER\*AR;
B=A/NULL4;
C=(SIGMA2/2)/B;
```
CINV=C**-1;
KB=1;
KT=ERP-ERP;
DO R=MNN TO MAXX BY STEP;
KVAR=R;
KI=KT/I/KVAR/KB;
OUTPUT R OUT=DS2 (RENAME=(COL1=RETURN));
XE=CINV*KI;
XI=XE(1:N,1);
WRS=;
OUTPUT WRS OUT=DS1;
STD=SQR(XI'*SIGMAH*XI);
OUTPUT STD OUT=DS3 (RENAME=(CDL1=STD DEll));
END;
************************************************
* =:
**THE TABLE ON THE SECOND PORTION OF OUTPUT **
* HAS THE FOLLOWING FORMAT- THE VALUE IN COLUMN N **
* IS THE WEIGHT OF SIMILARITY N IN THE **
** OPTIMAL **
* UNCONSTRAINED (SHORT SELLING ALLOWED) PORTFOLIO. **
** THE STANDARD DEVIATION **
* AND THE EXPECTED **
** RETURN FOR EACH PORTFOLIO ON THE EFFICIENT FRONTIER **
* ARE ALSO GIVEN. EACH OUTPUT DATA LINE COVERS **
* ONE POINT ON THE EFFICIENT FRONTIER. **
************************************************
* COMPUTATION SECTION ENDS HERE
* STANDARD OUTPUT SECTION BEGINS HERE
************************************************
DATA OUTSECl';
DS1 DS2 DS3;
DROP RCW';
PROC PRINT DATA=OUTS=;
PIDC pwr '
DATA=OUTSECl';
PIDT RE'lURN*S'ID DEV=' * / VZERO
TITLE PIDT OF
EXPECI'ED
REIURN VS. STD' DEV;
PROC PIDT DATA=OUTSECl';
PIDT RE'lURN*S'ID DEV=' * , ;
APPENDIX
THREE
Program Two
// EXEC SAS
*************************************************************************
*THE INPUT SECTION OF THE PROGRAM BEGINS HERE
*C=VAR-COV MATRIX AND YIELD VECTOR INPUT CASE
*************************************************************************
* OPTIONS LS=73; SETS OUTPUT FOR TERMINAL ;
* PROC MATRIX;
* FOLLOWING ENTERS THE VAR-COV MATRIX
* COLUMN NUMBERS IN THE VAR-COV MATRIX AND ER
* VECTOR CORRESPOND TO DEPOSIT NUMERAISS AS
* 1=US 2=SF 3=DM 4=DE 5=JP 6=YN
* 7=ST 8=CN;
*SIGMAH=.0001725 -.0001456 -.0001264 -.0001294
-.0001297 .00000466 -.0001217 .0009707
-.0001312 .0001559 .0001369 .0001333 .0001405
-.0000470 .0001085 .0000327
-.0001256 .0001369 .0001320 .0001351 .0000808
-.0000609 .0001045 .00003156
-.0001294 .0001333 .0001351 .0001416 .0001386
-.0000255 .0001069 .00003299
-.0001297 .0001405 .0001354 .0001385 .0001564
-.0000275 .0001072 .00003237
-.0000468 .0000637 .0000609 .0000255
-.0000275 .0001098 .0000346 .00003692
-.00002157 .0001084 .0001045 .0001069
.00011027 .0003406 .00001449 .000002478
.000009707 .0000274 .00003156 .00003229
.00003277 .00003692 .00002478 .00004165
**FOLLOWING ENTERS THE MEAN VECTOR OF YIELDS ;
ER=.01081 .004048 .002767 .002921 .0003392
.003830 .00003956 .014276 ;
PRINT SIGMAH;
PRINT ER;
*************************************************************************
* INPUT SECTION ENDS HERE
* CONSTRAINED COMPUTATION SECTION BEGINS HERE
*************************************************************************
SIGMA2=SIGMAH+SIGMAH;
Z=SIGMAH;
MINN=MIN(ER);
MAXX=MAX(ER);
STEP=(MAXX-MINN) #/10;
NULL4=0 0
0 0
ER=ER';
AR=ER#/ER';
A=ER#1
B=1(1NULL4);
KB=1;
KT=ERP-ERP;
MINN=MINN+STEP#/2;
MAXX=MAXX+STEP#/2;
DO R=MNN TO MAXX BY STEP;
TABRONUM=1;
NUMAPRIM=1;
BIGSQ(J, NCOL(2), NCOL(2), 0);
SIGMA2=3;
N=NCOL(ER);
C=SIGMAH/1/;
KVAR=R;
KI=KT/I/KVAR/KB;
LABEL:
INVERC=INV(C);
XE=INVERC*KI;
XI=XE(1:N,1);
WRS=;
STD=SQR(XI'*SIGMAH*XI);
K=1;
J=1;
IF MIN(WRS)(J THEN DO I=1 TO N BY 1;
IF WRS(I,J) THEN DO;
J=J+1;
END;
ELSE DO;
END;
WRS=;
SIGMA2=SQR(XI'*SIGMAH*XI);
**************************************************************************
INDEX=1;
DO V=1 TO NCOL(Z);
   IF WSS(V,V) > 0 THEN DO;
      WSS(V,V)=WSS(V,INDEX);
      INDEX=INDEX+1;
   END;
   ELSE DO;
      WSS(V,V)=0;
   END;
END;

*FOLLOWING OUTPUTS WEIGHT VECTOR AND STD DEV;
OUTPUT WSS =OUT=DS1;
OUTPUT STD OUT=DS3 (RENAME=(COL1=STD_DEV));
GO TO LABEL;
END;

*FOLLOWING PROVIDES APRIME VECTOR OF LOCATION OF NON NEG ELEMENTS IN PRIOR TABLEAU AND PLACES THIS IN BIGSQ MATRIX AND SETS UP THE PARAMETERS FOR NEXT TABLEAU COMPUTATION;
G=N+1;
H=N+2;
APRIME=E(2:J);
TABFIL=J(1, (NCOL(Z)-NCOL(APRIME)), 0);
BIGSQ(TABFIL)=APRIME*TABFIL;
TABRONUM=TABRONUM+1;
CPRIME=APRIME/G/H;
C=C(CPRIME,CPRIME);
SIGMAH=SIGMAH(APRIME,APRIME);
N=NCOL(APRIME);
KI=APRIME/-APRIME'//KVAR/K8;
GO TO LABEL;
LABEL1: END;

************************************************
*NOTE: THE TABLE ON THE SECOND PAGE OF OUTPUT*
*HAS THE FOLLOWING FORMAT: THE VALUE IN COLUMN*
*N IS THE WEIGHT OF SECURITY N IN THE OPTIMAL*
*CONSTRAINED PORTFOLIO. THE STANDARD DEVIATION*
*AND THE EXPECTED RETURN FOR EACH PORTFOLIO ON*
*THE EFFICIENT FRONTIER ARE ALSO GIVEN. EACH*
*DATA LINE IN THE OUTPUT TABLE COVERS ONE POINT*
*ON THE EFFICIENT FRONTIER.
*
******************************************************************************;
*COMPUTATION SECTION ENDS HERE*
******************************************************************************;

************************************************
* STANDARD OUTPUT SECTION BEGINS HERE*
******************************************************************************;
DATA OUTSECT;
MERGE DS1 DS2 DS3;
DROP ROW;
PROC PRINT DATA=OUTSECT;
PROC PLOT DATA=OUTSECT;
   PLOT RETURN*STD_DEV="" / VBAR HEERO;
   TITLE PLOT OF EXPECTED RETURN (VERTICAL) VS. STANDARD DEVIATION;
PROC PLOT DATA=OUTSECT;
   PLOT RETURN*STD_DEV="";

APPENDIX FOUR

Sample Output of Program Two

Output Table

<table>
<thead>
<tr>
<th>OBS</th>
<th>COL1</th>
<th>COL2</th>
<th>COL3</th>
<th>COL4</th>
<th>COL5</th>
<th>COL6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.065</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>2</td>
<td>.196</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>3</td>
<td>.327</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>4</td>
<td>.458</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>5</td>
<td>.589</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>6</td>
<td>.720</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>7</td>
<td>.821</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>8</td>
<td>.735</td>
<td>.003</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>9</td>
<td>.634</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>10</td>
<td>.211</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
</tbody>
</table>

(Note: weights may not sum to one due to truncation errors associated with shortening table for this display.)

(Copies of a more complete set of output with plots etc, are available from the author.)

60