

Advanced Analysis of Stock Prices Using the CRSP Tapes and SAS: Protocols for Both Academic Research and Classroom Use

Glenn Millard and Garland Simmons¹

ABSTRACT

Analysis of stock price data made available by the Center for Research in Security Prices (CRSP) is conducted using SAS to write computer code which permits the extraction and statistical analysis of NYSE and AMEX listed common stock returns. These programs use advanced features of SAS version 6.08 such as random access of indexed datasets to provide error checking and information for output. Care has been taken in the estimation of Alpha and Beta statistics to address problems of autocorrelation which may be present when working with a time series of daily returns.

Introduction

The subject of this paper is a collection of SAS computer programs which have been developed to serve as vehicles of empirical research in financial economics.² These programs were designed, among other things, to permit scholars and analysts to calculate what are known in finance as alpha and Beta estimates for common stocks using a time series of daily returns made available by the Center for Research in Security Prices (CRSP) at the University of Chicago. It is hoped that our work here will prove useful to academics as they seek to test to evaluate elements of the theory of financial economics as well as to money managers as they design portfolios of common stocks to maximize the expected return on investment for some given level of risk.

CRSP provides data in a number of formats; however, our analysis is limited to the

¹Stephen F Austin State University

² SAS is a registered trademark of the SAS Corporation.

analysis of the Daily NYSE/AMEX Returns file. It is a reasonably large file, about 200 megabytes in size for data from 1962 to 1993. CRSP also provides FORTRAN source programs for accessing the files, and we use these programs after some modification to output the data in a format amenable for analysis by SAS. Parts of the datafile were also made into SAS indexed datasets for use later in the analysis. After the data has been extracted SAS programs check the data for missing values and automatically drop stocks that have more than six days of missing data. Various records from the main datafile have also been made into SAS index datasets for ease in merging this data into the reports these include Header records which contain identification and data range variables for each security, Names records containing information on the security's name history, and Delist records about security delisting are defined on a company by company basis. Return records are defined by company and by trading day. Indices such as the Standard & Poors 500 Index and the CRSP Value-weighted Index are defined by Index and by trading day, similar to the way Return records are organized.

DESCRIPTION OF STATISTICAL ANALYSIS

Both money managers and scholars alike wish to define the connection between the rate of return on investment earned by investors and the risk of that investment. Capital Asset Pricing theory, which is sometimes referred to as Beta theory, provides that the expected rate of return on investment is a function of its Beta risk measure. At a practical level, Beta estimates necessary to operate the theory are derived from an application of what

has become known as the market model which is described below.

$$R_{i,t} = \hat{\alpha}_i + \hat{\beta}_i R_{m,t} + \epsilon_{i,t}$$

The rate of return $R_{i,t}$ for security i at time period t is expressed as a simple linear regression against an index of stock market activity $R_{m,t}$, where Alpha, the y-intercept term, and Beta, the slope of the regression line, are parameters estimated for each security i are from a time-series of N historical rates of return such that the expected value of the error term is zero. Because we use daily returns Ordinary Least Squares (OLS) estimation procedures may not produce efficient, unbiased parameter estimates of Alpha and Beta. Although the use of daily returns is desirable because one is able acquire a large number of observations from a recent history of returns, there exists the problem of autocorrelation in the error term when using a series of daily returns. Brown and Warner deal with this problem and their computing procedures are employed in the computer code we provide.

THE DATA SELECTION PROGRAM

(See Appendix A)

Many times financial analysis of a security, or a portfolio of securities is conducted because of some particular event. For example, we may wish to study returns for all companies that have cut their dividends to see if the returns before the announcement of a

dividend cut are different than the returns after. Our data selection program uses SAS dataset windows to provide a user-friendly interface so that people with very little computer experience can study events which occur in the stock market. Users are permitted to select companies to be studied and a range of trading days before and/or after a desired event day, also selected by the user. This event day can be the same for each security under study or different dates for each security can be used. In the dividend cut example above, we would need to specify a different date for each company under study, but if we were going to study the stock returns for retailers around Christmas, we would choose the same event day for all companies. The program allows for the securities to be selected by the CUSIP identifier or the CRSP permanent identifier. The selected event day may be selected by either the calendar date or by the trading date. No doubt most users will use the normal calendar time definitions for the time periods which they choose to study, however, some studies of stock returns do require knowledge of trading days, so both options are made available. Date selections no matter which format the user selects are tested for validity so that impossible choices as to dates will not trouble the user.

Readability and ease-of-use is enhanced by using the Windows GROUP subcommand allowing screens to provide just the needed information needed during that particular part of the entry process. Proper use of these subcommands can give your screens the appearance that you would expect from SCL code in FSP or AF windows.

The selection program outputs two data files that are used by the CRSP FORTRAN program to control the data it outputs. One of these files is used by later programs to aid in the extraction of possible missing data in the output.

THE LOOK FOR MISSING DATA

(See Appendix B and C)

In previous studies of the data it was noted that even though the Header record showed the starting and ending trading dates for the security it may still have not traded for a lengthy period of time between these two dates. Because of this it was deemed necessary to drop any security from the analysis that had not traded for over 6 days during the selected period. The first of the two SAS programs that handle the possible missing data merges in index data from the Calendar dataset that will be needed in the later analysis. It then separates the valid data and the missing data into two separate datasets. Nothing is done with the missing data file, but it is retained in case further study of this data is desired. The second SAS program is then used to determine how many missing data records exist for each security selected. A file is created that lists this information so that the analysis program can be provided with information as to whether a security should be dropped from the analysis. Some programming to handle special cases is present in these programs to handle various cases, such as the fact that an empty dataset would have been created if there was no missing data and this would cause problems with the analysis program.

BETA ANALYSIS AND INDEX PROGRAMS

(See Appendix D and E)

Many times portfolio managers wish to consider properties of a large universe of common stocks, and then from out of this large universe construct a portfolio which possess certain Beta and Alpha properties. We have provided SAS code to analyze the annual return on investment, Alpha, and Beta of NYSE and AMEX listed common stocks on a year by year basis. This analysis program uses the various statistical methods described above to analyze the data. Each security is checked against the results of the missing data tests. If it is determined that the amount of missing data for a given security is excessive, then it is dropped from the analysis. One other SAS program was written to produce yearly CRSP value weighted returns and Standard & Poor 500 return of investment calculations. This information is contained in the Calendar dataset and can be run at any time, independently from the other programs.

Further Reading

Those who wish to know more about Beta Theory, the debate about this theory and the future of Beta in theory and in practice should consult: Grinold, Richard C., "Is Beta Dead Again," FINANCIAL ANALYSTS JOURNAL, July-August, 1993. For those interested in the special econometric problems which must be addressed when analyzing a

time series of daily returns, the following citation is indispensable: Brown, Stephen J., and Jerome B. Warner, "Using Daily Stock Returns: The Case of Event Studies," JOURNAL OF FINANCIAL ECONOMICS, 1984, p.431. Materials describing the CRSP files are available from the Center For Research in Security Prices, Graduate School of Business, University of Chicago. CRSP_MAIL@GSBACD.UCHICAGO.EDU

Appendix A

DATA ENTRY PROGRAM

```

/*****
*          DATA ENTRY PROGRAM          *
*                                     *
* Variables:                          *
* crspnum/cusip = S&P assigned identifier for the security *
* permno       = CRSP assigned identifier for the security *
* calbeg       = first list date for security              *
* calend       = last list date for security              *
* compnm       = Company name                             *
* cdate        = the event date                           *
* bdays       = number of trading days before the event date *
* adays        = number of trading days after the event date *
* j            = number of securities selected             *
* caldt        = Calendar date                            *
*****/

data _null_;
length crspnum $8. cusip $8. fin $1. type $1. calbeg 3. calend 3. compnm $32.
      ffile $9. cdate 4. ydate 6. bdays 3. adays 3. pmt $29. errmsg $45.;

```

```

ffile="stock.dat";
file ifile filevar=ffile noprint notitle;
loop1;;

```

```

/*****
*   Entry window #1
*****/

```

window startup

group=start1

```

#3 @20 "CRSP DATA ENTRY PROGRAM"
#5 @10 "This program allows entry of CRSP id's and event days."
#6 @10 "First you need to enter how many days you wish to"
#7 @10 "gather data on before and after the event day."
#9 @10 "Enter number of days you wish to gather data on -"
#11 @15 "before the event day:" +1 bdays a=rev
#12 @15 "after the event day:" +2 adays a=rev
#14 @10 "Do you wish to enter the event day as:"
#15 @15 "1 - CRSP trading date or 2 - calendar date : " +1 type
#17 @15 "Do you wish to enter companies by:"
#18 @15 "1 - CUSIP number or 2 - PERMNO : " +1 ctype
group=start2
#20 @20 "Is this correct?" +1 fin;

```

```

/*****
*   Entry window #2
*****/

```

window entry

group=begin1

```

#3 @20 "CRSP STOCK SELECTION UTILITY"
#5 @5 "Enter CRSP number of company you wish to select:" +1 crspnum a=rev
group=begin2
#3 @20 "CRSP STOCK SELECTION UTILITY"
#5 @5 "Enter PERM number of company you wish to select:" +1 permnum a=rev
group=data
#3 @20 "CRSP STOCK SELECTION UTILITY"
#5 @10 "CUSIP:" +1 cusip protect=yes + 5 "PERMNO:" +1 permno protect=yes
#6 @10 "Company name:" +1 compnm protect=yes
#8 @10 "First list date for company:" +1 calbeg protect=yes
+4 begdat protect=yes
#9 @10 "Last list date for company:" +2 calend protect=yes
+4 enddat protect=yes
#12 @10 pmt protect=yes +1 ydate a=rev

```

```

group=end
#18 @20 "Are you finished?"
#18 @40 fin a=rev
group=error
#18 @20 errormsg protect=yes;

/*****/
*   Begin program by displaying first screen   *
/*****/

display startup.start1 blank;
display startup.start2;
if fin = "y" or FIN = "Y" then goto entloop;
goto loop1;

/*****/
*   Test for entry of valid CRSP number or PERMNO   *
/*****/

entloop;;
if ctype = 1 then do;
    display entry.begin1 blank;
    cusip=crspnum;
    set crsp93.head key=cusip;
    if _iorc_=1230015 then do;
        errormsg = "Invalid cusip number";
        display entry.error;
        goto entloop;
    end;
end;
if ctype = 2 then do;
    display entry.begin2 blank;
    permno=permnum;
    set crsp93.head key=permno;
    if _iorc_=1230015 then do;
        errormsg = "Invalid permno";
        display entry.error;
        goto entloop;
    end;
end;

/*****/
*   Test for entry of valid calendar or trading date   *
/*****/

```

```

newdates;;
if type = 1 then do;
  pmt = "Enter CRSP trading date:";
  display entry.data blank;
  cdate = ydate;
end;
if type = 2 then do;
  pmt = "Enter calendar date (yymmdd):";
  display entry.data blank;
  caldt=ydate;
  set crsp93.calendar key=caldt;
  if _iorc_=1230015 then do;
    errormsg = "Invalid date";
    display entry.error;
    goto newdates;
  end;
  cdate=i;
end;

/*****
* Test to make sure security was trading for all selected dates *
*****/

bdate = cdate - bdays;
edate = cdate + adays;
if bdate < begdat then do;
  errormsg = "Beginning date before first trading day";
  display entry.error;
  goto newdates;
end;
if edate > enddat then do;
  errormsg = "Ending date after last trading day";
  display entry.error;
  goto newdates;
end;
j+1;
put crspnum bdate edate;
display entry.end;

/*****
* Ask if other securities are to be selected, if not, end *
*****/

if fin = "Y" or fin = "y" then goto endit;

```

```
goto entloop;  
endit;;  
ffile="param.dat";  
file ifile filevar=ffile noprint notitle;  
put j 4. bdays 4. adays 4.;  
stop;  
run;  
quit;
```

Appendix B

MISSING DATA PROGRAM #1

```
data sasuser.mergedat sasuser.except;
```

```
/******  
* The DCL for the FORTRAN program that extracts the returns *  
* allows the output to written to a user selected file. This *  
* filename is written to another file that is now read into *  
* this program so it can find the data. *  
/******
```

```
infile 'outset.txt' pad;  
input fname $20.;  
put fname;  
infile ifile filevar=fname pad missover end=done;
```

```
/******  
* Read in the extracted returns and merge them with index *  
* information from the Calendar file. *  
* *  
* vwretcd = Value Weighted Return Index *  
* cusip = S&P assigned security identifier *  
* permno = CUSP assigned security identifier *  
* tdate = current trading date *  
* return = daily holding period return *  
* hsiccd = the Standard Industrial Classification Code *  
* hexcd = the Header Exchange Code *  
/******
```

```
do until(done);  
input cusip $ permno tdate return hsiccd hexcd;  
i=tdate;  
set crsp93.calendar key=i;  
keep cusip permno tdate return vwretcd hsiccd hexcd;
```

```
/******  
* Look for missing data and separate the missing data from the *  
* valid data. *  
* On date when the security was not trading a code greater than *  
* -50.0 is used to indicate the reason it wasn't traded. *  
/******
```

```

/*****/

if return < -50.0 then output sasuser.except;
else output sasuser.mergedat;
end;

/*****/
* Write a last null record out. This is to make sure the file *
* is created in case there are no missing records. This can *
* cause problems with later programs. *
/*****/

cusip = "000000000";
tdate = 0;
return = 0;
hsiccd = 0;
hexcd = 0;
output sasuser.except;
run;

```

Appendix C

MISSING DATA PROGRAM #2

```
data sasuser.notrade;
file 'except.report' noprint notitle;
retain notrade 0 stock ' ' i 1;

/*****
 * Check to see if any missing data records were written, if not, *
 * then write no missing records to file for each security.      *
*****/

infile 'stock.dat' end=endit;
set sasuser.except end=last;
if cusip = '00000000' then go to finish;
put last i;
if last then go to finish;
by cusip;

/*****
 * Read file and determine how many missing records exist for each *
 * security and write the results out.                               *
*****/

if first.cusip then do;
loop;;
if stock not = cusip then do;
if i = 0 or stock = ' ' then go to stepover;
put stock 'missed no trading days';
output;
stepover;;
i=1;
input stock $8. begdat enddat;
go to loop;
end;
end;
notrade + 1;
if last.cusip then do;
set crsp93.head key=cusip;
put cusip compnm 'was not trading for ' notrade ' days';
keep stock notrade;
output;
```

```

    notrade = 0;
    i = 0;
end;
return;

```

```

/*****
* Make sure no missing records are written for each security.      *
* Main file contains only missing records so this program must    *
* determine which securities have no missing records.              *
*****/

```

```

finish::
    notrade = 0;
    put stock 'missed no trading days';
    output;
    input stock $8. begdat enddat;
    if endit then go to thatsall;
    go to finish;
thatsall::
    put stock 'missed no trading days';
    output;
run;

```

Appendix D

ANALYSIS PROGRAM

```
options linesize=80;
data all;
infile 'stock.dat' pad missover;;
file 'alpha.report';
retain delcomp a 0 bday eday bdate edate;
set sasuser.mergedat end=last;
by cusip;

/*****
* Retrieve beginning and ending calendar dates for each security
* from Calendar dataset.
*****/

if first.cusip then do;
  i=0;
  input cusp $8. bdate edate;
  i=bdate;
  set crsp93.calendar point=i;
  if _iorc_ = 1230015 then put "error #1 " i;
  bday = caldt;
  i=edate;
  set crsp93.calendar point=i;
  if _iorc_ = 1230015 then put "error #2 " i;
  eday = caldt;
  a+1;

/*****
* Check to see if there are more than the desired minimal missing
* records for the security. If yes, then remove the security from
* the analysis.
*****/

set sasuser.notrade;
delcomp = 0;
if notrade > 6 then delcomp = 1;
end;
if delcomp = 1 then return;
```

```

/*****
* Accumulate totals from individual records before making
* calculations.
*****/

y=return;
x=vwretd;
retain sumxy sumxsq sumysq sumx sumy lagx lagy j 0 hperiod 1;
hperiod = hperiod * (1.0 + return);
j+1;
lagy=lag1(y);
lagx=lag1(x);
yield =Yield*(1+Y);
iyield =iyield*(1+x);
retain yield iyield 1.;
sumx+x;
sumy+y;
sumxy+x*y;
sumxsq+x*x;
sumysq+y*y;
if j > 1 then do;
retain sumxy sumxsq sumysq lagxy leadxy sumx sumy
lagxx sumlagx sumlagy 0;
sumx1+x;
sumy1+y;
sumlagx+lagx;
sumlagy+lagy;
sumxy1+x*y;
sumxsq1+x*x;
sumysq1+y*y;
lagxx+lagx*x;
lagxy+lagx*y;
leadxy+x*lagy;
retain summark sumcomp k 0;
if not (last or last.cusip) then do;
summark=summark+x;
sumcomp=sumcomp+y;
k=j-1;
end;
end;
if last or last.cusip then go to final;
delete;
return;

```

```

/*****
* Calculate Alpha, Beta, Rsquare, and 1+Holding Period Statistics *
*****/

final;;
keep hsiccd hexcd cusip permno;
t=j;
covxy=(sumxy-(sumx)*(sumy)/t)/(t-1);
varx=(sumxsq-(sumx)**2/t)/(t-1);
vary=(sumysq-(sumy)**2/t)/(t-1);
betaols=covxy/varx;
n=j-1;
covlagxy=(lagxy-sumlagx*sumy1/n)/(n-1);
covlagxx=(lagxx-sumlagx*sumx1/n)/(n-1);
covxlgy=(leadxy-sumx1*sumlagy/n)/(n-1);
keep beta rho RSQUARE alpha;
betalead=covxlgy/varx;
betalag=covlagxy/varx;
rho=covlagxx/varx;
beta=(betalead+betaols+betalag)/(1+2*rho);
RSQUARE=((beta)*SQRT(varx)/SQRT(vary))**2;
keep yield iyield;
keep summark sumcomp k;
alpha=((1/k)*sumcomp)-((beta)*(1/k)*summark));
output;
j=0;

/*****
* Output calculations. *
*****/

if a = 1 then do;
put ' NYSE and AMEX Assets';
put ' ';
put ' CRSP PERM SIC 1+HOLDING';
put ' COMPANY NAME NUMBER NUMBER CODE PERIOD'
' BETA ALPHA RSQUARE BEGIN DATE END DATE';
end;
set crsp93.head key=cusip;
put compnm $32. cusip $9. permno 6. hsiccd 6. hperiod 9.5 beta 10.5
alpha 9.5 rsquare 9.5 bday 10. eday 10.;
if not last then return;
run;

```

Appendix E

INDEX ANALYSIS PROGRAM

```

/*****
* Index Return Yearly Statistics
*
* caldt  = calendar date of trading day
* vwretd = value weighted return index for trading day
* sprtrn = Standard & Poor's Composite Index for trading day
*
* Determine calendar year each trading date belongs in.
*****/

data sasuser.rinvest;
attrib year length=$2.;
set crsp93.calendar;
chardate=put(caldt,z6.);
year = substr (chardate,1,2);
keep year vwretd sprtrn;
run;

/*****
* Read and calculate yearly indexes for value weighted return and
* S&P Composite Index.
*
* vannual = 1 + Value Weighted Return for year
* sannual = 1 + S&P Composite Index for year
*****/

data _null_;
file 'iret.report' notitle;
do until (last.year);
set sasuser.rinvest (firstobs=2) end=stopnow;
by year;
retain vannual sannual 1.0 j 0;
if first.year then j + 1;
if j = 1 and first.year then do;
put '          Annual Return on Investment';
put '          CRSP Value Weighted Index and S&P 500 Index';
put '          1962 - 1993';
put ' ';

```

```

put ' ';
put ' YEAR ANNUAL CRSP (VWI) RETURN ON INVESTMENT S&P 500';
put ' ';
end;
vannual = vannual * (1 + vwretd);
sannual = sannual * (1 + sprtrn);
end;
put ' ' year ' ' vannual 9.5 ' ' sannual 9.5;
vannual = 1.0;
sannual = 1.0;
if not stopnow then return;
run;

```