

```
*comp.funda;
*crsp;
```

```
*Amendations: 1. Changed the definition of be in compf;
*2. In the first instance of LME, made the first. an amended first.;
```

```

/***** */
/* Program      : fama_french_factors_replication.sas      */
/* Authors      : Luis Palacios (WRDS), Premal Vora (Penn State) */
/* Date        : 04/2009                                     */
/* Description   : Replicates Fama-French (1993) SMB & HML Factors */
/***** */

*options obs=20000;
/*libname comp  '~/bigspace/dec2009/correlations/btm_portfolios/correct/wrdsdata/';
libname crsp   '~/bigspace/dec2009/correlations/btm_portfolios/correct/wrdsdata/';*/
libname cc     '~/bigspace/dec2009/correlations/btm_portfolios/current/wrdsdata/';
libname myh    '~/bigspace/dec2009/correlations/btm_portfolios/current/data/';

/***** Part 1: Compustat *****/
/* Compustat XpressFeed Variables: */
/* AT      = Total Assets */
/* PSTKL   = Preferred Stock Liquidating Value */
/* TXDITC  = Deferred Taxes and Investment Tax Credit */
/* PSTKRV  = Preferred Stock Redemption Value */
/* CEQ     = Common/Ordinary Equity - Total */
/* PSTK    = Preferred/Preference Stock (Capital) - Total */

/* In calculating Book Equity, incorporate Preferred Stock (PS) values */
/* use the redemption value of PS, or the liquidation value */
/* or the par value (in that order) (FF,JFE, 1993, p. 8) */
/* USE Balance Sheet Deferred Taxes TXDITC if available */
/* Flag for number of years in Compustat (<2 likely backfilled data) */
%let vars = AT PSTKL TXDITC PSTKRV CEQ PSTK seq lt;
data compf;
  set comp.funda(keep= gvkey datadate &vars indfmt datafmt popsrc consol csho);
  by gvkey datadate;
  where indfmt='INDL' and datafmt='STD' and popsrc='D' and consol='C'
    and datadate >= '01Jan1959'd;
    /* Two years of accounting data before 1962 */
  PS = coalesce(PSTKRV,PSTKL,PSTK,0);
  if missing(TXDITC) then TXDITC = 0 ;

  se=coalesce(seq, sum(ceq, ps), at-lt);
  be=se-ps+txditc;
  year = year(datadate);
  label BE='Book Value of Equity FYear t-1' ;
  drop indfmt datafmt popsrc consol ps &vars;
  retain count;
  if first.gvkey then count=1;
  else count = count+1;
run;

/***** Part 2: CRSP *****/
/* Create a CRSP Subsample with Monthly Stock and Event Variables */
/* This procedure creates a SAS dataset named "CRSP_M" */
/* Restrictions will be applied later */
/* Select variables from the CRSP monthly stock and event datasets */
%let msevars=ticker ncusip shrcd exchcd;
%let msfvars = prc ret retx shrout cfacpr cfacshr;
```

```
%include 'crspmerge.sas' ;
```

```
%crspmerge(s=m,start=01jan1959,end=31dec2009,sfvars=&msfvars,sevars=&msevars,filters=exchcd in (1,2,3));
```

```
/* CRSP_M is sorted by date and permno) and has historical returns */
/* as well as historical share codes and exchange codes */
```

```
/* Add CRSP delisting returns */
```

```
proc sql;
  create table crsp_m2
    as select a.*, b.dlret,
      sum(1,ret)*sum(1,dlret)-1 as retadj "adjusted return for delisting",
      abs(a.prc*a.shrcd) as MEq 'Market Value of Equity',
      abs(a.prc) as absprc 'Market Price'
    from CRSP_M as a left join crsp.msdelist (where=(missing(dlret)=0 ) ) as b
    on a.permno=b.permno and
      intnx('month',a.date,0,'E')=intnx('month',b.DLSTDT,0,'E')
    order by date, permco, MEq;
```

```
quit;
```

```
/* There are cases when the same firm (permco) has two or more */
/* securities (permno) at same date. For the purpose of ME for */
/* the firm, we aggregated all ME for a given permco, date. This */
/* aggregated ME will be assigned to the Permno with the largest ME */
```

```
data crsp_m2a (drop = Meq);
  set crsp_m2;
  by date permco Meq;
  retain ME;
  if first.permco and last.permco then do;
    ME=meq;
    uniquepermco=1;
    output; * most common case where a firm has a unique permno;
  end;
  else do ;
    if first.permco then ME=meq;
    else ME=sum(meq,ME);
    uniquepermco=0;
    if last.permco then output;
  end;
```

```
run;
```

```
proc sort data=crsp_m2a; by permno date;run;
```

```
/* The next step does 2 things: */
/* - Create weights for later calculation of VW returns. */
/* Each firm's monthly return RET t will be weighted by */
/* ME t-1 = ME t-2 * (1 + RETX t-1) */
/* where RETX is the without-dividend return. */
/* - Create a File with December t-1 Market Equity (ME) */
```

```
data crsp_m3 (keep = permno date ret dlret retadj weight_port ME exchcd shrcd cumretx)
  decme (keep = permno date ME absprc uniquepermco permco rename=(me=DEC_ME absprc=de
cprc) ) ;
  set crsp_m2a;
  by permno date;
  retain weight_port cumretx me_base;
  Lpermno=lag(permno);
  LME=lag(me);
  if first.permno then do;
    LME=me/(1+retx);
    cumretx=sum(1,retx);
    me_base=LME;
```

```
        weight_port=.;
    end;
else do;
    if month(date)=7 then do;
        weight_port= LME;
        me_base=LME; /* lag ME also at the end of June */
        cumretx=sum(1,retx);
        end;
    else do;
        if LME>0 then weight_port=cumretx*me_base;
        else weight_port=.;
        cumretx=cumretx*sum(1,retx);
        end;
    end;
    gap = weight_port - lme;
    output crsp_m3;
    if month(date)=12 and ME>0 then output decme;
run;
/* Create a file with data for each June with ME from previous December */
proc sql;
    create table crsp_june as
        select a.*, b.DEC_ME, b.uniquepermco, b.decprc
        from crsp_m3( where=(month(date)=6)) as a, decme as b
        where a.permno=b.permno and
        intck('month',b.date,a.date)=6;
quit;
run;

/***** Part 3: Merging CRSP and Compustat *****/
/* Add Permno to Compustat sample */

proc sql;
    create table ccm1 as
        select a.*, b.lpermno as permno, b.linkprim
        from Compf as a, cc.ccmxpf_linktable as b
        where a.gvkey=b.gvkey
        and substr(b.linktype,1,1)='L'
        and (intnx('month',intnx('year',a.datadate,0,'E'),6,'E') >= b.linkdt    or miss
ing(b.linkdt))
        and (b.linkenddt >= intnx('month',intnx('year',a.datadate,0,'E'),6,'E') or miss
ing(b.linkenddt));
quit;

/*OK, take a particular datadate: 31Dec2000. The year intnx takes it to 01Jan2000. The
6 mo ahead takes it to 30 Jun 2000
What about 31May2000? 01Jan2000 and 30Jun2000.*/

/* Cleaning Compustat Data for no relevant duplicates */
/* 1. Eliminating overlapping matching : few cases where different gvkeys */
/* for same permno-date --- some of them are not 'primary' matches in CCM.*/
/* Use linkprim='P' for selecting just one gvkey-permno-date combination; */
proc sort data=ccm1 out=ccm2;
    by datadate permno descending linkprim gvkey;
run;

data ccm1a;
    set ccm2;
    by datadate permno descending linkprim gvkey;
    if first.permno;
run;

/* Sanity Check -- No Duplicates */
```

```
proc sort data=ccm1a nodupkey; by datadate permno; run;

/* 2. However, there other type of duplicates within the year */
/* Some companies change fiscal year end in the middle of the calendar year. */
/* In these cases, there are more than one annual record for accounting data.*/
/* We will be selecting the last annual record in a given calendar year; */

proc sort data=ccm1a; by permno year datadate ; run;

data ccm2a ;
    set ccm1a;
    by permno year datadate ;
    if last.year =1 ;
run;

/* Sanity Check -- No Duplicates */
proc sort data=ccm2a nodupkey; by permno datadate; run;
/* End cleaning Compustat */

/*Import DFF*/
data DFF_BE;
    infile '~/bigspace/dec2009/correlations/btm_portfolios/current/wrdsdata/DFF_BE_With
_Nonindust.txt'
        missover dsd lrecl=32767;
    input
        permno 1-7
        year1 8-12
        year2 13-17
        var1 18-29 var2 30-41 var3 42-53 var4 54-65 var5 66-77 var6 78-89 var7 90-101 v
ar8 102-113
        var9 114-125 var10 126-137 var11 138-149 var12 150-161 var13 162-173 var14 174
-185 var15 186-197
        var16 198-209 var17 210-221 var18 222-233 var19 234-245 var20 246-257 var21 258
-269 var22 270-281
        var23 282-293 var24 294-305 var25 306-317 var26 318-329 var27 330-341 var28 342
-353 var29 354-365
        var30 366-377 var31 378-389 var32 390-401 var33 402-413 var34 414-425 var35 426
-437 var36 438-449
        var37 450-461 var38 462-473 var39 474-485 var40 486-497 var41 498-509 var42 510
-521 var43 522-533
        var44 534-545 var45 546-557 var46 558-569 var47 570-581 var48 582-593 var49 594
-605 var50 606-617
        var51 618-629 var52 630-641 var53 642-653 var54 654-665 var55 666-677 var56 678
-689 var57 690-701
        var58 702-713 var59 714-725 var60 726-737 var61 738-749 var62 750-761 var63 762
-773 var64 774-785
        var65 786-797 var66 798-809 var67 810-821 var68 822-833 var69 834-845 var70 846
-857 var71 858-869
        var72 870-881 var73 882-893 var74 894-905 var75 906-917 var76 918-929;
run;

proc sort data=DFF_BE out=dff_be;
    by permno year1 year2;

proc transpose data=DFF_BE out=dff_be(rename=(col1=be));
    by permno year1 year2;
    var var1-var76;
run;

data dff_be(keep=be datadate permno inccm2a csho);
    format datadate date9.;
    set dff_be;
    retain inccm2a 0;
```

```
retain csho .;
_name_=substr(_name_, 4, length(_name_)-3);
year=1926+_name_-1;
if (year<=year2 and year>=year1);
if (be=-99.99) then be=.;
datadate=mdy(12,31,year-1);
if not(missing(be));
run;

data ccm2a(keep=permno be datadate inccm2a csho);
set ccm2a;
retain inccm2a 1;

proc append data=dff_be base=ccm2a;

proc sort data=ccm2a;
by permno datadate inccm2a;

data toprint;
set ccm2a;
by permno datadate inccm2a;
if not(first.datadate and last.datadate);
run;

%comprint(toprint, overlaps);

data ccm2a;
set ccm2a;
by permno datadate inccm2a;
if first.datadate; **DFF is authoritative;
run;

data ccm2a;
set ccm2a;
by permno datadate;
retain count;
if first.permno then count=1;
else count = count+1;
run;

proc sort data=ccm2a out=te nodupkey;
by permno datadate;
run;

/* Merge CRSP with Compustat data, at June of every year */
/* Match fiscal year ending calendar year t-1 with June t */

proc print data=ccm2a;
where permno=10436 and year(datadate)<=1966;
run;
data ccm2a;
set ccm2a;
putyear=year(datadate);
run;

proc sort data=ccm2a;
by permno putyear datadate;
run;
data ccm2a(drop=putyear);
set ccm2a;
by permno putyear datadate;
if last.putyear;
```

```
run;
proc print data=ccm2a;
  where permno=10436 and year(datadate)<=1966;
run;

proc sql;
  create table ccm2_june as
    select a.*, b.BE, 1000*b.BE/a.DEC_ME as BEME, b.count, b.csho
    from crsp_june as a , ccm2a as b
    where a.permno=b.permno
    and intnx('month',a.date,0,'E')=
    intnx('month',intnx('year',b.datadate,0,'E'),6,'E')
    order by date;
quit;

proc print data=ccm2_june;
  where permno=10436 and year(date)=1965;
run;

proc sort data=ccm2_june out=te nodupkey;
  by permno date;
run;

data ccm2_june;
  set ccm2_june;
  dec_me_mill=dec_me/1000;
  putative_me=decprc*csho;
run;

proc print data=ccm2_june(obs=500);
  where uniquepermco and csho*decprc>0;
run;

data ccm2_june(drop=csho uniquepermco decprc);
  set ccm2_june;
  used_be=be;
  used_me=dec_me_mill;
  if uniquepermco and csho*decprc>0 then do; *Use Compustat shrout if available and n
o confusion;
    beme=be/(decprc*csho);
    used_be=be;
    used_me=decprc*csho;
  end;
run;

/***** Part 4: Size and Book to Market Portfolios *****/
/* Initializing Portolio by ME and BEME as of each June t */
/* Calculate NYSE Size-Breakpoints for Market Equity ME */

data myh.cshopermnosjun_with_be_me;
  set ccm2_june;
run;

proc univariate data=ccm2_june noprint;
  where exchcd=1 and beme>0 and shrcd in (10,11) and me>0 and count>=2;
  var ME;
  by date; * at june;
  output out=nyse_me median = SIZEMEDN ;
run;
/* Calculate NYSE Size-Breakpoints for Book-to-Market BEME */
proc univariate data=ccm2_june noprint;
```

```

    where exchcd=1 and beme>0 and shrcd in (10,11) and me>0 and count>=2;
    var BEME;
    by date; /* at June */
    output out=nyse_beme pctlpts=30 70 pctlpre=BEME;
run;

/* Use Breakpoints to classify stock only at end of all June's */
proc sql;
    create table ccm3_june as
        select a.*, b.sizemedn, c.beme30, c.beme70
        from ccm2_june as a, nyse_me as b, nyse_beme as c
        where a.date=b.date=c.date;
quit;

/* Create portfolios as of June */
/* SIZE Portfolios : S[mall] or B[ig] */
/* Book-to-market Portfolios: L[ow], M[edium], H[igh] */
data june ;
    set ccm3_june;
    if beme>0 and me >0 and count>=2 then do;
        positivebeme=1; * beme>0 includes the restrictions that ME at Dec t-1 >0 and B
E t-1 >0 and
        more than two years in compustat;
        if ME <= sizemedn and ME>0 then SIZEPORT = 'S';
        else if ME > sizemedn then sizeport = 'B';
        else sizeport='';
        if beme <= beme30 and beme > 0 then BTMPORT = 'L';
        else if beme > beme30 and beme <= beme70 then BTMPORT = 'M' ;
        else if beme > beme70 then BTMPORT = 'H';
        else btmpport='';
        end;
    else positivebeme=0;
    if missing(sizeport)=0 and missing(btmpport)=0 then nonmissport=1;
    else nonmissport=0;
    keep permno date sizeport btmpport positivebeme exchcd shrcd nonmissport;
run;

/* Identifying each month the securities of */
/* Buy and hold June portfolios from July t to June t+1 */

proc sql;
    create table ccm4 as
        select a.*,
        b.sizeport, b.btmpport, b.date as portdate format date9., b.positivebeme , b.non
missport
        from crsp_m3 as a, june as b
        where a.permno=b.permno and intck('month',b.date,a.date) between 1 and 12
        order by date , sizeport, btmpport;
quit;

proc freq data=ccm4;
    tables date;
    title "XX";
proc freq data=ccm4;
    where weight_port>0 and retadj is not missing;
    tables date;
    title "XX2";
run;

/***** Part 5: Calculating Fama-French Factors *****/
/* Calculate monthly time series of weighted average portfolio returns */
proc means data=ccm4 noprint;

```

```
where weight_port>0 and positivebeme=1 and exchcd in (1,2,3) and shrccd in (10,11) a
nd nonmissport=1;
by date sizeport btmport;
var retadj;
weight weight_port;
output out=vwret(drop= _type_ _freq_ ) mean=vwret n=n_firms;
run;
```

```
/* Monthly Factor Returns: SMB and HML */
```

```
proc transpose data=vwret(keep=date sizeport btmport vwret) out=vwret2 (drop=_name_ _la
bel_);
by date ;
ID sizeport btmport;
Var vwret;
run;
```

```
/****** Part 6: Saving Output *****/
```

```
data myh.ff_factors_amnd_jbe_csho;
```

```
set vwret2;
WH = (bh + sh)/2 ;
WL = (sl + bl)/2 ;
WHML = WH - WL;
WB = (bl + bm + bh)/3 ;
WS = (sl + sm + sh)/3 ;
WSMB = WS - WB;
label WH = 'WRDS High'
      WL = 'WRDS Low'
      WHML = 'WRDS HML'
      WS = 'WRDS Small'
      WB = 'WRDS Big'
      WSMB = 'WRDS SMB';
```

```
run;
```

```
/* Number of Firms */
```

```
proc transpose data=vwret(keep=date sizeport btmport n_firms) out=vwret3 (drop=_name_ _
label_) prefix=n_;
by date ;
ID sizeport btmport;
Var n_firms;
run;
```

```
data myh.ff_nfirms_amnd_jbe_csho;
```

```
set vwret3;
N_H = n_sh + n_bh;
N_L = n_sl + n_bl;
N_HML = N_H + N_L;
N_B = n_bl + n_bm + n_bh;
N_S = n_sl + n_sm + n_sh ;
N_SMB = N_S + N_B;
Total1= N_SMB;
label N_H = 'N_firms High'
      N_L = 'N_firms Low'
      N_HML = 'N_firms HML'
      N_S = 'N_firms Small'
      N_B = 'N_firms Big'
      N_SMB = 'N_firms SMB';
```

```
run;
```

```
/* END */
```

```
/******
```