```
new;
cls;
d = ChangeDir( "c:\\alte\\finance\\programs\\gauss\\proj2v2");

external matrix __fmtnv;

library pgraph;
graphset;

oldmode = setvwrmode("many");

print "Select a Country: 1 = The USA, 2 = The UK, 3 = Switzerland : ";;
country = con(1,1);

if (country == 1);
  crty = "USA";
elseif (country == 2);
  crty = "UK";
elseif (country == 3);
  crty = "CH";
endif;


/* SETUP THE STARTING DATE  */
      print "Enter a starting date in MOYR format where MO is three letters ";
      print "and YR is two numbers (e.g. Dec01): ";;
      dt1 = cons;
      dt1 = upper(dt1);  /* dt1 is a string */
      mo = strsect(dt1,1,3);
      mo1 = 0$+mo;
      yr = stof(strsect(dt1,4,2));

/* READ THE COMPUSTAT DATA  */
/* THE RETURNS COVER ALL FIRMS AND MAY CONTAIN MISSING VALUES  */
      {xp,xbm,xmv,e} = readdata(country,dt1);

/* ELIMINATE THE FIRST TWO COLUMNS (FIRM IDS) FROM E */
   e = e-2;

/* CONVERT XP TO RATES OF RETURN  */
/* FIRST PUT IN ASCENDING ORDER WITH DATES ON VERTICAL AXIS  */
/* THE ROR, MW, BM MATRICES WILL START WITH DEC01 IN ROW 1 */

      ROR = rev(xp');
      mw = xmv';
      BM = xBM';
      ROR = (ROR-lag1(ROR))./lag1(ROR);
      ROR = REV(ROR);

/* TRANSPOSE, SO MATCH INITIAL FORMAT OF FIRM ROWS */
      MW = MW';
      BM = BM';
      ROR = ROR';

/* ADD FIRM IDS */
   ror = ror~seqa(1,1,rows(ror));
   xbm = xbm~seqa(1,1,rows(xbm));
   xmv = xmv~seqa(1,1,rows(xmv));
```

```
59    XP = MISS(XP,0);
60    XBM = MISS(XBM,0);
61    XMV = MISS(XMV,0);
62
63    /* READ THE MSCI DATA  */
64       msci = loadd("msci");
65       msci = msci[1:e,4:cols(msci)];
66
67 /* MSCI RATES OF RETURN  */
68    msci = rev(msci);
69       msci1 = lag1(msci);
70       rormsci = (msci - msci1)./msci1;
71       RORMSCI = REV(RORMSCI);        /* NOW ROW1 IS THE MOST RECENT */
72       RORMSCI = RORMSCI';
73    rormsci = rormsci[country,.];
74
75 /* READ THE RISK-FREE RATES OF RETURN */
76 /* c1 = MO, c2 = yr, c3 = US, c4 = UK, c5 = CH */
77       rf = loadd("rfree");
78       rf = rf[1:E,country+2];
79       rf = rf';
80 /* CONVERT TO MONTHLY RETURN */
81       rf = rf/12;
82    rf = rf/100;
83
84 /* DEFINE WEIGHTED COUNTRY RETURNS (E.G. S&P RETURNS) */
85    mw1 = packr(mw[.,1:e-1]~ror[.,1:e-1 cols(ror)]);  /* ADD FIRM NUMBER TO THE END */
86    FIRMS = MW1[.,COLS(MW1)];
87    sandp = mw1[.,E:(2*(e-1))];
88    mw1 = mw1[.,1:e-1];
89    mw1 = mw1[.,1:E-1]./sumc(mw1[.,1:E-1])';   /* firm shares */
90       sandp = sandp[.,1:E-1].*mw1[.,1:E-1];
91    sandp = packr(sandp);
92    sandp = sumc(sandp);
93
94  /* DEFINE THE BENCHMARK */
95    print;
96    print;
97    print "       Enter 1 to use MSCI as market returns.";
98    print "       Enter 2 to use country weighted returns as market returns : ";;
99    mkt = con(1,1);
100    if (mkt == 1);
101       string bench = "MSCI Market Returns";
102    elseif (mkt == 2);
103       bench = "MARKET RETURNS: "$+CRTY;
104       rormsci = sandp';
105    endif;
106    print;
107    print;
108
109 /* CALCULATE EXCESS RETURNS */
110    ermsci = rormsci[1:E-1]' - rf[1:E-1]';
111
112 /*Info combo and packing with breakpoints 1--e-1 | e--2*(e-1) | 2*e-1--3*(e-1) */
113    mbr=packr(mw[.,1:e-1]~bm[.,1:e-1]~ror[.,1:e-1]);
114
115 /*Initialize sought after vectors*/
116    smb={};hml={};r=rows(mbr);
```

```
117
118 /*start loop to proceed by month*/ /*originally set to e-1 but caused out of range error*/
119    for i(1,e-1,1);
120
121 /*each month stocks are sorted into two size portfolios using market values*/
122    sort1=sortc(mbr,i);/*size sorting*/
123    s=sort1[1:r/2,.];/*small stocks chosen*/
124    b=sort1[r-r/2+1:r,.];/* large stocks chosen*/
125
126 /*B/M sorting of size portfolios*/
127 /*small size decomposition*/
128    sort2=sortc(s,e+i-1);
129    sl=sort2[1:trunc(rows(s)*0.30),.];
130    sm=sort2[trunc(rows(s)*0.30)+1:trunc(rows(s)*0.70),.];
131    sh=sort2[trunc(rows(s)*0.70)+1:rows(s),.];
132
133 /*large size decomposition*/
134    sort3=sortc(b,e+i-1);
135    bl=sort3[1:trunc(rows(b)*0.30),.];
136    bm=sort3[trunc(rows(b)*0.30)+1:trunc(rows(b)*0.70),.];
137    bh=sort3[trunc(rows(b)*0.70)+1:rows(b),.];
138
139 /*market value weighting of returns*/
140 /*get relevant returns for each grouping*/
141 /*sl*/
142    slw=sl[.,i]./sumc(sl[.,i]);
143    slr=sl[.,2*e-1+i-1].*slw;
144    slr=sumc(slr);
145 /*sm*/
146    smw=sm[.,i]./sumc(sm[.,i]);
147    smr=sm[.,2*e-1+i-1].*smw;
148    smr=sumc(smr);
149 /*sh*/
150    shw=sh[.,i]./sumc(sh[.,i]);
151    shr=sh[.,2*e-1+i-1].*shw;
152    shr=sumc(shr);
153 /*bl*/
154    blw=bl[.,i]./sumc(bl[.,i]);
155    blr=bl[.,2*e-1+i-1].*blw;
156    blr=sumc(blr);
157 /*bm*/
158    bmw=bm[.,i]./sumc(bm[.,i]);
159    bmr=bm[.,2*e-1+i-1].*bmw;
160    bmr=sumc(bmr);
161 /*bh*/
162    bhw=bh[.,i]./sumc(bh[.,i]);
163    bhr=bh[.,2*e-1+i-1].*bhw;
164    bhr=sumc(bhr);
165
166  /*factor SMB*/
167  fs=(slr-blr+smr-bmr+shr-bhr)/3;
168  smb=smb|fs;
169
170  /*factor HML*/
171  fb=(shr-slr+bhr-blr)/2;
172  hml=hml|fb;
173
174  endfor;
```

```
175
176 /*coskewness*/
177
178    ermscil1 = trimr(ermsci,1,0);
179    ermsci = trimr(ermsci,0,1);
180    x = ones(rows(ermsci),1)~ermsci~ermscil1;
181    coskew={};
182    r=rows(mbr);
183
184
185 for i(1,r,1);
186    y=mbr[i,2*e-1:3*(e-1)-1];
187    b=y'/x;
188    resid=y'-x*b;
189    emsqr=(ermsci-meanc(ermsci))'(ermsci-meanc(ermsci));
190    eremsqr=sumc(resid.*emsqr);
191    ersqr=resid'resid;
192    eremsqr=eremsqr/r;
193    ersqr=ersqr/r;
194    emsqr=emsqr/r;
195    coskew=coskew|eremsqr/sqrt(ersqr)*emsqr;
196 endfor;
197
198    /*mbr=trimr(mbr,0,1); we shouldn't trim the number of firms but rather the number of months*/
199    /*now we need to sort the firms by coskewness for the entire period*/
200    /*Info combo and packing with breakpoints 1--e-1 | e--2*(e-1) | 2*e-1--3*(e-1) /cols(mbrs)*/
201    mbrs=mbr~coskew;
202    srt=sortc(mbrs,cols(mbrs));
203    smin=srt[1:trunc(rows(srt)*0.30),1:cols(srt)-1];
204    spls=srt[trunc(rows(srt)*0.70)+1:rows(srt),1:cols(srt)-1];
205
206    /*after having consumed coffee please proceed to get S- by subtracting corresponding returns on a monthly basis*/
207
208 w=smin[.,1:(e-1)]./sumc(smin[.,1:(e-1)])';
209 smin=smin[.,2*e-1:3*(e-1)].*w;
210 smin=sumc(smin);
211 w=spls[.,1:(e-1)]./sumc(spls[.,1:(e-1)])';
212 spls=spls[.,2*e-1:3*(e-1)].*w;
213 spls=sumc(spls);
214
215 sks=smin-spls;
216 smini=smin-rf[1,1:rows(smin)]';
217
218 /*now you have your factors, now please get portfolios created with b/m and skewness or coskewness each month*/
219
220    e2=trunc(rows(mbr)/5);
221    e2=0|e2|2*e2|3*e2|4*e2|rows(mbr); /*5 categories are created*/
222    sizeport={}; sizebmavg={};
223     /*sizeport=error(0).*ones(rows(5,e-1));*/
224 /*-----------------------------------------------------------------------*/
225 /*formation of regressee portfolios
226 5 size portfolios*/
227 for i(1,e-1,1);
228
229    sort=sortc(mbr,i); /*sorting on size*/
230
231    for k(1,5,1);
232       t1=sort[e2[k]+1:e2[k+1],1:e-1];  /*take out market weights for each segment*/
```

```
233       w=t1./sumc(t1)';
234       r=sort[e2[k]+1:e2[k+1],2*e-1:3*(e-1)];
235       t1=sumc(r.*w); /*weighted return corresponding to all category returns over entire time*/
236       t2=meanc(sort[e2[k]+1:e2[k+1], e:2*(e-1)]);
237       sizeport=sizeport|t1; /*yields size portfolios pondered by market weights*/
238       sizebmavg=sizebmavg|t2; /*yields average bm of portfolios which were sorted by market weights*/
239
240    endfor;  /*end counter for portfolios having been weighted by size*/
241
242 endfor; /*end counter for time periods*/
243
244 sizeport=reshape(sizeport,5,e-1);
245 sizebmavg=reshape(sizebmavg,5,e-1);
246 /*-----------------------------------------------------------------*/
247 /*formation of five B/M portfolios*/
248 bmport={}; bmbmavg={};
249 for i(1,e-1,1);
250
251    sort=sortc(mbr,e+i-1);   /*sorting on bm*/
252
253    for k(1,5,1);
254       t1=sort[e2[k]+1:e2[k+1],1:e-1];  /*take out market weights for each segment*/
255       w=t1./sumc(t1)';
256       r=sort[e2[k]+1:e2[k+1],2*e-1:3*(e-1)];
257       t1=sumc(r.*w); /*weighted return corresponding to all category returns over entire time*/
258       t2=meanc(sort[e2[k]+1:e2[k+1], e:2*(e-1)]);
259       bmport=bmport|t1;
260       bmbmavg=bmbmavg|t2;
261    endfor;
262
263 endfor;
264
265 bmport=reshape(bmport,5,e-1);
266 bmbmavg=reshape(bmbmavg,5,e-1);
267
268 /*obtain characteristics for portfolios in two different ways: for the portfolio B/M we need to go back to the mbr matrix,
269 perhaps above and get arithmetic averages of the  e--2*(e-1) columns (BM)
270 later we take the portfolio returns (perhaps above) and with a six month skewness that rolls we get necessary values, knowing th
271 regressions will have to be adjusted accordingly*/
272
273 /*CHARACTERISTICS - SKEWNESS AND COSKEWNESS FOR EACH PORTFOLIO FOR REGRESSIONS */
274
275 s_csk={}; b_csk={}; s_sk={}; b_sk={};
276 /*take members of five rows k, and then 6 month returns: 2*e-1:2*e-1+i+5*/
277
278 z=sizeport~bmport;
279
280    for m(1,e+1,e-1); /*this trick loop starts at 1 and then skips to e, which is where bmport begins*/
281       for k(1,5,1); /* we consider each portfolio based on size and bm rankings*/
282          for i(1,e-7,1); /* (e-7)-(e-1) yields a difference of six months*/
283             t3=z[k,m+i-1:m+i-1+6]; /*six months of return data for skewness and coskewness*/
284
285    /*WE ARE DEALING WITH SIZE RANKED PORTFOLIOS*/
286
287             if(m==1);
288             tt3=(meanc(t3')^3/stdc(t3')^3);  /*skewness measurement*/
289             s_sk=s_sk|tt3; /*skewness depot*/
290
```

```
/*Regression of 6 month skewness values on corresponding market returns and lagged market return*/
/*Somewhere adjustments need to be made for the fact that we are missing one value in the x
we take out lagged value and adjust x accordingly*/
        ermsci=rormsci[1:e-1]'-rf[1:e-1]';
        x=ones(rows(ermsci),1)~ermsci;
        xa=x[m+i-1:m+i-1+6,.];
        b=t3'/xa;

/*We get skewness information*/
        resid=t3'-xa*b;
/* ************MATRICES NOT COMPATIBLE PLEASE DIVIDE UP INTO SMALL PARTS AND TRY AGAIN */
a=ermsci[m+i-1:m+i-1+6,1];
 emsqr=(a-meanc(a))'(a-meanc(a));
    eremsqr=sumc(resid.*emsqr);
    ersqr=resid'resid;
    eremsqr=eremsqr/rows(x);
    ersqr=ersqr/rows(x);
    emsqr=emsqr/rows(x);
    s_csk=s_csk|eremsqr/sqrt(ersqr)*emsqr; /*skewness depot*/

        else; /*if m><1 and we are in bm territory*/
 /* ------------------------------------------------------------------------------ */
        tt3=(meanc(t3')^3/stdc(t3')^3);  /*skewness measurement*/
        b_sk=b_sk|tt3; /*skewness depot*/
        ermsci=rormsci[1:e-1]'-rf[1:e-1]';
        a=ermsci[i:i+6,1];
        x=ones(rows(ermsci),1)~ermsci;
        xa=x[i:i+6,.];
        b=t3'/xa;
        resid=t3'-xa*b;
       emsqr=(a-meanc(a))'(a-meanc(a));
        eremsqr=sumc(resid.*emsqr);
        ersqr=resid'resid;
        eremsqr=eremsqr/rows(x);
        ersqr=ersqr/rows(x);
        emsqr=emsqr/rows(x);
        b_csk=b_csk|eremsqr/sqrt(ersqr)*emsqr; /*skewness depot*/

        endif;
     endfor;
   endfor;
 endfor;

 s_sk=reshape(s_sk, 5, e-7);
 s_csk=reshape(s_csk, 5, e-7);
 b_sk=reshape(b_sk, 5, e-7);
 b_csk=reshape(b_csk, 5, e-7);

/*ORTHOGANIZATION*/

/*we take SMB, HML, and SKS, and SMINI and make SMO, HMO and CSKO by regressing factors separately
 on Rm-Rf (ermsci-newly defined */
/* SMB=const+ermsci*/

y=smb;
b=y/x;
resid=y-x*b;
smo=b[1]+resid;
```

NEEDS ADJUSTMENT

PRIOR TO REGRESSIONS

```
349
350 y=hml;
351 b=y/x;
352 resid=y-x*b;
353 hmo=b[1]+resid;
354
355 y=smini;
356 b=y/x;
357 resid=y-x*b;
358 csko=b[1]+resid;
359
360 y=sks;
361 b=y/x;
362 resid=y-x*b;
363 skso=b[1]+resid;
364
365 all=ermsci~smo~hmo~csko;
366 cx=corrx(all);
367
368 /*REGRESSIONS*/
369
370 /*TESTS*/
371
372
373
374
375
376
377
378
```

export

bmport