

```

/* library for Financial Econometrics */
/* common data library such as fama-french portfolios and factors */

libname ff5 "C:\FEdata";

libname fe "C:\FEdata";

libname Korea "C:\FEdata";

libname SKtests "C:\FEdata\tests";

/*
The complete codes for the course and the topic for idiosyncratic volatility
and can be modified to do the other two topics that are book-to-market ratio
and earnings to price ratio
*/

/* Reading Fama-French three factors (monthly series) */

PROC IMPORT OUT= FF5.FF3
            DATAFILE= "C:\FEdata\F-F_Research_Data_Factors.CSV"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;

/* reading Fama-French three factors (daily series) */
PROC IMPORT OUT= FF5.FF3D
            DATAFILE= "C:\FEdata\F-F_Research_Data_Factors_daily.CSV"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;

/* print out the data for review */

Proc Print data=FF5.FF3; run;

/* calculating the market return */

Data FF5.ff3m; set FF5.ff3;
Mkt=mkt_rf-rf;
Run;

/* print out the data for review */

Proc Print data=FF5.FF3m; run;

/* getting the simple statistics for market returns*/

Proc means data=ff5.FF3 noprint;
Var mkt_rf;
/*by year;*/
output out=ff5.FF3stat;
run;

/* finding correlation between factors */

```

```

Proc Corr data=FF5.FF3;
run;

/* finding the correlation between market return and value factor */

Proc Corr Data=FF5.FF3m;
Var Mkt;
With HML;
Run;

/* dropping variables inside a data */

Data FF5.mkt; set FF5.FF3m;
Drop HML SMB;
run;

Data FF5.mkt; set FF5.FF3m;
Keep date mkt;
run;

/* please note these two operations have different results! You should look
at it*/

/* extracting year and month in the data */

Data FF5.FF3new; set FF5.FF3;
year = int(date/100);
month= date - year*100;
run;

/* merge two data set */

Data FF5.merge; merge FF5.FF3m FF5.FF3new; by date;
Run;

/* extracting year month day from the date */
Data FF5.FF3Dnew; set FF5.FF3D;
year = int(date/10000);
month= int(date/100 - year*100);
day = date - year*10000 - month*100;
run;

/* read size and value 6 portfolios */
PROC IMPORT OUT= FF5.PF6
            DATAFILE= "C:\FEdata\6_Portfolios_2x3.CSV"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;

/* merge Fama French 3 factors with their 6 portfolios */
Data FF5.mergeFP; merge FF5.PF6 FF5.FF3m; by date;
p1x = p1-rf;
p2x = p2-rf;
p3x = p3-rf;
p4x = p4-rf;
p5x = p5-rf;

```

```

p6x = p6-rf;
Run;

/* run portfolio 1 excess return on the CAPM model */

/* the following is wrong code */

proc reg data=FF5.mergeFP outest=FF5.CAPM6out tableout;
model p1-rf = mkt_rf;
model p1x = mkt - rf;
run;quit;

/* the following is correct code */
proc reg data=FF5.mergeFP outest=FF5.CAPM6out tableout;
model p1x = mkt_rf;
run;quit;

/* stop here for lecture 4*/

/* run 6 portfolios on Fama-French three factor model */

proc reg data=FF5.mergeFP outest=FF5.FF6out tableout;
model p1x= mkt_rf hml smb;
/*
model p2x= mkt_rf hml smb;
model p3x= mkt_rf hml smb;
model p4x= mkt_rf hml smb;
model p5x= mkt_rf hml smb;
model p6x= mkt_rf hml smb;
*/
run;quit;

/* noprint for the regression */

proc reg data=FF5.mergeFP noprint outest=FF5.FF6out tableout;
model p1x= mkt_rf hml smb;
/*
model p2x= mkt_rf hml smb;
model p3x= mkt_rf hml smb;
model p4x= mkt_rf hml smb;
model p5x= mkt_rf hml smb;
model p6x= mkt_rf hml smb;
*/
run;quit;

/* finding the Newey-White (NW) adjusted t-stastic */

proc model data=FF5.mergeFP;
parms b0 b1 b2 b3;
p1x = b0+b1*mkt_rf+b2*hml+b3*smb;
fit p1x /gmm kernet = (bart,1,0);
instruments mkt_rf hml smb;
test mkt_rf, hml, smb;
run;quit;

```

```

/* assign a new data setname and store in the same directory */

libname Korea "C:\FEdata";

/* Read Market return data(daily series) */

PROC IMPORT OUT= KOREA.Mktd
            DATAFILE= "C:\KoreaReadSas\KoreaMktdaily.csv"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;

/* Read Market return data(monthly series) */

PROC IMPORT OUT= KOREA.Mktm
            DATAFILE= "C:\KoreaReadSas\KoreaMktmonth.csv"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;

data Korea.mktd;set Korea.mktd;
year = year(date);
month= month(date);
day = day(date);
run;

data Korea.mktm;set Korea.mktm;
year = year(date);
month= month(date);
day = day (date);
run;

/* sorting from early to latest */
proc sort data=Korea.Mktd;
by year month day;
run;

proc sort data=Korea.Mktm;
by year month day;
run;

/* read monthly percent per annum risk free rate downloaded from Federal
Reserve Bank of St Louis
webpage:
https://fred.stlouisfed.org/series/INTDSRKRM193N
*/

PROC IMPORT OUT= KOREA.RF
            DATAFILE= "C:\KoreaReadSas\Korearf.csv"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;

```

```
/* converting annual interest rate into monthly interest rate and daily
interest rate*/
```

```
data Korea.rf;set Korea.rf;
year = year(date);
month= month(date);
rfy = rfannual/100;
rfm = rfy/12;
rfd = rfy/365;
run;
```

```
/* reading stock's monthly return index (ri) */
```

```
PROC IMPORT OUT= Korea.ri_n
            DATAFILE= "C:\KoreaReadSas\Koreastockmonthri.csv"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
            GUESSINGROWS=10;
RUN;
```

```
proc contents data=Korea.ri_n;
run;
```

```
proc sort data=Korea.ri_n; by date;
run;
```

```
data ri_n; set Korea.ri_n;
_name_ = 'ri';
run;
```

```
proc transpose data=ri_n out=ri_n_t name=dscode;
by date;
run;
```

```
proc sort data=ri_n_t; by dscode; run;
```

```
data Korea.ri2018n_t; set ri_n_t;
run;
```

```
proc sort data=Korea.ri2018n_t; by dscode;
run;
```

```
data Korea.ri2018n_t; set Korea.ri2018n_t;
year = year(date);
month = month(date);
day = day(date);
if ri = . then delete;
lagri = lag(ri);
r = (ri-lag(ri))/lag(ri);
run;
```

```
/* cleaning the dscode */
```

```

data Korea.ri2018n_t; set Korea.ri2018n_t;
dscodel=substr(dscore,2,7);
run;

data Korea.ri2018n_t; set Korea.ri2018n_t;
drop dscore;
rename dscodel=dscore;
if r = . then delete;
run;

proc sort data=Korea.ri2018n_t; by dscore date;
run;

/* starting second observation because each stock's first observation is
lagged from previous stock's last obs */

data korea.return2018; set Korea.ri2018n_t;
by dscore;
if first.dscore then r =.;
logr=log(1+r);
run;

/* read stock's daily return index (ri) */

PROC IMPORT OUT= Korea.ridaily_n
            DATAFILE= "C:\KoreaReadSas\Koreastockdailyri.csv"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
            GUESSINGROWS=10;
RUN;

proc contents data=Korea.ridaily_n;
run;

proc sort data=Korea.ridaily_n; by date;
run;

data ridaily_n; set Korea.ridaily_n;
_name_ = 'ridaily';
run;

proc transpose data=ridaily_n out=ridaily_n_t name=dscore;
by date;
run;

proc sort data=ridaily_n_t; by dscore; run;

data Korea.ridaily2018n_t; set ridaily_n_t;
run;

proc sort data=Korea.ridaily2018n_t; by dscore;
run;

data Korea.ridaily2018n_t; set Korea.ridaily2018n_t;
year = year(date);

```

```

month = month(date);
day = day(date);
if ridaily = . then delete;
lagridaily = lag(ridaily);
rdaily = (ridaily-lag(ridaily))/lag(ridaily);
run;

/* cleaning the dscore */

data Korea.ridaily2018n_t; set Korea.ridaily2018n_t;
dscodel=substr(dscore,2,7);
run;

data Korea.ridaily2018n_t; set Korea.ridaily2018n_t;
drop dscore;
rename dscodel=dscode;
if rdaily = . then delete;
run;

proc sort data=Korea.ridaily2018n_t; by dscode date;
run;

/* starting second observation because each stock's first observation is
lagged from previous stock's last obs */

data korea.returndaily2018; set Korea.ridaily2018n_t;
by dscode;
if first.dscode then rdaily =.;
logrdialy=log(1+rdaily);
run;

/* assign a new directory to store the data for tests
but for university edition, you use the same directory /folder/myfolder */

libname SKtests "C:\FEdata\tests";

/* to keep it in the same directory */

/*
libname SKtests "C:\FEdata";
*/

/* data starts in 1987 */

data SKtests.mktd; set korea.mktd;
if year < 1987 then delete;
run;

proc sort data=SKtests.mktd;
by year month;
run;

data SKtests.rf; set korea.rf;
if year <1987 then delete;

```

```

proc sort data=Sktests.rf;
by year month;
run;

/* daily market return, rf and mkt excess return data */

data SKtests.mktrfd; merge SKtests.rf SKtests.mktfd;
by year month;
if mktr = . then delete;
mktdrx = mktr - rfd;
run;

proc sort data=Sktests.mktrfd;
by year month day;
run;

/* monthly market return, rf and mkt excess return data */

data SKtests.mktm; set Korea.Mktm;
run;

proc sort data=SKtests.mktm;
by year month;
run;

data Sktests.mktrfm; merge SKtests.mktm Sktests.rf;
by year month;
/* if mktr = . then delete;*/
mktmrxf = mktr - rfd;
run;

/* cleaning daily return */

data SKtests.returndaily; set korea.returndaily2018;
if rdaily = . then delete;
if year < 1987 then delete;
run;

proc sort data=SKtests.returndaily;
by year month day;
run;

proc sort data=Sktests.mktrfd;
by year month day;
run;

/* merge daily stock returns with daily market returns into a dataset */

data SKtests.returnmktfd; merge SKtests.returndaily SKtests.mktrfd;
by year month day;
rxdaily = rdaily - rfd;
run;

data SKtests.returnmktfd; set Sktests.returnmktfd;
if rxdaily = . then delete;

```



```

run;

/* calculate total volatility of each stock */

proc sort data=SKtests.returndaily;
by dscore year month day;
run;

proc means data=SKtests.returndaily noprint; by dscore year month; var
rdaily;
output out=SKtests.tvol std=tvol;run;

/* cleaning data if there is no trading for this stock on this month (tvol
=0)) */
data SKtests.tvol; set SKtests.tvol;
if tvol = 0 then delete;
drop _type_ _freq_;
run;

/* the following codes calculate the each stock's idiosyncratic volatility*/

data SKtests.regdaily; set SKtests.returnmkt;
index = year*12+month;
run;

proc sort data=SKtests.regdaily;
by dscore year month day;
run;

data lag1;set SKtests.regdaily;
run;

proc reg data=SKtests.regdaily noprint outest=lagmout tableout;
by dscore year month;
model rxdaily = mktdrx / r cli clm;
output out=residual p=pr r=idio;
run;quit;

proc means data=SKtests.residual noprint; by dscore year month; var idio;
output out=SKtests.idvol std=idvol;run;

data SKtests.idvol; set SKtests.idvol;
drop _type_ _freq_;
run;

/* merge total vol and idio vol into one data set */

proc sort data=SKtests.tvol; by dscore year month;
run;

proc sort data=SKtests.idvol; by dscore year month;
run;

data SKtests.stockvol; merge SKtests.tvol SKtests.idvol;

```

```

by dscore year month;
if tvol = . then delete; /* cleaning data */
run;

data SKtests.stockvol; set SKtests.stockvol;
lagtvol= lag(tvol);
lagidvol= lag(idvol);
run;

proc sort data=SKtests.stockvol; by dscore year month;
run;

data Sktests.stockvol;set Sktests.stockvol;
by dscore;
if first.dscore then lagtvol =.;
if first.dscore then lagidvol =.;
run;

proc sort data=korea.return2018; by dscore year month;
run; quit;

/* merge vol data with return data into one data set, then sort it by year
month and merge with rf*/
data SKtests.FMreg; merge korea.return2018 Sktests.stockvol;
if lagtvol = . then delete;
if lagidvol = . then delete;
run;quit;

proc sort data=Sktests.FMreg; by year month; run; quit;

data SKtests.FMreg2; merge Sktests.FMreg Sktests.rf; by year month;
rx = r-rfm;
if rx = . then delete;
if lagtvol = 0 then delete;
run;quit;

/* FAMA MacBath regressions */
proc reg data=SKtests.FMreg2 noprint outest=Sktests.lagmout tableout;
by year month;
model rx = lagidvol;
run;quit;

/* calculate the statistics */

data Sktests.lagmout2; set Sktests.lagmout;
if _type_ ^= "PARMS" then delete;
run;

proc means data=Sktests.lagmout2; var lagidvol;
output out=SKtests.FMidvol mean=idvol t=tstat;run;

/* the following codes are for sorting stocks into five portfolios and
calculating these portfolios' equal- and value-weighted returns */

/* reading the market value of each stock */

/* market value (size/mv) per month */

```

```

PROC IMPORT OUT= Korea.mv_n
            DATAFILE= "C:\KoreaReadSas\Koreastocksize.csv"
            DBMS=CSV REPLACE;
            GETNAMES=YES;
            DATAROW=2;
            GUESSINGROWS=10;
RUN;

proc contents data=Korea.mv_n;
run;

proc sort data=Korea.mv_n; by date;
run;

data mv_n; set Korea.mv_n;
      _name_ = 'mv';
run;

proc transpose data=mv_n out=mv_n_t name=dscode;
by date;
run;

proc sort data=mv_n_t; by dscode; run;

data Korea.mv2018n_t; set mv_n_t;
run;

proc sort data=Korea.mv2018n_t; by dscode;
run;

data Korea.mv2018n_t; set Korea.mv2018n_t;
year = year(date);
month= month(date);
day = day (date);
if mv = . then delete;
run;

/* cleaning the dscode */

data Korea.mv2018n_t; set Korea.mv2018n_t;
dscodel=substr(dscode,2,7);
run;

data Korea.mv2018n_t; set Korea.mv2018n_t;
drop dscode;
rename dscodel=dscode;
run;

/* merging return data, tvol, idiovol and mv into one data set */

proc sort data=SKtests.FMreg; by dscode year month; run; quit;

proc sort data=Korea.mv2018n_t; by dscode year month; run; quit;

data SKtests.FMregmv; merge SKtests.FMreg Korea.mv2018n_t;
by dscode year month;

```

```

/* delete missing observations */
if lagidvol = . then delete;
if mv = . then delete;
run;quit;

/* sort data and rank stocks into five portfolios by lagged idiovol*/

proc sort data=SKtests.FMregmv; by year month; run; quit;

proc rank data=SKtests.FMregmv out=SKtests.idvolrank groups=5; var lagidvol;
ranks idvolrank; by year month;run; quit;

/* note: rank starts from 0 to 4, rank 0 has lowest value of idvol and rank 4
has the highest value of idvol */

proc sort data=SKtests.idvolrank; by idvolrank year month; run; quit;

/* calculating equal-weighted return of each ranked portfolio */

proc means data=SKtests.idvolrank noprint; by idvolrank year month; var r;
output out=SKtests.idvolpret mean=avgr t=tstat;run;

proc means data=SKtests.idvolpret noprint; by idvolrank; var avgr;
output out=SKtests.idvolpretall mean=avgr t=tstat;run;

/* calculating value-weighted return of each ranked portfolios */

proc means data=SKtests.idvolrank noprint; by idvolrank year month; var r;
weight mv;
output out=SKtests.idvolpretv mean=avgr t=tstat;run;

proc means data=SKtests.idvolpretv noprint; by idvolrank; var avgr;
output out=SKtests.idvolpretvall mean=avgr t=tstat;run;

/* calculating the high minus low (4-0) ranked portfolio */

data Sktests.idvol0; set Sktests.idvolrank;
if idvolrank > 0 then delete;
run;quit;

data Sktests.idvol4; set Sktests.idvolrank;
if idvolrank < 4 then delete;
run;quit;

proc sort data=Sktests.idvol0; by year month;run; quit;
proc sort data=Sktests.idvol4; by year month;run; quit;

/* getting the equal- and value- weighted returns of each portfolio */

/* portfolio ranked 0 */
proc means data=Sktests.idvol0 noprint; by year month; var r;
output out=SKtests.idvol0pret mean=avgr0;run;

proc means data=Sktests.idvol0 noprint; by year month; var r; weight mv;
output out=SKtests.idvol0pretv mean=avgr0;run;

```

```

/* portfolio ranked 4 */

proc means data=Sktests.idvol4 noprint; by year month; var r;
output out=Sktests.idvol4pret mean=avgr4;run;

proc means data=Sktests.idvol4 noprint; by year month; var r; weight mv;
output out=Sktests.idvol4pretv mean=avgr4;run;

/* get the difference of returns of these two portfolios */

/* equal-weighted returns */
data Sktests.idvol40ret; merge Sktests.idvol0pret Sktests.idvol4pret;
by year month;
drop _type_ _freq_;
return40 = avgr4-avgr0;
run;quit;

/* value-weighted returns */
data Sktests.idvol40retv; merge Sktests.idvol0pretv Sktests.idvol4pretv;
by year month;
drop _type_ _freq_;
return40 = avgr4-avgr0;
run;quit;

/* the time-series average of equal- and value-weighted returns of the
difference of two portfolios*/

proc means data=Sktests.idvol40ret noprint; var return40;
output out=Sktests.idvol40pret mean=avgr40 t=tstat;run;quit;

proc means data=Sktests.idvol40retv noprint; var return40;
output out=Sktests.idvol40pretv mean=avgr40 t=tstat;run;quit;

```