library pgraph;

screen off;

output file=asstB.res reset;

outwidth 256;

format /rd 10,4;

/\*Sample Size: Daily=1258, Weekly=262, Monthly=60\*/

lags=10;

y=filesa("\*csv");

y\_rows= ROWS(y);

daily\_matrix=zeros((y\_rows/3),7);

weekly\_matrix=zeros((y\_rows/3),7);

monthly\_matrix=zeros((y\_rows/3),7);

monthly\_i=1;

weekly\_i=1;

daily\_i=1;

/\*creating a matrix of returns for daily, weekly and monthly data\*/

mor\_d=zeros(1247,(y\_rows/3));

mor\_w=zeros(251,(y\_rows/3));

mor\_m=zeros(49,(y\_rows/3));

file\_count=1;

do until file\_count>y\_rows;

file\_name=y[file\_count,1];

/\*loading the data set as a single string\*/

load data[]= ^file\_name;

T= rows(data)/8-1;

/\*Transforming the single string data set into a proper (T+1) by 8 data set\*/

data1=reshape(data,(T+1),8);

/\*Assigning the 'adjusted close' data - the data to be worked with\*/

adjclose= data1[(T+1):2,8];

/\*Fixing the Irrational number error when loading raw data from Yahoo finance\*/

adjclose=abs(adjclose);

logclose=ln(adjclose);

/\*net and log of return #ofreturns x 1 matrices\*/

net\_ret=(adjclose[2:T,1]-adjclose[1:(T-1),1])./adjclose[1:(T-1),1];

log\_ret=logclose[2:T,1]-logclose[1:(T-1),1];

/\*Means, Variance, Skewness, Kurtosis for Normality Test\*/

mean=meanc(net\_ret);

variance=vcx(net\_ret);

log\_mean=meanc(log\_ret);

log\_variance=vcx(log\_ret);

sumcubed=0;

sumquad=0;

logsumcubed=0;

logsumquad=0;

i=1;

do until i>(T-1);

sumcubed=sumcubed+(net\_ret[i,.]-mean)^3;

sumquad=sumquad+(net\_ret[i,.]-mean)^4;

logsumcubed=logsumcubed+(log\_ret[i,.]-log\_mean)^3;

logsumquad=logsumquad+(log\_ret[i,.]-log\_mean)^4;

i=i+1;

endo;

/\*Net Results - Normality Test\*/

skewness=sumcubed/((T-1)\*(variance^(3/2)));

kurtosis=sumquad/((T-1)\*(variance^2));

JB=((T-1)/6)\*(skewness^2+((kurtosis-3)^2)/4);

if JB < 5.99;

normality = "pass";

else;

normality = "fail";

endif;

/\*Log Results - Normality Test\*/

log\_skewness=logsumcubed/((T-1)\*log\_variance^(3/2));

log\_kurtosis=logsumquad/((T-1)\*log\_variance^2);

log\_JB=(T-1)\*(log\_skewness^2+((log\_kurtosis-3)^2)/4)/6;

if log\_JB < 5.99;

log\_normality = "pass";

else;

log\_normality = "fail";

endif;

/\*Random Walk 1:\*/

/\*Creating a 1247x11 matrix of returns rt, rt-1,..., rt-10\*/

r\_matrix=zeros(T-(lags+1),(lags+1));

m=1;

do until m>(lags+1);

r\_matrix[.,m]=log\_ret[(12-m):(T-m),1];

m=m+1;

endo;

cov\_matrix=vcx(r\_matrix);

/\*Making a 10x1 matrix of Pearson Coefficient Values:\*/

p\_matrix=zeros(lags,1);

z\_matrix=zeros(lags,1);

f=1;

do until f>lags;

p\_matrix[f,1]=cov\_matrix[1,(f+1)]/(cov\_matrix[1,1]^(1/2)\*cov\_matrix[(f+1),(f+1)]^(1/2));

z\_matrix[f,1]=p\_matrix[f,1]/((1/(T-11))^(1/2));

f=f+1;

endo;

/\*Box-Pierce Q-statistic\*/

Qm=zeros(lags,1);

Qm\_counter=1;

do until Qm\_counter>lags;

i=1;

do until i>Qm\_counter;

Qm[Qm\_counter,1]=Qm[Qm\_counter,1]+((T-11)\*(p\_matrix[i,1]^2));

i=i+1;

endo;

Qm\_counter=Qm\_counter+1;

endo;

/\*RW3 Testing\*/

VR=zeros((lags),1)+1;

q=1;

do until q>(lags);

k=1;

do until k>(q-1);

VR[q,1]=VR[q,1]+2\*(1-(k/(q-1)))\*p\_matrix[k,1];

k=k+1;

endo;

q=q+1;

endo;

/\* SD for VR... NOTE: k = q-1\*/

gamma\_tot = zeros((lags-1),1);

gamma\_top = zeros((lags-1),1);

gamma\_bottom = zeros((lags-1),1);

log\_mean2=meanc(log\_ret[1:(T-1),1]);

k=1;

do until k>(lags-1);

j=(k+1);

do until j>(T-1);

gamma\_top[k,1]=gamma\_top[k,1]+(T-1)\*((log\_ret[j,1]-log\_mean2)^2\*(log\_ret[(j-k),1]-log\_mean2)^2);

j=j+1;

endo;

j=1;

do until j>(T-1);

gamma\_bottom[k,1]=gamma\_bottom[k,1]+(log\_ret[j,1]-log\_mean2)^2;

j=j+1;

endo;

k=k+1;

endo;

gamma\_tot=gamma\_top./(gamma\_bottom^2);

theta=zeros((lags),1);

q=1;

do until q>(lags);

k=1;

do until k>(q-1);

theta[q,1]=theta[q,1]+gamma\_tot[k,1]\*(2\*(1-(k/q)))^2;

k=k+1;

endo;

q=q+1;

endo;

standardized=((T-1)^(1/2))\*(VR[3:lags,1]-1)./(theta[2:(lags-1)]^(1/2));

RW3\_test=zeros((lags-2),2);

i=1;

do until i>(lags-2);

if (standardized[i,1]^2)^(1/2)>1.96;

RW3\_test[i,1]="Reject";

RW3\_test[i,2]=1;

else;

RW3\_test[i,1]="RW3";

RW3\_test[i,2]=0;

endif;

i=i+1;

endo;

/\*Forecast Predictability based on VR result\*/

i=1;

predictability=0;

do until i>(lags-2);

if RW3\_test[i,2]>0;

predictability=i;

endif;

i=i+1;

endo;

/\*Assign returns based on predictability\*/

/\*filling in the matrix of returns for daily, weekly and monthly data\*/

rt\_plus1=zeros((T-lags-1),1);

i=1;

do until i>(T-lags-1);

rt\_plus1[i,1]=meanc(log\_ret[i:(i+lags),1]);

i=i+1;

endo;

rt=zeros((T-lags-1),lags);

if predictability>0;

i=1;

do until i>lags;

rt[.,i]=log\_ret[i:(T-1-(lags+1)+i),1];

i=i+1;

endo;

reg\_x=zeros((T-lags-1),predictability);

reg\_y=rt[.,2];

i=1;

do until i>(predictability);

reg\_x[.,i]=rt[.,(i+2)];

i=i+1;

endo;

{ vnam, m, b, stb,

vc, stderr, sigma,

cx, rsq, resid, dwstat } = ols(0,reg\_y,reg\_x);

i=1;

rt[.,1]=zeros(T-lags-1,1);

rt[.,1]=rt[.,1]+b[1,1];

do until i>predictability;

rt[.,1]=rt[.,1]+rt[.,(i+1)].\*b[(i+1),1];

i=i+1;

endo;

endif;

rt\_plus1[.,1]=exp(rt\_plus1)-1;

mean1=meanc(rt\_plus1[.,1]);

if predictability>0;

rt\_plus1[.,1]=exp(rt[.,1])-1;

rt1=meanc(rt[.,1]);

rt1=exp(rt1)-1;

else;

rt1=mean1;

endif;

if T<100;

mor\_m[.,monthly\_i]=rt\_plus1[.,1];

elseif T>400;

mor\_d[.,daily\_i]=rt\_plus1[.,1];

else;

mor\_w[.,weekly\_i]=rt\_plus1[.,1];

endif;

/\*To get result with RW assumption, 5th column in should be changed from “rt1” to “mean1”\*/

if T<100;

monthly\_matrix[monthly\_i,1]=file\_count;

monthly\_matrix[monthly\_i,2]=file\_name;

monthly\_matrix[monthly\_i,3]="Monthly";

monthly\_matrix[monthly\_i,4]=3;

monthly\_matrix[monthly\_i,5]=rt1;

monthly\_matrix[monthly\_i,6]=predictability;

monthly\_matrix[monthly\_i,7]=mean1;

monthly\_i=monthly\_i+1;

elseif T>499;

daily\_matrix[daily\_i,1]=file\_count;

daily\_matrix[daily\_i,2]=file\_name;

daily\_matrix[daily\_i,3]="Daily";

daily\_matrix[daily\_i,4]=1;

daily\_matrix[daily\_i,5]=rt1;

daily\_matrix[daily\_i,6]=predictability;

daily\_matrix[daily\_i,7]=mean1;

daily\_i=daily\_i+1;

else;

weekly\_matrix[weekly\_i,1]=file\_count;

weekly\_matrix[weekly\_i,2]=file\_name;

weekly\_matrix[weekly\_i,3]="Weekly";

weekly\_matrix[weekly\_i,4]=2;

weekly\_matrix[weekly\_i,5]=rt1;

weekly\_matrix[weekly\_i,6]=predictability;

weekly\_matrix[weekly\_i,7]=mean1;

weekly\_i=weekly\_i+1;

endif;

file\_count=file\_count+1;

endo;

/\*End of single file parsing\*/

/\*Start of portfolio building\*/

oldfmt = formatnv("\*.\*lf" ~ 8 ~ 4);

let mask[1,7]= 1 0 0 1 1 1 1;

d=printfmt(daily\_matrix,mask);

print;

d=printfmt(weekly\_matrix,mask);

print;

d=printfmt(monthly\_matrix,mask);

print;

call formatnv(oldfmt);

/\*Reminder: mor\_w, mor\_m, mor\_d for Matrix of net Returns - with different frequencies\*/

/\*Risk Free Returns: Daily, Weekly, Monthly\*/

rf\_d=1.017^(1/260)-1;

rf\_w=1.017^(1/52)-1;

rf\_m=1.017^(1/12)-1;

/\*------------------------------------------------------------\*/

/\* Daily: \*/

/\*------------------------------------------------------------\*/

Y=(1.2);

X=(1);

daily\_Y=zeros(15,4);

g=1;

do until g>15;

/\*Inverse of the Variance/covariance Matrix\*/

v\_ij=vcx(mor\_d);

v\_ij=inv(v\_ij);

Wpi=zeros((daily\_i-1),1);

a=zeros((daily\_i-1),1);

b=zeros((daily\_i-1),1);

c=zeros((daily\_i-1),1);

/\*Weight of stock i in portfolio p Wpi\*/

i=1;

do until i>(daily\_i-1);

j=1;

do until j>(daily\_i-1);

Wpi[i,1]=Wpi[i,1]+ v\_ij[i,j]\*(daily\_matrix[j,5]-rf\_d);

a[i,1]=a[i,1]+v\_ij[i,j]\*(daily\_matrix[j,5]);

b[i,1]=b[i,1]+v\_ij[i,j]\*(daily\_matrix[j,5])\*(daily\_matrix[i,5]);

c[i,1]=c[i,1]+v\_ij[i,j];

j=j+1;

endo;

i=i+1;

endo;

i=1;

positive\_sum=0;

negative\_sum=0;

do until i>rows(Wpi);

if Wpi[i,1]>0;

positive\_sum=positive\_sum+Wpi[i,1];

else;

negative\_sum=negative\_sum+Wpi[i,1];

endif;

i=i+1;

endo;

a=sumc(a);

b=sumc(b);

c=sumc(c);

D=(c\*rf\_d^2-2\*a\*rf\_d+b);

E= (rf\_d+(Y/X)\*D-D)/(1+D);

theta\_k=2\*(Y-X\*(1+E))/X;

/\*Solving for Weight of Risk Free Asset, E[Zp\*], Var[P\*] and RRA associated with them\*/

Wp0=1-(theta\_k/2)\*(a-c\*rf\_d);

epr=rf\_d+(theta\_k/2)\*D;

varp=(theta\_k/2)^2\*D;

RRA=(X\*(1+epr))/(Y-X\*(1+epr));

print "-------------------------------------------------------------------------";

print "Daily:";

print "+ve" positive\_sum\*(theta\_k/2);

print "-ve" negative\_sum\*(theta\_k/2);

print "Risky Asset Weights: " Wpi'.\*(theta\_k/2);

print;

print "Risky Sum: " sumc(Wpi.\*(theta\_k/2));

print "Risk Free: " Wp0;

print;

print "a b c: " a b c;

print "D E" D E;

print "Theta: " theta\_k;

print "E[Zp]= " epr;

print "Var[Zp]= " varp;

print "RRA = " RRA;

print;

daily\_Y[g,1]=epr;

daily\_Y[g,2]=varp;

daily\_Y[g,3]=RRA;

daily\_Y[g,4]=Y;

Y=Y+0.1;

g=g+1;

endo;

print " E[Zp] Var[Zp] RRA Y";

print daily\_Y;

/\*------------------------------------------------------------\*/

/\* Weekly: \*/

/\*------------------------------------------------------------\*/

Y=(1.5);

X=(1);

weekly\_Y=zeros(15,4);

g=1;

do until g>15;

v\_ij=vcx(mor\_w);

v\_ij=inv(v\_ij);

Wpi=zeros((weekly\_i-1),1);

a=zeros((weekly\_i-1),1);

b=zeros((weekly\_i-1),1);

c=zeros((weekly\_i-1),1);

/\*Weight of stock i in portfolio p Wpi\*/

i=1;

do until i>(weekly\_i-1);

j=1;

do until j>(weekly\_i-1);

Wpi[i,1]=Wpi[i,1]+ v\_ij[i,j]\*(weekly\_matrix[j,5]-rf\_w);

a[i,1]=a[i,1]+v\_ij[i,j]\*(weekly\_matrix[j,5]);

b[i,1]=b[i,1]+v\_ij[i,j]\*(weekly\_matrix[j,5])\*(weekly\_matrix[i,5]);

c[i,1]=c[i,1]+v\_ij[i,j];

j=j+1;

endo;

i=i+1;

endo;

a=sumc(a);

b=sumc(b);

c=sumc(c);

rra\_graph=zeros(15,3);

/\*Solving for E\_m = E[Zp] and then sub it in to solve for Theta \*/

D=(c\*rf\_w^2-2\*a\*rf\_w+b);

E= (rf\_w+(Y/X)\*D-D)/(1+D);

theta\_k=2\*(Y-X\*(1+E))/X;

/\*Solving for Weight of Risk Free Asset, E[Zp\*], Var[P\*] and RRA associated with them\*/

Wp0=1-(theta\_k/2)\*(a-c\*rf\_w);

epr=rf\_w+(theta\_k/2)\*D;

varp=(theta\_k/2)^2\*D;

RRA=(X\*(1+epr))/(Y-X\*(1+epr));

print "-------------------------------------------------------------------------";

print "Weekly:";

print "Risky Asset Weights: " Wpi'.\*(theta\_k/2);

print;

print "Risky Sum: " sumc(Wpi.\*(theta\_k/2));

print "Risk Free: " Wp0;

print;

print "a b c: " a b c;

print "D E" D E;

print "Theta: " theta\_k;

print "E[Zp]= " epr;

print "Var[Zp]= " varp;

print "RRA = " RRA;

print;

weekly\_Y[g,1]=epr;

weekly\_Y[g,2]=varp;

weekly\_Y[g,3]=RRA;

weekly\_Y[g,4]=Y;

Y=Y+0.2;

g=g+1;

endo;

print " E[Zp] Var[Zp] RRA Y";

print weekly\_Y;

/\*------------------------------------------------------------\*/

/\* Monthly: \*/

/\*------------------------------------------------------------\*/

Y=(1.5);

X=(1);

monthly\_Y=zeros(15,4);

g=1;

do until g>15;

v\_ij=vcx(mor\_m);

v\_ij=inv(v\_ij);

Wpi=zeros((monthly\_i-1),1);

a=zeros((monthly\_i-1),1);

b=zeros((monthly\_i-1),1);

c=zeros((monthly\_i-1),1);

/\*Weight of stock i in portfolio p Wpi\*/

i=1;

do until i>(monthly\_i-1);

j=1;

do until j>(monthly\_i-1);

Wpi[i,1]=Wpi[i,1]+ v\_ij[i,j]\*(monthly\_matrix[j,5]-rf\_m);

a[i,1]=a[i,1]+v\_ij[i,j]\*(monthly\_matrix[j,5]);

b[i,1]=b[i,1]+v\_ij[i,j]\*(monthly\_matrix[j,5])\*(monthly\_matrix[i,5]);

c[i,1]=c[i,1]+v\_ij[i,j];

j=j+1;

endo;

i=i+1;

endo;

a=sumc(a);

b=sumc(b);

c=sumc(c);

rra\_graph=zeros(15,3);

/\*Solving for E\_m = E[Zp] and then sub it in to solve for Theta \*/

D=(c\*rf\_m^2-2\*a\*rf\_m+b);

E= (rf\_m+(Y/X)\*D-D)/(1+D);

theta\_k=2\*(Y-X\*(1+E))/X;

/\*Solving for Weight of Risk Free Asset, E[Zp\*], Var[P\*] and RRA associated with them\*/

Wp0=1-(theta\_k/2)\*(a-c\*rf\_m);

epr=rf\_m+(theta\_k/2)\*D;

varp=(theta\_k/2)^2\*D;

RRA=(X\*(1+epr))/(Y-X\*(1+epr));

print "-------------------------------------------------------------------------";

print "Monthly:";

print "Risky Asset Weights: " Wpi'.\*(theta\_k/2);

print;

print "Risky Sum: " sumc(Wpi.\*(theta\_k/2));

print "Risk Free: " Wp0;

print;

print "a b c: " a b c;

print "D E" D E;

print "Theta: " theta\_k;

print "E[Zp]= " epr;

print "Var[Zp]= " varp;

print "RRA = " RRA;

print "-------------------------------------------------------------------------";

print;

monthly\_Y[g,1]=epr;

monthly\_Y[g,2]=varp;

monthly\_Y[g,3]=RRA;

monthly\_Y[g,4]=Y;

Y=Y+1;

g=g+1;

endo;

print " E[Zp] Var[Zp] RRA Y";

print monthly\_Y;