% =========================================================================

% BGG\_SSM.m — 2025‑07‑12 (final export: lagged betas + R² + Excel + PDF)

% =========================================================================

function BGG\_SSM(numLags, startDate, endDate)

if nargin < 1 || isempty(numLags), numLags = 4; end

if nargin < 2 || isempty(startDate),startDate = datetime(-Inf,1,1); end

if nargin < 3 || isempty(endDate), endDate = datetime( Inf,1,1); end

fprintf('[BGG\_SSM] lags = %d | window = %s → %s\n', numLags, string(startDate), string(endDate));

%% 1. LOAD DATA

DATA\_FILE = "PE\_data.xlsx";

opts = detectImportOptions(DATA\_FILE,'PreserveVariableNames',true);

TBL = sortrows(readtable(DATA\_FILE, opts), 'Date');

if isnumeric(TBL{1,'Date'})

dates = datetime(TBL{:,'Date'}, 'ConvertFrom','excel');

else

dates = TBL{:,'Date'};

end

peR = TBL{:,'PE - RF'};

F0 = [TBL{:,'Mkt-RF'}, TBL{:,'SMB'}, TBL{:,'HML'}, TBL{:,'Liq'}];

mask = (dates >= startDate) & (dates <= endDate);

[dates, peR, F0] = deal(dates(mask), peR(mask), F0(mask,:));

valid = all(~isnan([peR F0]), 2);

[dates, peR, F0] = deal(dates(valid), peR(valid), F0(valid,:));

[T, K] = size(F0);

allNames = ["Mkt","SMB","HML","LIQ"];

baseNames = allNames(1:K);

%% 2. RUN MAIN PIPELINE

[resultMain, tStatTable, finalBeta, rollR2] = runOneLag(peR, F0, baseNames, numLags);

% Print RMSE

fprintf('RMSE (in / OOS): %.4f | %.4f', resultMain.rmse\_in, resultMain.rmse\_oos);

% Console diagnostics

fprintf('\n--- Kalman Results ---\n');

fprintf('Lambda (λ̂) : %.3f\n', resultMain.lambda);

fprintf('--- Model Fit ---\n');

fprintf('RMSE (in / OOS): %.4f | %.4f', resultMain.rmse\_in, resultMain.rmse\_oos);

fprintf('\nAIC : %.2f\n', resultMain.AIC);

fprintf('BIC : %.2f\n', resultMain.BIC);

fprintf('Final Beta means :\n');

disp(varfun(@mean, finalBeta, 'OutputFormat', 'table'));

%% 3. EXPORT RESULTS

runStamp = datestr(now,'yyyymmdd\_HHMMSS');

OUT = sprintf('results\_%s', runStamp);

if ~isfolder(OUT), mkdir(OUT); end

FIG = fullfile(OUT,'figs'); if ~isfolder(FIG), mkdir(FIG); end

save(fullfile(OUT,'BGG\_SSM\_output.mat'), 'resultMain','tStatTable','finalBeta','numLags','startDate','endDate','rollR2','dates');

% Excel export

excelFile = fullfile(OUT,'BGG\_diagnostics.xlsx');

writetable(tStatTable, excelFile, 'Sheet','TStats');

writetable(finalBeta, excelFile, 'Sheet','FinalBeta');

writetable(table(dates, rollR2, 'VariableNames',{ 'Date','RollingR2'}), excelFile, 'Sheet','RollingR2');

summaryTbl = table;

summaryTbl.NumLags = numLags;

summaryTbl.Start = startDate;

summaryTbl.End = endDate;

summaryTbl.Lambda = resultMain.lambda;

summaryTbl.RMSE\_in = resultMain.rmse\_in;

summaryTbl.RMSE\_oos = resultMain.rmse\_oos;

summaryTbl.AIC = resultMain.AIC;

summaryTbl.BIC = resultMain.BIC;

writetable(summaryTbl, excelFile, 'Sheet','Summary');

%% 4. FIGURES

% T-statistics

figT = figure('Visible','off');

bar(tStatTable.tStat, 'FaceColor', [0.2 0.6 0.8]); hold on;

yline([-1.96 1.96], 'r--', 'LineWidth',1);

yline([-1.28 1.28], 'g--', 'LineWidth',1);

set(gca, 'XTick', 1:height(tStatTable), 'XTickLabel', tStatTable.Factor, 'XTickLabelRotation', 45);

ylabel('t-Statistic'); title(sprintf('Factor-lag t-stats (L=%d)', numLags));

grid on;

tstatFigPath = fullfile(FIG, sprintf('tstats\_L%d.png', numLags));

print(figT, tstatFigPath, '-dpng','-r150');

% Rolling R²

figR = figure('Visible','off');

plot(dates, rollR2, 'LineWidth',1.5); ylim([0 1]); grid on

xlabel('Date'); ylabel('R²'); title('Rolling R² (20 quarters)');

print(figR, fullfile(FIG,'rolling\_R2.png'), '-dpng','-r150');

% Dynamic Beta

figB = figure('Visible','off');

plot(dates, finalBeta.Variables, 'LineWidth',1.3);

legend(finalBeta.Properties.VariableNames, 'Location','eastoutside');

xlabel('Date'); ylabel('\beta\_t'); title('Dynamic Factor Loadings');

grid on;

print(figB, fullfile(FIG,'dynamic\_betas.png'), '-dpng','-r150');

% Export combined PDF (first plot only, placeholder)

pdfReport = fullfile(OUT, 'BGG\_Report.pdf');

exportgraphics(figT, pdfReport, 'ContentType','image');

close(figT);

fprintf('All diagnostics, Excel and figures saved to → %s', OUT);

end

function [res, tTbl, finalBeta, rollR2] = runOneLag(peR, F0, baseNames, numLags)

[T, K] = size(F0);

Ffull = []; facNames = strings(1,(numLags+1)\*K);

for lag = 0:numLags

if lag == 0

blk = F0;

suffix = "";

else

blk = [zeros(lag,K); F0(1:end-lag,:)];

suffix = "\_L" + string(lag);

end

Ffull = [Ffull blk];

facNames(1, lag\*K + (1:K)) = string(baseNames) + suffix;

end

first = numLags + 1;

ySig = peR(first:end);

XSig = Ffull(first:end,:);

mdlSig = fitlm(XSig,ySig,'Intercept',true);

tStats = mdlSig.Coefficients.tStat(2:end);

tTbl = table(facNames', tStats, 'VariableNames',{'Factor','tStat'});

keep = abs(tStats) >= 1.28;

Fuse = Ffull(:, keep); Kuse = nnz(keep);

if Kuse == 0, error('No significant factor at lag %d', numLags); end

Xols = [ones(T,1) Fuse]; beta0 = Xols \ peR;

state\_mu0 = beta0; state\_P0 = diag([0.1; ones(Kuse,1)]);

Q = diag([0.005^2 ; 0.05^2 \* ones(Kuse,1)]);

sigma2 = var(peR - Xols \* beta0);

function ll = LL(lambda)

x=state\_mu0; P=state\_P0; ll=0; prev=0;

for t = 1:T

P = P + Q; y = peR(t) - lambda\*prev; H = (1 - lambda)\*[1; Fuse(t,:)'];

S = H'\*P\*H + sigma2; K = P\*H/S; v = y - H'\*x;

x = x + K\*v; P = P - K\*H'\*P; ll = ll - 0.5\*(log(2\*pi\*S) + v^2/S);

prev = peR(t);

end

end

lambdas = 0:0.01:0.94; llvec = arrayfun(@LL, lambdas);

[~, ix] = max(llvec); lambda = lambdas(ix);

x = state\_mu0; P = state\_P0; resid = zeros(T,1); prev = 0;

finalB = zeros(T,Kuse);

for t = 1:T

P = P + Q; y = peR(t) - lambda\*prev; H = (1 - lambda)\*[1; Fuse(t,:)'];

S = H'\*P\*H + sigma2; K = P\*H/S; v = y - H'\*x;

x = x + K\*v; P = P - K\*H'\*P;

finalB(t,:) = x(2:end)';

yhat = lambda\*prev + (1 - lambda)\*(x(1) + x(2:end)' \* Fuse(t,:)');

resid(t) = peR(t) - yhat; prev = peR(t);

end

finalBeta = array2table(finalB, 'VariableNames', facNames(keep));

ROLL = 20;

rollR2 = NaN(T,1);

for i = ROLL:T

yy = peR(i-ROLL+1:i);

XX = [ones(ROLL,1) Fuse(i-ROLL+1:i,:)];

b = XX \ yy; yhat = XX \* b;

rollR2(i) = 1 - sum((yy - yhat).^2) / sum((yy - mean(yy)).^2);

end

split = floor(0.8 \* T);

rmse\_in = sqrt(mean(resid(1:split).^2));

rmse\_oos = sqrt(mean(resid(split+1:end).^2));

Kparam = 1 + 1 + Kuse + 1;

AIC = -2\*llvec(ix) + 2\*Kparam;

BIC = -2\*llvec(ix) + Kparam\*log(T);

res = struct('lambda', lambda, 'rmse\_in', rmse\_in, ...

'rmse\_oos', rmse\_oos, 'AIC', AIC, 'BIC', BIC);

end