Return predictability

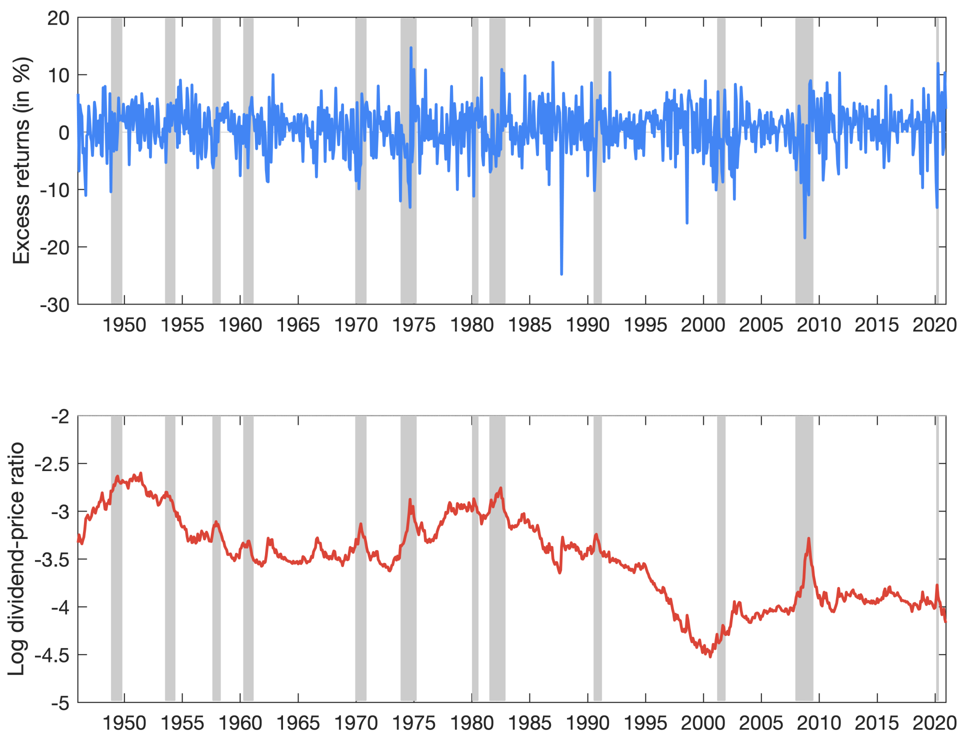
Table of Contents

Data

We construct the log dividend-price ratio using data available from Amit Goyal's website (https://sites.google.com/view/agoyal145), where you can download updated data from Goyal and Welch (2008). To plot the log dividend-price ratio against the National Bureau of Economic Research (NBER) business cycle dates, we collect the variable *USREC* on a monthly basis (in .csv format) from St. Louis Federal Reserve Economic Data (FRED) (https://fred.stlouisfed.org).

```
% Housekeeping
clear;
clc;

% Reading in monthly data from raw file: 1946.01 - 2020.12 (Post-war period)
gwData      = readmatrix('PredictorData2020','sheet','Monthly','range','a901:r1801');
recDates    = readmatrix('USREC.csv','range',[1095 2 1994 2]);

% Constructing log excess returns
riskFree    = log(1+gwData(1:end-1,11));    % Log risk-free rate
returnSP500 = log(1+gwData(2:end,17));      % Log stock market return
retExcess   = returnSP500 - riskFree;       % Log excess stock market return

% Constructing the log dividend-price ratio
sp500       = gwData(2:end,2);              % SP500 index
d12         = gwData(2:end,3);              % 12-month average of dividends
dpRatio     = log(d12./sp500);              % Log dividend-price ratio

% Setting number of observations
nObs        = size(retExcess,1);

% Setting datenum index
datenumIdx  = datenum(num2str(gwData(2:end,1)),'yyyymm');

% Plotting returns and dividend-price ratio
figure;
subplot(2,1,1);
hold on
b1 = bar(datenumIdx,recDates.*19.8);
b2 = bar(datenumIdx,-recDates.*29.8);
b1.EdgeColor = [0.8 0.8 0.8];
```

```
b1.FaceColor = [0.8 0.8 0.8];
b2.EdgeColor = [0.8 0.8 0.8];
b2.FaceColor = [0.8 0.8 0.8];
b1.ShowBaseLine = 'Off';
b2.ShowBaseLine = 'Off';
p1 = plot(datenumIdx,retExcess.*100);
p1.Color = colorBrewer(1);
p1.LineWidth = 1.4;
box on
datetick('x','yyyy');
axis([-inf inf -30 20]);
ylabel('Excess returns (in %)');
subplot(2,1,2);
hold on
b1 = bar(datenumIdx,recDates.*-4.98);
p1 = plot(datenumIdx,dpRatio);
b1.EdgeColor = [0.8 0.8 0.8];
b1.FaceColor = [0.8 0.8 0.8];
b1.ShowBaseLine = 'Off';
p1.Color = colorBrewer(2);
p1.LineWidth = 1.4;
box on
datetick('x','yyyy');
axis([-inf inf -5 -2]);
ylabel('Log dividend-price ratio');
```



In-sample predictability

We begin by exploring in-sample predictability using a standard predictive regression model of the form

$$r_{t+1} = \alpha + \beta(d_t - p_t) + \varepsilon_{t+1}$$

and we also run the following autoregressive regression for the log dividend-price ratio

$$d_{t+1} - p_{t+1} = \lambda + \rho(d_t - p_t) + \nu_{t+1}$$

which we will use later to conduct a parametric bootstrap on the significance of the slope paratemeter β in the predictive model for excess stocks returns.

```
% Estimating the predictive regression model
isRes   = nwRegress(retExcess(2:end,1),dpRatio(1:end-1,1),1,3);
isRes.bv
```

```
ans = 2×1
    0.0261
    0.0059
```

```
isRes.tbv
```

```
ans = 2×1
    2.3636
    1.8725
```

```
% Estimaing AR(1) model for the log dividend-price ratio
arDP    = nwRegress(dpRatio(2:end,1),dpRatio(1:end-1,1),1,3);
arDP.bv
```

```
ans = 2×1
   -0.0136
    0.9964
```

```
arDP.tbv
```

```
ans = 2×1
   -1.2015
  312.3386
```

Parametric bootstap

To examine if the in-sample predictability results are robust to finite sample bias, we consider a parametric bootstrap and simulate from the following system

$$\begin{bmatrix} \varepsilon_{t+1} \\ \nu_{t+1} \end{bmatrix} \sim \text{iid}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma \right), \text{ with } \Sigma = \begin{bmatrix} \sigma_\varepsilon^2 & \sigma_{\varepsilon\nu} \\ \sigma_{\nu\varepsilon} & \sigma_\nu^2 \end{bmatrix}$$

```
% Setting numbers for the bootstrap procedure
nBoot   = 9999;
bSim    = NaN(nBoot,1);
vProb   = zeros(nBoot,1);
errVar  = cov(isRes.resid,arDP.resid);

% Running nBoot number of bootstrap samples
for iBoot = 1:nBoot
```

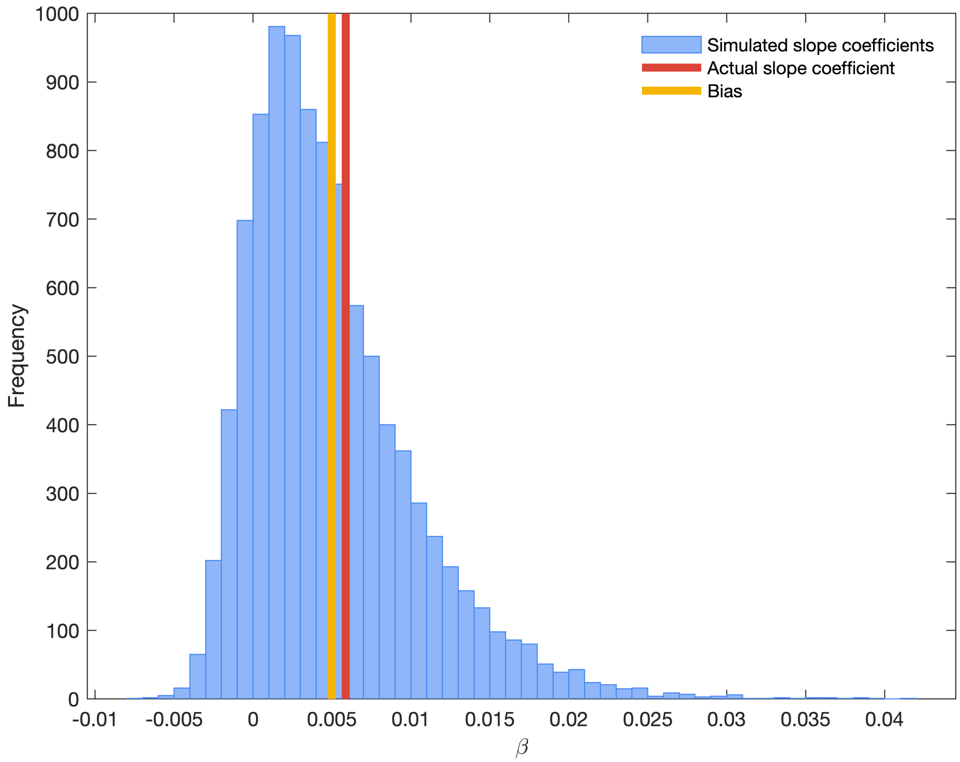
```
    % Drawing residuals from assumed distribution
    residSim        = mvnrnd([0 0],errVar,nObs);

    % Start the dividend-price ratio at its mean + noise
    dpRatioStart    = mean(dpRatio) + std(dpRatio)*randn(1);
    dpRatioSim      = zeros(nObs,1);
    dpRatioSim(1)   = dpRatioStart;

    % Simulating the dividend-price ratio
    for iObs = 2:nObs

        % Computing dp-ratio from AR(1) specification
        dpRatioSim(iObs,1)  = arDP.bv(1) + arDP.bv(2)*dpRatioSim(iObs-1,1) + residSim(

    end

    % Simulate returns
    returnSim       = isRes.bv(1)*ones(nObs-1,1) + residSim(2:end,1);

    % Running the auxiliary regression
    auxReg          = nwRegress(returnSim,dpRatioSim(1:end-1,1),1,3);
    bSim(iBoot,1)   = auxReg.bv(2);

    % Computing bootstrap p-values
    if ( auxReg.bv(2) > isRes.bv(2) )
        vProb(iBoot) = 1;
    end

end

% Display results
disp(['Bias in slope coefficient: ' num2str(mean(bSim))])
```

```
Bias in slope coefficient: 0.0049854
```

```
disp(['Bootstrapped p-value: ' num2str(mean(vProb))])
```

```
Bootstrapped p-value: 0.34423
```

```

% Display results in a figure
figure;
hold on;
p1 = histogram(bSim); % Make a histogram of the simulated slope coefficients
p1.FaceColor = colorBrewer(1);
p1.EdgeColor = colorBrewer(1);
line([isRes.bv(2) isRes.bv(2)], ylim, 'LineWidth',4, 'Color', colorBrewer(2));
line([mean(bSim) mean(bSim)], ylim, 'LineWidth',4, 'Color', colorBrewer(3));
xlabel('\beta');
ylabel('Frequency');
box on
leg = legend('Simulated slope coefficients','Actual slope coefficient','Bias');
set(leg,'Box','Off','Location','NorthEast');
```



Lewellen's correction

This section implements Lewellen's (2004) adjustment to the regression coefficient and t-statistic to test for return predictability using the log dividend-price ratio. We first compute the $\gamma = \frac{\sigma_{\varepsilon\nu}}{\sigma_\nu^2}$ coefficient from the regression $\varepsilon = \gamma\nu + \eta$ and then compute Lewellen's adjusted regression coefficient

$$\widehat{\beta}_{\mathrm{Adj.}} = \widehat{\beta} - \gamma[\widehat{\rho} - 1]$$

and then we can compute the adjusted variance as follows

$$\mathrm{Var}\left[\widehat{\beta}_{\mathrm{Adj.}}\right] = \frac{\sigma_\eta^2}{\left(T\sigma_x^2\right)}$$

```
% Computing the gamma regression coefficient
lewReg = nwRegress(isRes.resid,arDP.resid,0,3);

% Computed the bias-adjusted beta following Lewellen
betaLewellen    = isRes.bv(2) - lewReg.bv*(arDP.bv(2)-1)
```

```
betaLewellen = 0.0024
```

```
% Computing the t-statistic following Lewellen
tstatLewellen    = betaLewellen/sqrt(var(lewReg.resid,1)/(size(lewReg.resid,1)*var(dpRa
```

```
tstatLewellen = 4.0008
```

Out-of-sample predictability

Next, we consider an out-of-sample exercise in which we emulate a real-time forecaster. This is usually an important analysis of the validity of any proposed predictor. We consider here an initial window of $R = 240$ months and use an expanding window for the forecasting routine. We entertain both unrestricited forecasts (as in Goyal and Welch (2008)) and forecasts with economically motivated restrictions (as in Campbell and Thompson (2008)). The restrictions are that the predicted risk premium should be non-negative and that the coefficient β should follows its theoretical value, i.e., positive in our example for the log dividend-price ratio.

```
% Setting parameters of the environment
baseSample  = 240;
nFrcst      = nObs-baseSample;

% Preallocations prior to loop
actual      = NaN(nFrcst,1);
bench       = NaN(nFrcst,1);
retOOS      = NaN(nFrcst,1);
retVar      = NaN(nFrcst,1);
bOOS        = NaN(nFrcst,2);
sOOS        = NaN(nFrcst,2);

% Running out-of-sample forecasts
for iFrcst = 1:nFrcst

    % Estimating predictive regression model
    res                 = nwRegress(retExcess(2:baseSample+iFrcst-1,1),dpRatio(1:baseSa

    % Construct unrestricted forecasts
    retOOS(iFrcst,1)    = [1 dpRatio(baseSample+iFrcst-1,1)]*res.bv;

    % Construct restricted forecasts (Campbell and Thompson, 2008)
    retOOS(iFrcst,2)    = max(0,[1 dpRatio(baseSample+iFrcst-1,1)]*max(0,res.bv));

    % Actual realized returns
    actual(iFrcst,:)    = retExcess(baseSample+iFrcst,1);

    % Historical average benchmark
    bench(iFrcst,1)     = mean(retExcess(1:baseSample+iFrcst-1,1));

    % Rolling window variane for economic value
    retVar(iFrcst,1)    = std(retExcess(iFrcst:baseSample+iFrcst-1,1)).^2;

    % Saving estimated parameters and standard errors
    bOOS(iFrcst,:)      = res.bv;
    sOOS(iFrcst,:)      = res.sbv;

end
```

Statistical evaluation

We begin with a statistical evaluation of the forecasts using the log dividend-price ratio. We first compute the out-of-sample R^2 suggested in Campbell and Thompson (2008)

$$R^2_{OS} = 1 - \frac{\mathrm{MSFE}_x}{\mathrm{MSFE}_{HA}} = 1 - \frac{\sum_{i=R+1}^{T} \left(r_i - \widehat{r}_i\right)^2}{\sum_{i=R+1}^{T} (r_i - \overline{r}_i)^2}$$

and test for significance using the Diebold and Mariano (1995) and Clark and West (2007) tests. The first number represents the unrestricted forecasts, and the second number represents the forecasts with economic restrictions.

```
% Computing out-of-sample R2
R2oos   = 100*(1 - mean((retOOS-actual).^2)./mean((bench-actual).^2))
```

```
R2oos = 1×2
   -0.2844    0.1339
```

```
% Conducting Diebold-Mariano test
ft      = (bench-actual).^2 - (retOOS-actual).^2;
dmTest  = nwRegress(ft,ones(size(ft,1),1),0,3);
dmPval  = 1-normcdf(dmTest.tbv,0,1)
```

```
dmPval = 1×2
    0.6186    0.4318
```

```
% Conducting Clark-West test
ft      = (bench-actual).^2 - ((retOOS-actual).^2 - (bench-retOOS).^2 );
cwTest  = nwRegress(ft,ones(size(ft,1),1),0,3);
cwPval  = 1-normcdf(cwTest.tbv,0,1)
```

```
cwPval = 1×2
    0.1042    0.0602
```

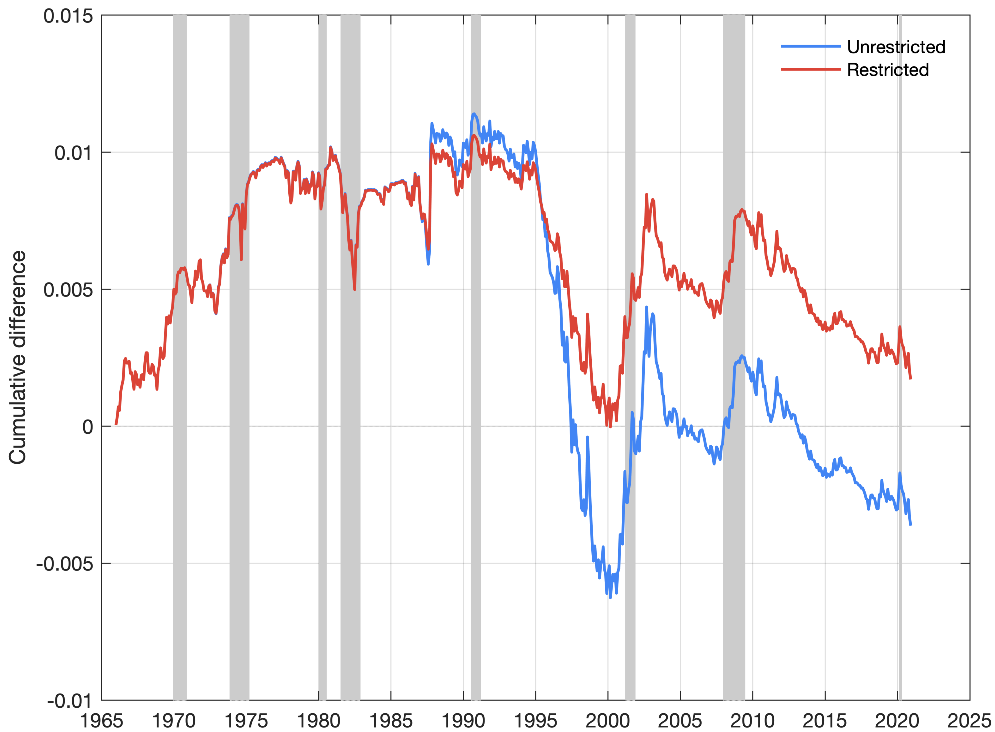
Finally, we can compute and plot the Goyal and Welch (2008) graphical device using the cumulative differences in squared forecast errors (CDSFE)

$$\mathrm{CDSFE}_t = \sum_{i=R+1}^{t} (r_i - \overline{r}_i)^2 - \sum_{i=R+1}^{t} \left(r_i - \widehat{r}_i\right)^2$$

This graphical device is a brilliant way to examine how stable the forecasts gains are over time, and can help rule out that the results are due to a few observations.

```
% Computing CDSFE
cdsfe = cumsum( (bench-actual).^2 ) - cumsum( (retOOS-actual).^2 );

% Plotting CDSFE
figure;
hold on
b1 = bar(datenumIdx(1+baseSample:end,1),recDates(1+baseSample:end,:).*0.015);
b2 = bar(datenumIdx(1+baseSample:end,1),-recDates(1+baseSample:end,:).*0.01);
```

```
b1.EdgeColor = [0.8 0.8 0.8];
b1.FaceColor = [0.8 0.8 0.8];
b2.EdgeColor = [0.8 0.8 0.8];
b2.FaceColor = [0.8 0.8 0.8];
b1.ShowBaseLine = 'Off';
b2.ShowBaseLine = 'Off';
p1 = plot(datenumIdx(1+baseSample:end,1),cdsfe);
p1(1).Color = colorBrewer(1);
p1(2).Color = colorBrewer(2);
p1(1).LineWidth = 1.4;
p1(2).LineWidth = 1.4;
hold off
datetick('x','yyyy');
ylabel('Cumulative difference');
box on
grid on
leg = legend(p1,'Unrestricted','Restricted');
set(leg,'Box','Off','Location','NorthEast');
```



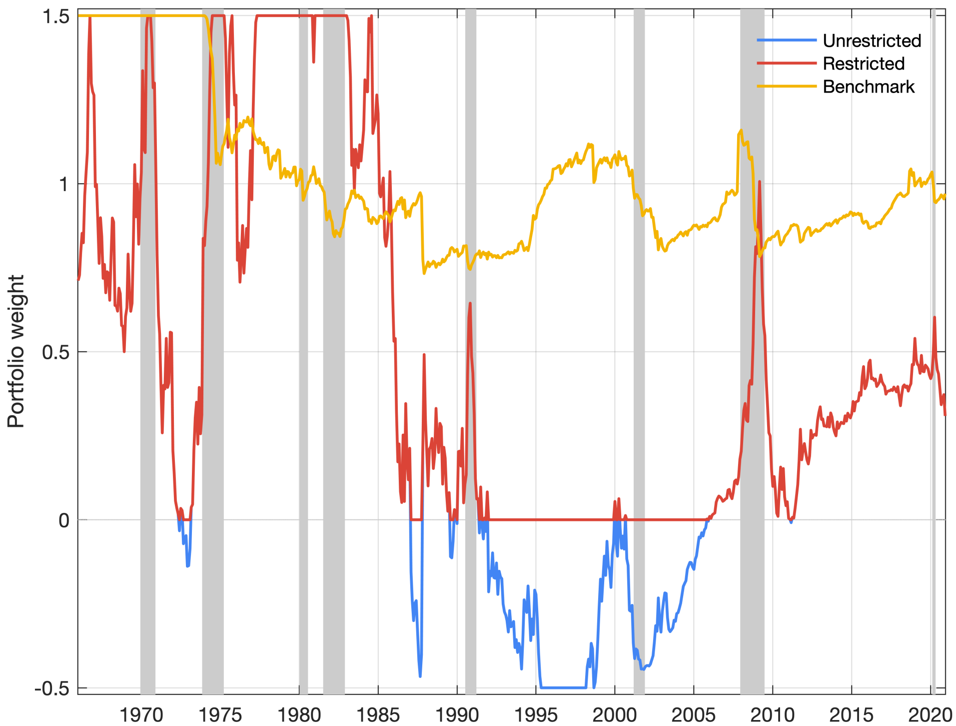
Economic evaluation

We next turn to an economic evaluation of the forecasts. We consider an investor with mean-variance preferences that allocates the fraction

$$\omega_t = \left(\frac{1}{\gamma}\right) \frac{\mathbb{E}_t[r_{t+1} - r_{f,t+1}]}{\mathrm{Var}[r_{t+1} - r_{f,t+1}]}$$

of her wealth to the risky asset. We set $\gamma = 3$ and compute the weights below and impose the restriction that we can at most short sell 50% and a maximum leverage of 50%. This prevents the investor from taking extreme positions.

```
% Computing risky asset weights
riskAversion    = 3;
wrisk           = [(retOOS./retVar)./riskAversion (bench./retVar)./riskAversion];

% Putting sensible bounds on the weights
wrisk(wrisk < -0.5) = -0.5;
wrisk(wrisk > 1.5)  = 1.5;

% Plotting portfolio weights
figure;
hold on
b1 = bar(datenumIdx(1+baseSample:end,1),recDates(1+baseSample:end,:).*1.55);
b2 = bar(datenumIdx(1+baseSample:end,1),-recDates(1+baseSample:end,:));
b1.EdgeColor = [0.8 0.8 0.8];
b1.FaceColor = [0.8 0.8 0.8];
b2.EdgeColor = [0.8 0.8 0.8];
b2.FaceColor = [0.8 0.8 0.8];
b1.ShowBaseLine = 'Off';
b2.ShowBaseLine = 'Off';
p1 = plot(datenumIdx(1+baseSample:end,1),wrisk);
p1(1).Color = colorBrewer(1);
p1(2).Color = colorBrewer(2);
p1(3).Color = colorBrewer(3);
p1(1).LineWidth = 1.4;
p1(2).LineWidth = 1.4;
p1(3).LineWidth = 1.4;
hold off
datetick('x','yyyy');
ylabel('Portfolio weight');
box on
grid on
axis([-inf inf -0.52 1.52]);
leg = legend(p1,'Unrestricted','Restricted','Benchmark');
set(leg,'Box','Off','Location','NorthEast');
```



Having found the optimal portfolio weights, we can compute portfolio returns as

$$r_{p,t+1} = (1-\omega_t)r_{f,t+1} + \omega_t r_{t+1} = r_{f,t+1} + \omega_t(r_{t+1} - r_{f,t+1})$$

```
% Setting the risk-free rate
riskFree         = riskFree(1+baseSample:end,1);

% Computing portfolio returns
pfReturns        = riskFree + wrisk.*actual;
```

Having obtained the resukting portfolio returns, we can compute the certainty equivalent return

$$\text{CER} = \mu_p - \frac{1}{2}\gamma\sigma_p^2$$

and the annualized utility gain as

$$\Delta = 1200 \times (\text{CER}_x - \text{CER}_{HA})$$

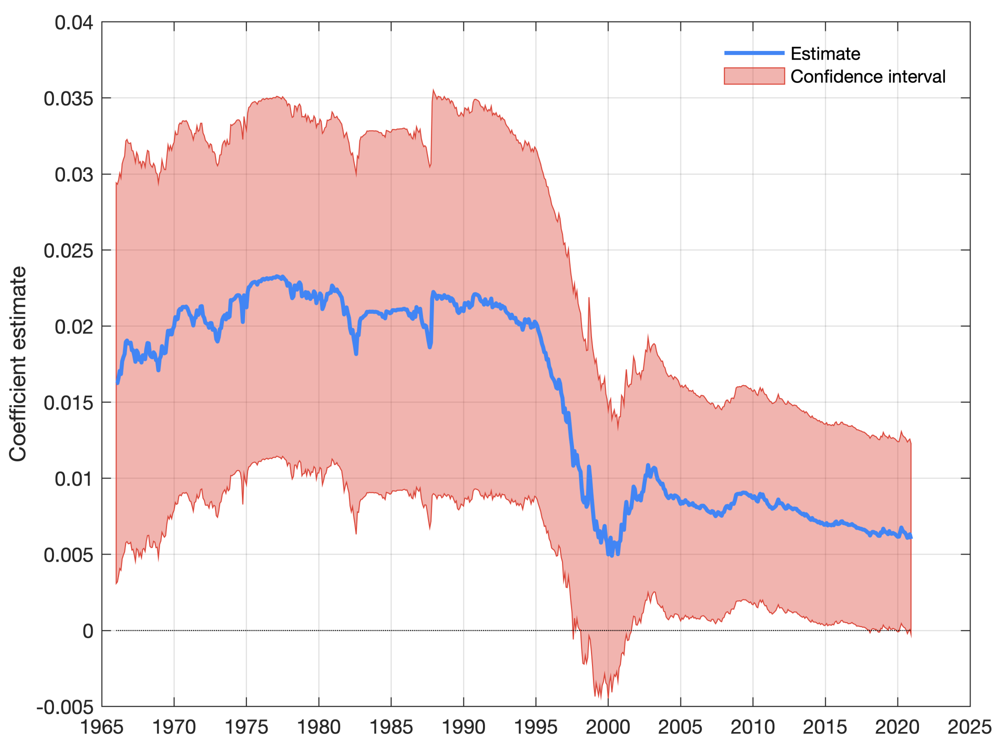
```
% Computing average utility and utility gains
utility          = mean(pfReturns) - 0.5*riskAversion*var(pfReturns);
uGain            = 1200*(utility(1:2) - utility(3))
```

```
uGain = 1×2
   -0.6577    0.0045
```

Instability of slope parameter

We can then study how the slope parameter changes over time in our out-of-sample forecasting environment to examine whether there are instabilities in the predictive relationship.

```
% Plotting slope parameter over time with confidence region
figure;
hold on
f1 = fill([flipud(datenumIdx(1+baseSample:end,1)); (datenumIdx(1+baseSample:end,1))],[f
p1 = plot(datenumIdx(1+baseSample:end,1),bOOS(:,2));
hold off
p1.Color = colorBrewer(1);
p1.LineWidth = 2;
f1.FaceAlpha = 0.4;
f1.LineStyle = '-';
f1.EdgeColor = colorBrewer(2);
h1 = line(datenumIdx(1+baseSample:end,1),zeros(size(bOOS,1),1));
h1.Color = 'k';
h1.LineStyle = ':';
datetick('x','yyyy');
ylabel('Coefficient estimate');
box on
grid on
leg = legend([p1 f1],'Estimate','Confidence interval');
set(leg,'Box','Off','Location','NorthEast');
```



Time-varying predictability

Last, we demonstrate that the predictive ability of the log dividend-price ratio appears to be concentrated in recession periods. We use the NBER recession indicator to partition predictability ex post. We start by creating indices for recessions and expansions, respectively.

```
%Computing recession/expansion index
recIdx      = recDates(241:end,1) == 1;
expIdx      = recDates(241:end,1) == 0;
```

We can then conducts a "state-dependent" evaluation of the forecasts by computing the out-of-sample R^2 for the full out-of-sample period (for reference) and for recessions and expansions separately. We can similarly test whether there is significant differences in predictive ability using the Diebold-Mariano and the Clark-West tests again.

```
% Computing out-of-sample R2 across states
R2oosALL    = 100*(1 - mean((retOOS-actual).^2)./mean((bench-actual).^2));
R2oosREC    = 100*(1 - mean((retOOS(recIdx,:)-actual(recIdx,:)).^2)./mean((bench(recId
R2oosEXP    = 100*(1 - mean((retOOS(expIdx,:)-actual(expIdx,:)).^2)./mean((bench(expId

% Conducting Diebold-Mariano test across states
ft          = (bench-actual).^2 - (retOOS-actual).^2;
dmTest      = nwRegress(ft,ones(size(ft,1),1),0,3);
dmPvalALL   = 1-normcdf(dmTest.tbv,0,1);

ft          = (bench(recIdx,:)-actual(recIdx,:)).^2 - (retOOS(recIdx,:)-actual(recIdx,
dmTest      = nwRegress(ft,ones(size(ft,1),1),0,3);
dmPvalREC   = 1-normcdf(dmTest.tbv,0,1);

ft          = (bench(expIdx,:)-actual(expIdx,:)).^2 - (retOOS(expIdx,:)-actual(expIdx,
dmTest      = nwRegress(ft,ones(size(ft,1),1),0,3);
dmPvalEXP   = 1-normcdf(dmTest.tbv,0,1);

% Conducting Clark-West test across states
ft          = (bench-actual).^2 - ((retOOS-actual).^2 - (bench-retOOS).^2 );
cwTest      = nwRegress(ft,ones(size(ft,1),1),0,3);
cwPvalALL   = 1-normcdf(cwTest.tbv,0,1);

ft          = (bench(recIdx,:)-actual(recIdx,:)).^2 - ((retOOS(recIdx,:)-actual(recIdx
cwTest      = nwRegress(ft,ones(size(ft,1),1),0,3);
cwPvalREC   = 1-normcdf(cwTest.tbv,0,1);

ft          = (bench(expIdx,:)-actual(expIdx,:)).^2 - ((retOOS(expIdx,:)-actual(expIdx
cwTest      = nwRegress(ft,ones(size(ft,1),1),0,3);
cwPvalEXP   = 1-normcdf(cwTest.tbv,0,1);
```

Last, we can conduct an economic evaluation from the perspective of a mean-variance investor and compute utility gains in recessions and expansions, respectively. Finally, we collect all results for time-varying predictability in a matrix and conclude that the log dividend-price ratio seems to be a stronger predictor during recession periods.

```
% Computing average utility and utility gains across states
utilityALL         = mean(pfReturns) - 0.5*riskAversion*var(pfReturns);
```

```
uGainALL            = 1200*(utilityALL(1:2) - utilityALL(3));

utilityREC          = mean(pfReturns(recIdx,:)) - 0.5*riskAversion*var(pfReturns(recIdx
uGainREC            = 1200*(utilityREC(1:2) - utilityREC(3));

utilityEXP          = mean(pfReturns(expIdx,:)) - 0.5*riskAversion*var(pfReturns(expIdx
uGainEXP            = 1200*(utilityEXP(1:2) - utilityEXP(3));

% Collecting results in matrix
stateDependet = [
    R2oosALL    R2oosEXP    R2oosREC
    dmPvalALL   dmPvalEXP   dmPvalREC
    cwPvalALL   cwPvalEXP   cwPvalREC
    uGainALL    uGainEXP    uGainREC
]
```

```
stateDependet = 4×6
   -0.2844    0.1339   -1.0833   -0.4827    1.6661    1.6394
    0.6186    0.4318    0.8171    0.6943    0.1149    0.1029
    0.1042    0.0602    0.2592    0.1833    0.0599    0.0546
   -0.6577    0.0045   -1.8991   -1.1251    7.4347    7.3528
```