```
function
 [TopPerformance,BottomPerformance,top_portfolio_statistics,bottom_portfolio_stati
 = ...

 PortfolioBacktest(dataset,year_start,year_end,window,quantiles,top_index_criteria

 rebalancing_intervall,index_variables,TopPerfModel,BottomPerfModel)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% This function backtests the portfolio performances of the share
% repruchase strategy for the top portolio and the bottom portfolio
 for the
% specified index thresholds.
%
% Inputs: Data, Starting Year, Ending Year, estimation window lenght,
% quantiles, top criteria, bottom criteria, rebalancing frequency,
 index
% variables, and performance evaluation models
%
% Ouput: Performance evaluation, portfolio statistics, and Time-Series
%
% Author: Sascha Jakob
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

% Time Period and Dataset/Variables
time_period = dataset.year>=(year_start-
(window/12))&dataset.year<=(year_end);
dataset = dataset(time_period,:);
dataset.time = dataset.time-min(dataset.time)-(window-1);
dataset = dataset(:, ismember(dataset.Properties.VariableNames,...

 {'month','time','dealnumber','year','permno','time','entropy_H',...

  'entropy_I','cash_atl1','ivol','top10instown_percl1','payout_yield_all',...

  'ret','rf','mktrf','smb','hml','umd','rmw','cma','ps_vwf','mkvalt',...

  'vol_shares_mean', 'vol_dollar_mean','ret_sd','buyback', 'retrf'}));

dataset.ret(isnan(dataset.ret)) = 0;
dataset.retrf(isnan(dataset.retrf))=0;

% Inital month
estimation_data = dataset.time>=(0-window+1)&dataset.time<=(0);
estimation_set = dataset(estimation_data,:);
quantile_data = estimation_set.buyback==1;
quantile_set = estimation_set(quantile_data,:);
quantile_set = quantile_set(:,
 ismember(quantile_set.Properties.VariableNames,
 {'dealnumber','year','permno','time','entropy_H',...
```

```
  'entropy_I','cash_atl1','ivol','top10instown_percl1','payout_yield_all'}));
return_sample  = dataset.time==0;
return_data = dataset(return_sample,:);

    quantile_set = sortrows(quantile_set, {'year','permno','time'});
        quantile_set.H_quantile =
 yearly_quantiles(quantile_set.entropy_H,quantiles,quantile_set.year);
        quantile_set.H_quantile(quantile_set.H_quantile==0)=NaN;
        quantile_set.I_quantile =
 yearly_quantiles(quantile_set.entropy_I,quantiles,quantile_set.year);
        quantile_set.I_quantile(quantile_set.I_quantile==0)=NaN;
        quantile_set.cash_quantile =
 yearly_quantiles(quantile_set.cash_atl1,quantiles,quantile_set.year);
        quantile_set.cash_quantile(quantile_set.cash_quantile==0)=NaN;
        quantile_set.ivol_quantile =
 yearly_quantiles(quantile_set.ivol,quantiles,quantile_set.year);
        quantile_set.ivol_quantile(quantile_set.ivol_quantile==0)=NaN;
        quantile_set.ownership_quantile =
 yearly_quantiles(quantile_set.top10instown_percl1,quantiles,quantile_set.year);

 quantile_set.ownership_quantile(quantile_set.ownership_quantile==0)=NaN;
        quantile_set.payout_quantile =
 yearly_quantiles(quantile_set.payout_yield_all,quantiles,quantile_set.year);

 quantile_set.payout_quantile(quantile_set.payout_quantile==0)=NaN;
    quantile_set = sortrows(quantile_set, {'permno','time'});

quantile_set = quantile_set(:,
 ismember(quantile_set.Properties.VariableNames,
 {'dealnumber','permno',...

  'H_quantile','I_quantile','cash_quantile','ivol_quantile','ownership_quantile','

estimation_set = outerjoin(estimation_set,quantile_set,'Keys',
{'permno','dealnumber'},'MergeKeys',true);
estimation_set.ownership_quantile =
 estimation_set.ownership_quantile.*(-1)+(quantiles+1);

  estimation_set.index =
 sum(table2array(estimation_set(:,ismember(estimation_set.Properties.VariableNames
    index_variables))),2);

top_portfolio_data = estimation_set(estimation_set.index
 >=top_index_criteria,:);
top_portfolio_data = sortrows(top_portfolio_data, {'permno','time'});
top_stocks = unique(top_portfolio_data.permno); % Extract permno of
 top stocks to invest during the next quarter

bottom_portfolio_data = estimation_set(estimation_set.index
 <=bottom_index_criteria,:);
bottom_portfolio_data = sortrows(bottom_portfolio_data,
 {'permno','time'});
```

```
bottom_stocks = unique(bottom_portfolio_data.permno); % Extract permno
 of bottom stocks to invest during the next quarter

portfolio_ret = [unique(return_data.month) 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0];

% Next Period weights
return_sample = dataset.time==1;
return_data = dataset(return_sample,:);

    top_portfolio_data =
 return_data(ismember(return_data.permno,top_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));
    top_stocks_w = zeros(height(top_portfolio_data),1);
    top_stocks_w(:) = 1/height(top_portfolio_data);

    bottom_portfolio_data =
 return_data(ismember(return_data.permno,bottom_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));
    bottom_stocks_w = zeros(height(bottom_portfolio_data),1);
    bottom_stocks_w(:) = 1/height(bottom_portfolio_data);

% subsequent months
for i = 1:max(dataset.time)

    if mod(i,rebalancing_intervall)==0

    estimation_data = dataset.time>=(i-window+1)&dataset.time<=(i);
    estimation_set = dataset(estimation_data,:);
    quantile_data = estimation_set.buyback==1;
    quantile_set = estimation_set(quantile_data,:);
    quantile_set = quantile_set(:,
 ismember(quantile_set.Properties.VariableNames,
 {'dealnumber','year','permno','time','entropy_H',...

  'entropy_I','cash_atl1','ivol','top10instown_percl1','payout_yield_all'}));
    return_sample = dataset.time==i;
    return_data = dataset(return_sample,:);

    top_portfolio_data =
 return_data(ismember(return_data.permno,top_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));

    bottom_portfolio_data =
 return_data(ismember(return_data.permno,bottom_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));
```

```
       if height(top_portfolio_data)==length(top_stocks_w)
           top_stocks_w = top_stocks_w;
       else
            top_stocks_w = zeros(height(top_portfolio_data),1);
            top_stocks_w(:) = 1/height(top_portfolio_data);
       end

       if height(bottom_portfolio_data)==length(bottom_stocks_w)
            bottom_stocks_w = bottom_stocks_w;
       else
            bottom_stocks_w = zeros(height(bottom_portfolio_data),1);
           bottom_stocks_w(:) = 1/height(bottom_portfolio_data);
       end

   temp_ret = [ unique(return_data.month)
unique(return_data.time) top_stocks_w'*top_portfolio_data.ret
bottom_stocks_w'*bottom_portfolio_data.ret ...
       mean(top_portfolio_data.mktrf + top_portfolio_data.rf)
length(top_portfolio_data.ret) length(bottom_portfolio_data.ret)
top_stocks_w'*top_portfolio_data.retrf...
       bottom_stocks_w'*bottom_portfolio_data.retrf
mean(top_portfolio_data.mktrf) mean(top_portfolio_data.smb)
mean(top_portfolio_data.hml)...
       mean(top_portfolio_data.umd) mean(top_portfolio_data.rmw)
mean(top_portfolio_data.cma) mean(top_portfolio_data.ps_vwf)...
       mean(top_portfolio_data.rf)];

   portfolio_ret = [portfolio_ret; temp_ret]; %#ok<AGROW>

   quantile_set = sortrows(quantile_set, {'year','permno','time'});
       quantile_set.H_quantile =
yearly_quantiles(quantile_set.entropy_H,quantiles,quantile_set.year);
       quantile_set.H_quantile(quantile_set.H_quantile==0)=NaN;
       quantile_set.I_quantile =
yearly_quantiles(quantile_set.entropy_I,quantiles,quantile_set.year);
       quantile_set.I_quantile(quantile_set.I_quantile==0)=NaN;
       quantile_set.cash_quantile =
yearly_quantiles(quantile_set.cash_atl1,quantiles,quantile_set.year);
       quantile_set.cash_quantile(quantile_set.cash_quantile==0)=NaN;
       quantile_set.ivol_quantile =
yearly_quantiles(quantile_set.ivol,quantiles,quantile_set.year);
       quantile_set.ivol_quantile(quantile_set.ivol_quantile==0)=NaN;
       quantile_set.ownership_quantile =
yearly_quantiles(quantile_set.top10instown_percl1,quantiles,quantile_set.year);

quantile_set.ownership_quantile(quantile_set.ownership_quantile==0)=NaN;
       quantile_set.payout_quantile =
yearly_quantiles(quantile_set.payout_yield_all,quantiles,quantile_set.year);

quantile_set.payout_quantile(quantile_set.payout_quantile==0)=NaN;
   quantile_set = sortrows(quantile_set, {'permno','time'});
```

```
    quantile_set = quantile_set(:,
 ismember(quantile_set.Properties.VariableNames,
 {'dealnumber','permno',...

  'H_quantile','I_quantile','cash_quantile','ivol_quantile','ownership_quantile','

    estimation_set = outerjoin(estimation_set,quantile_set,'Keys',
{'permno','dealnumber'},'MergeKeys',true);

    estimation_set.ownership_quantile =
 estimation_set.ownership_quantile.*(-1)+(quantiles+1);

   estimation_set.index =
 sum(table2array(estimation_set(:,ismember(estimation_set.Properties.VariableNames
    index_variables))),2);

    top_portfolio_data = estimation_set(estimation_set.index
 >=top_index_criteria,:);
    top_portfolio_data = sortrows(top_portfolio_data,
 {'permno','time'});
    top_stocks = unique(top_portfolio_data.permno); % Extract permno
 of top stocks to invest during the next quarter

    bottom_portfolio_data = estimation_set(estimation_set.index
 <=bottom_index_criteria,:);
    bottom_portfolio_data = sortrows(bottom_portfolio_data,
 {'permno','time'});
    bottom_stocks = unique(bottom_portfolio_data.permno); % Extract
 permno of bottom stocks to invest during the next quarter

    % Next Period weights
    return_sample = dataset.time==i+1;
    return_data = dataset(return_sample,:);

    top_portfolio_data =
 return_data(ismember(return_data.permno,top_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));
    top_stocks_w = zeros(height(top_portfolio_data),1);
    top_stocks_w(:) = 1/height(top_portfolio_data);

    bottom_portfolio_data =
 return_data(ismember(return_data.permno,bottom_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));
    bottom_stocks_w = zeros(height(bottom_portfolio_data),1);
    bottom_stocks_w(:) = 1/height(bottom_portfolio_data);

    else
    return_sample = dataset.time==i;
    return_data = dataset(return_sample,:);
    top_portfolio_data =
 return_data(ismember(return_data.permno,top_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));

    bottom_portfolio_data =
 return_data(ismember(return_data.permno,bottom_stocks),...
    ismember(return_data.Properties.VariableNames,
 {'ret' 'retrf' 'mktrf' 'rf'...
    'smb' 'hml' 'umd' 'cma' 'rmw' 'ps_vwf'}));

    if height(top_portfolio_data)==length(top_stocks_w)
       top_stocks_w = top_stocks_w; %#ok<*ASGSL>
    else
       top_stocks_w = zeros(height(top_portfolio_data),1);
       top_stocks_w(:) = 1/height(top_portfolio_data);
    end

    if height(bottom_portfolio_data)==length(bottom_stocks_w)
       bottom_stocks_w = bottom_stocks_w;
    else
       bottom_stocks_w = zeros(height(bottom_portfolio_data),1);
       bottom_stocks_w(:) = 1/height(bottom_portfolio_data);
    end

         temp_ret = [ unique(return_data.month)
 unique(return_data.time) top_stocks_w'*top_portfolio_data.ret
 bottom_stocks_w'*bottom_portfolio_data.ret...
              mean(top_portfolio_data.mktrf + top_portfolio_data.rf)
 length(top_portfolio_data.ret) length(bottom_portfolio_data.ret)
 top_stocks_w'*top_portfolio_data.retrf...
              bottom_stocks_w'*bottom_portfolio_data.retrf
 mean(top_portfolio_data.mktrf) mean(top_portfolio_data.smb)
 mean(top_portfolio_data.hml)...
              mean(top_portfolio_data.umd)
 mean(top_portfolio_data.rmw) mean(top_portfolio_data.cma)
 mean(top_portfolio_data.ps_vwf)...
              mean(top_portfolio_data.rf)];

          portfolio_ret = [portfolio_ret; temp_ret];  %#ok<AGROW>

          % Next period normalized weights due to no rebalancing
             top_stocks_w = top_stocks_w.*(1+top_portfolio_data.ret);
             top_stocks_w = normalize(top_stocks_w,'norm',1);
             bottom_stocks_w =
 bottom_stocks_w.*(1+bottom_portfolio_data.ret);
             bottom_stocks_w = normalize(bottom_stocks_w,'norm',1);
    end
end

 portfolio_ret = portfolio_ret(2:end,:);
 portfolio_returns = cell2table(portfolio_ret);
 portfolio_returns.Properties.VariableNames =
 {'Month' 'Time' 'TopReturn' 'BottomReturn' 'MarketReturn' 'TopObs' 'BottomObs' 'T
     'BottomExcess' 'MKTRF' 'SMB' 'HML' 'UMD' 'RMW' 'CMA' 'LIQ' 'Rf'
 };

 TopPerformance = fitlm(portfolio_returns,TopPerfModel);
 IR_top = (TopPerformance.Coefficients{1,1}/sqrt(TopPerformance.SSE/
TopPerformance.NumObservations))*sqrt(12);
 TopPerformance = TopPerformance.Coefficients;

 BottomPerformance = fitlm(portfolio_returns,BottomPerfModel);
 IR_bottom = (BottomPerformance.Coefficients{1,1}/
sqrt(BottomPerformance.SSE/
BottomPerformance.NumObservations))*sqrt(12);
 BottomPerformance = BottomPerformance.Coefficients;

 top_portfolio_statistics
 =array2table([mean(portfolio_returns.TopReturn) ...
                    std(portfolio_returns.TopReturn) ...
                    min(portfolio_returns.TopReturn)...
                    (mean(portfolio_returns.TopExcess)/
std(portfolio_returns.TopExcess))*sqrt(12)...
                    IR_top]);

 top_portfolio_statistics.Properties.VariableNames =
 {'Mean' 'Volatility' 'MaxDrawdown' 'SharpeRatio' 'InformationRatio'};

 bottom_portfolio_statistics
 =array2table([mean(portfolio_returns.BottomReturn) ...
                    std(portfolio_returns.BottomReturn) ...
                    min(portfolio_returns.BottomReturn)...
                   (mean(portfolio_returns.BottomExcess)/
std(portfolio_returns.BottomExcess))*sqrt(12)...
                   IR_bottom]);

 bottom_portfolio_statistics.Properties.VariableNames =
 {'Mean' 'Volatility' 'MaxDrawdown' 'SharpeRatio' 'InformationRatio'};


 plot_data =
 portfolio_returns(:,ismember(portfolio_returns.Properties.VariableNames,
{'Month' 'Time' 'TopReturn' 'BottomReturn' 'MarketReturn' 'TopObs' 'BottomObs' 'Rf
 plot_data.MarketReturn = plot_data.MarketReturn+plot_data.Rf;
 plot_data = [{year_start,0, 1, 1, 1, 0, 0, 0};plot_data];
 plot_data.TopReturn(2:end) = cumprod(plot_data.TopReturn(2:end)+1);
 plot_data.BottomReturn(2:end) =
 cumprod(plot_data.BottomReturn(2:end)+1);
 plot_data.MarketReturn(2:end) =
 cumprod(plot_data.MarketReturn(2:end)+1);

 figure;
 y = timeseries(plot_data{:,3:5});
 y.Name = 'Cumulative Return';
 y.TimeInfo.Units = 'months';
 y.TimeInfo.StartDate = convertStringsToChars(string(year_start));
 y.TimeInfo.Format = 'yyyy';
 performance_plot = plot(y);
 title('Portfolio Performance');
 xlabel('Time');
 ylabel('Cumulative Return');
 legend('Top Portfolio','Bottom Portfolio','Market
 Return','Location','northwest');
 grid on;

 figure;
 y = timeseries(plot_data{:,3:5});
 y.Name = 'Cumulative Return';
 y.TimeInfo.Units = 'months';
 y.TimeInfo.StartDate = convertStringsToChars(string(year_start));
 y.TimeInfo.Format = 'yyyy';
 performance_plot_log = plot(y);
 title('Portfolio Performance (Log Scale)');
 xlabel('Time');
 ylabel('Cumulative Return');
 legend('Top Portfolio','Bottom Portfolio','Market
 Return','Location','northwest');
 grid on;
 set(gca, 'YScale', 'log');

 figure;
 y = timeseries(plot_data{2:end,6:7});
 y.Name = 'Number of Portfolio Firms';
 y.TimeInfo.Units = 'months';
 y.TimeInfo.StartDate = convertStringsToChars(string(year_start));
 y.TimeInfo.Format = 'yyyy';
 obs_plot = plot(y);
 title('Number of Portfolio Firms');
 xlabel('Time');
 ylabel('Number of Firms');
 legend('Top Portfolio','Bottom Portfolio','Location','northwest');
 grid on;

end
```

Published with MATLAB® R2018b