```
**filename ref1 'C:\Users\smukhopadhyay\Documents\M4 forecasting
competition\m4data\yearly-train.csv';
filename ref1 'C:\Users\smukhopadhyay\Documents\M4 forecasting
competition\m4data\train_random_sample100.csv';


/*options obs = 101;*/
/**read in the yearly time-series**/
data year1;
infile  ref1 dsd dlm=',' missover lrecl=5000 firstobs=2;
length series $6.;
length default=3;
/**length v1 $15.;**/

input series $ randi $ v2 - v106
;
v1 = int(substr(series,2,5));
run;

/** transpose to run the tim-series **/
proc transpose data = year1 out = year2;
 var v1-v106;
 run;

/**option obs = 40;**/
data year3 (drop = _name_) s_num (drop = t);
  set year2;
  if _n_ = 1 then output s_num;
  else do;
    t = _n_ - 1; /** add t as time-series point **/
    output year3;
  end;
run;


/**next create a dummy error stat and all_fcst files for each combination of
hidden layer number and number of units**/
/**later it will be appended for each time series*/
data errorstat;
length stoppingreason $16. best $1.;
trynum = 0; numiterations =0; trainingerror=0;validationerror
=0;stoppingreason="VE";best="n";
run;

data all_fcst;
  t = .;
  run;

/**next create the NN forecast for each seires upto 6 year horizon **/
%macro nn_fcst;

%do i = 1 %to 100;
  /**next create the NN training data**/
  data nn&i (keep = t y&i);
         set year3;
```

```
          if col&i = . then delete;
            y&i = col&i;
            run;

  /**next create forecast data with t value for scoring**/
  data nnf&i (keep = t);
         set nn&i end = last;
         if last then
        do j=1 to 6;
           t+1;
             output;
          end;
      run;


proc hpneural data = nn&i /*noprint*/;
/*id year;*/
input t / level = int;

hidden  6; /** first hidden layer and num of hidden units in it **/ /** final
model is one hidden layer with 6 units **/
/**hidden  2;**/  /**second hidden layer and num of hidden units in it**/

train outmodel = model&i maxiter = 1000;
/**partition fraction (validate=0.15);**/
target
    y&i  / level = int ;

score out=trainscore&i; **training data scored;
   ods output training = trnsr&i; **with no print option training data will
not be created**/
run;

/**next build the error stat file by appending**/
proc append base=errorstat data=trnsr&i;
 where best = "Y";  /* where statement working for all the series except the
first one */
  run;

/**add a series number s_num in the data set**/
data trnsr&i;
  set trnsr&i end = last;
  if last;
  run;

** test files scored next;
proc hpneural data = nnf&i;
id t ;
score model = model&i out=testscore&i;  ** forecast generated for each series
for six horizons;
run;

**next merge the forecast file;
data all_fcst (drop = t _warn_);
  merge all_fcst testscore&i;
  run;
```

```
**next delete all the files;
  proc delete data = nn&i nnf&i model&i trainscore&i testscore&i trnsr&i;
    run;

%end;

%mend nn_fcst;

%nn_fcst

/**next take the average of the validation and training errors**/
/**option obs=100;**/
data errorstat1;
  set errorstat;
  if _n_ > 1;
  run;

proc means data=errorstat1 noprint;
  var trainingerror validationerror;
  output out = errormean mean = mean_terr mean_verr;
  run;

/**next transpose the forecasts **/
proc transpose data = all_fcst out = final_fcst;
 /**var p_y1 - p_y100;**/
 run;
```