INTERACTIVE LEARNING CURVE CALCULATOR USING SAS® SOFTWARE
R. David Sisemore, General Dynamics/Convair Division

ABSTRACT
How much will the 205th widget cost your company to build? What is the average cost of any given production unit? What will the projected cost of any future production lot cost?

These questions and many others can be answered by applying the 'Theory of Learning Curves', sometimes known as 'Improvement Curves'.

Basically, a learning curve is a line on a graph that represents two primary facts: [1] the time required to do a job decreases each time that job is repeated, and [2] that the amount of decrease will be less with each successive unit.

Learning curve theory began as far back as the early 1920's, with the first published work on the subject in the February 1936 'Journal of Aeronautical Sciences' by Thomas Wright. Engineers and Project Managers recognized that employees engaged in manufacturing and assembly functions could work more quickly as more units were produced. This per unit reduction in labor hours appeared, in many cases, to be a constant function represented as a straight line if plotted on a logarithmic scale.

This paper describes an interactive program using SAS software, which was designed to provide all necessary support and calculations for application of a learning curve to any given situation.

Introduction
In practice, learning curves are used to estimate the cost of producing a given number of a product such as airplanes, automobiles or even mouse traps. The estimated cost may be expressed in terms of monetary units, such as dollars and cents, or labor units, such as man-hours.

This program will calculate all required learning curve information for a given situation assuming the user knows one of the following:
A] The cost of three (3) or more actual production units
B] The total cost of a given production lot (unit #1 thru unit #Y)
   The value or percentage of the observed curve.
C] The value of any one (1) production unit
   The value or percentage of the observed learning curve.

Program Operation
This program was written as an interactive module to allow the user to apply various 'What if' parameters to a given learning curve. The entire program is controlled from one major SAS/AF® screen (see figure #1) where the user may select any of a variety of functions.

Terms used by 'Learning Curve Calculator'
UNIT
The manufacturing sequence of an item. The first one produced is unit #1, the second is #2 etc. Proper unit numbering is critical to the accuracy of the results.

COST
The actual observed cost of a particular unit. This cost may be expressed in terms of monetary units, such as dollars, or labor units, such as man-hours.

PREDICT
The predicted or estimated cost of a given unit.

AREA
The approximated cumulative cost (Area under the Curve) between two units. This would also be the estimated total cost for a production lot.

SLOPE
The percentage of slope of the learning curve. The predicted value will decrease by this percentage every time the unit number doubles. For example on an 80% slope, if unit #1 cost $1000., then unit #2 would be $800., unit #4 would be $640. and so on.

CONFIDENCE
An indication of the accuracy of the results. The closer this number is to 100% the more accurate the learning curve.

SYN-1
The estimated cost of unit #1 (synthetic-one).

MIDPOINT
The midpoint value is calculated to be the unit number at which the average unit cost will occur.

Main Menu Functions
1] Edit Data
Allows the user to add, change or delete specific cost per unit information. NOTE: the two displayed fields (PREDICT and AREA) are informational only and changing their value has no effect. See figure #3.

2] View Current Data
Displays all current learning curve data, allowing the user to check for specific information and check for data errors. See figure #4.
3) Generate Data

While 'Edit Data' is used for calculations where actual cost of units is known, this function is used where only small pieces of the situation are known. There are two different methods used depending on what information the user actually knows.

Method #1

Given:
A] Range of production units (unit #H thru unit #Y)
B] Total cost of unit #H-#Y (area under curve)
C] Desired Learning Curve (slope)

Method #2

Given:
A] Range of units to calculate
B] Total cost of one actual unit
C] Desired Learning Curve (slope)

4) Predict Unit/Cost

The user provides the first and last unit numbers in a production lot and the program will display the predicted costs, area under the curve and midpoint values.

5) Display Graph

Allows the user to select a range of units for which a log-log graph will be produced (see figure #3). The selected unit range can be any combination of past, present or future units.

6) Create a Report

The same as function 5 (Display Graph), except that a numeric report will be produced instead of a graph. The report will be written to a file called 'REPORT.LEARNCRU' and can be printed after exiting the program.

7) Load Data

Retrieves a previously saved dataset to which the user may add, delete or change any information. Any valid dataset name is allowed with the exception of 'NONE'. This dataset is stored in a SAS database (LEARNCRU.SASDB).

8) Save Data

Saves the current information to a permanent SAS database (See 'Load Data').

9) Directory of Files

Displays a list of all datasets previously saved. Data can also be deleted by placing a 'D' in the DMD field.

V) View Current Report

Display last created report on terminal screen. This function works only if function #6 (Create Report) has been executed.

Example #1

Assume we know the following:

Unit 18 cost 1,287.5 manhours to produce
Unit 23 cost 1,017.3 manhours
Unit 32 cost 897.6 manhours
Unit 40 cost 723.8 manhours

We want to know:

What is the total estimated cost of units 55-75?

Since we have the actual cost of three (3) or more units, we can use the 'EDIT DATA' function to enter this information. After the information has been entered, the program returns to the main menu again. This time however, additional information is displayed at the bottom of the screen.

Units=4  Confidence= 98.42
Syn-l=8774.94  Slope= 72.76

In order to find the total cost of units 55 thru 75, select function #4 (Predict Unit/Cost). The program will display information relating to the currently defined lot (18 thru 40). Simply change the 18 to 55 and the 40 to 75 and we find the cost of the new lot will be 11,232.18.

Example #2

Assuming we know the following data:

Units 22 thru 47 cost a total of 57,321 man-hours to produce. The observed learning curve was 78 percent.

We want to know:

What is the individual costs of each of the units (22 thru 47)?

Since we do not have the actual cost of at least three (3) individual units, we must use function #3 (GENERATE DATA) to create the necessary information. When the 'Generate Data' menu appears (see figure #2), select method #1. When the data-entry screen appears, fill in the blank with the known information.

GENERATE DATA FOR UNITS 22 to 47 WITH AN INTERNAL DF 1 UNITS --
WHERE THE AREA UNDER THE CURVE IS 57321 AND THE PERCENT SLOPE IS 78.

Upon returning to the main menu screen, the bottom portion will display a summary of the created information.

Units=26  Confidence= 100.00
Syn-l=7741.27  Slope= 78.00

In order to see the individual cost per unit, select function #2 (VIEW DATA).
Example #3

Assume we know the following:

That Unit #18 cost 1,287.5 man hours to produce. The observed curve was 82 percent.

We want to know:

What is the individual cost of units 10 thru 20?

Again, since we must use function #3 (GENERATE DATA) when the 'Generate Data' menu appears (see figure #2), select method #2. When the data-entry screen appears, fill in the blank with the known information.

GENERATE DATA FOR UNITS 10 TO 20 WITH AN INTERVAL OF _1_ UNITS--

WHERE THE KNOWN COST OF UNIT 18 IS 1287.5 AND THE PERCENT SLOPE IS 82_.

Upon returning to the main menu screen, the bottom portion will display a summary of the created information.

<table>
<thead>
<tr>
<th>Units</th>
<th>Confidence</th>
<th>Syn-l</th>
<th>Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>100.00</td>
<td>7741.27</td>
<td>78.00</td>
</tr>
</tbody>
</table>

In order to see the individual cost per unit select function #2 (VIEW DATA).

Conclusion

The proper application of Learning Theory in a manufacturing environment can greatly enhance cost forecasting as well as future business and contract negotiations.

References

'Aircraft Cost Curves - Derivation Analysis Projection'
Tommie Fowkes, General Dynamics/Ft. Worth Division, August 1963

'The Improvement Curve Trainers Manual'

Contact Author

R. David Sisemore
General Dynamics/Convair
PO Box 85377
Mail Zone 63-2450
San Diego, California 92138
Generate data for units &UNIT1 to &UNIT2 with an interval of &INT units
Where the area under the curve is &AREA
and the percent of slope is &CURVE

Press PF3 or PF15 to Continue or Press PF1 or PF13 for HELP

run;

Generate data for units &UNITX to &UNIT2 with an interval of &INT units
Where the known cost of unit &UNITX is &COST
and the percent of slope is &CURVE

Press PF3 or PF15 to Continue or Press PF1 or PF13 for HELP

run;

Enter Data for &TYPEX

First Unit = &UNIT1
Last Unit = &UNIT2
Interval = &INT

&TYPEX Title = &TITLE

Press PF3 or PF15 to Continue or Press PF1 or PF13 for HELP

run;
PROGRAM: LEARNCRV
LANGUAGE: SAS / SAS-GRAPH 5.08
WRITTEN: R. DAVID SISEMORE
LOCATION: GENERAL DYNAMICS/CONVAIR
DATE: 22 JUL 86
UPDATED: 17 DEC 86
FUNCTION: LEARNING CURVE CALCULATION PROGRAM
NOTES: USERID IS PASSED TO PROGRAM USING SYSPARM= FUNCTION

SAS INPUT: LEARNCRV.SASDB
NON-SAS-OUTPUT: REPORT.LEARNCRV

OPTIONS NOTEXT82 PS=60 LS=80 DQUOTE;
LOG FORMAT FOR GRAPHS:

***** ***** MACRO: DEVICE - SELECT GRAPHICS OUTPUT DEVICE
***** THE FOLLOWING DEVICES ARE SUPPORTED BY THIS MACRO
***** IBM3279 - COLOR TERMINAL MODEL IBM 3279-3
***** HP7550A - 8 PEN PLOTTER MODEL HP 7550A

%GLOBAL DEVICE NODENAME HGT1 HGT2 FONT WIDTH COLOR
%MACRO DEVICE(DEVICE,NODENAME);

%IF &DEVICE=IBM3279 %THEN %00;
%LET HGT1=1.0; %LET HGT2=2.0; %LET FONT=NONE;
%LET WIDTH=; %LET COLOR=WHITE;

GOPTIONS DEVICE
COLORS (RED,YELLOW,CYAN,BLUE, GREEN, PINK, WHITE)
PENMOUNTS 17
HBY
CRY
FNY
CFULL
CTITLE
VPOS
HPOS
VSIZE
HSIZE
NOTEXT82;

OPTIONS DEVADDR=(.,., &NODENAME): %END;

%IF &DEVICE=IBM3287 %THEN %00;

GOPTIONS DEVICE
COLORS (BLUE,RED,GREEN,BLACK)
PENMOUNTS 4
HBY
CRY
FNY
CFULL
CTITLE
VPOS
HPOS
VSIZE
HSIZE
NOTEXT82;

%END;

%IF &DEVICE=HP7550 %THEN %00;

GOPTIONS DEVICE
COLORS (BLUE,RED,GREEN,PINK)
PENMOUNTS 8
HBY
CRY
FNY
CFULL
CTITLE
VPOS
HPOS
VSIZE
HSIZE
NOTEXT82;

%END;

***** ***** MACRO: DEVICE - SELECT GRAPHICS OUTPUT DEVICE

SPECIAL CASE:

$GLOBAL DEVICE NODENAME HGT1 HGT2 FONT WIDTH COLOR;
%MACRO DEVICE(DEVICE,NODENAME);

%GRAPHICS OUTPUT DEVICE=IBM3279-3 COLOR TERMINAL;

%END;

%IF &DEVICE=IBM3279 %THEN %00;

$LET HGT1=1.0; $LET HGT2=2.0; $LET FONT=NONE;
$LET WIDTH=; $LET COLOR=WHITE;

GOPTIONS DEVICE
COLORS (RED,YELLOW,CYAN,BLUE, GREEN, PINK, WHITE)
PENMOUNTS 17
HBY
CRY
FNY
CFULL
CTITLE
VPOS
HPOS
VSIZE
HSIZE
NOTEXT82;

OPTIONS DEVADDR=(.,., &NODENAME): %END;

%IF &DEVICE=IBM3287 %THEN %00;

GOPTIONS DEVICE
COLORS (BLUE,RED,GREEN,BLACK)
PENMOUNTS 4
HBY
CRY
FNY
CFULL
CTITLE
VPOS
HPOS
VSIZE
HSIZE
NOTEXT82;

%END;

%IF &DEVICE=HP7550 %THEN %00;

GOPTIONS DEVICE
COLORS (BLUE,RED,GREEN,PINK)
PENMOUNTS 8
HBY
CRY
FNY
CFULL
CTITLE
VPOS
HPOS
VSIZE
HSIZE
NOTEXT82;

%END;

***** ***** MACRO: DEVICE - SELECT GRAPHICS OUTPUT DEVICE

SPECIAL CASE:
%IF &DEVICE=HP7550A %THEN %DO;
%LET HGT1=1.0; %LET HGT2=1.6; %LET FONT=NULL; %LET WIDTH=1; %LET COLOR=BLACK;

GOPTIONS DEVICE = HP7550A
COLORS = (BLACK RED GREEN ORANGE LIME GOLD CYAN)
PENMOUNTS = D
CBY = BLACK
FIB = NONE
CTITLE = BLACK
VPOS = 40
HPOS = 100
VSIZE = D
HSIZE = D
BAND = 6600
HANDSHAKE = QMODE
GACCESS = GSASFILE
GPROTOCOL = GSASFILE
PROMPTCHARS = '0'DO04000000X
GOUTMODE = REPLACE
GOUTTYPE = INDEPENDENT
AUTOFEED = NOPROMPT
NOFILL = NOSPHERE
NOPOLYGON = NOGRAPH
NOTEXT = 82;

%END;

%MACRO closeDev(DEVICE, NODENAME);
%IF &DEVICE=HP7550A %THEN %DO;
* CLEAR;
* TRANSMITTING GRAPHICS FILE TO HP7550A PLOTTER;
* NO FREE (GSASFILE);
* DSPRINT HP7550.GRAPH &NODENAME;
* CLEAR;
%END;
%MEND closeDev;

%MACRO area(x,y);
PRE=predict(x,y); AREA=area(x,y); MIDPT=midpt(x,y);
%MEND area;

%MACRO midpt(x,y);
MIDPT=midpt(x,y);
%MEND midpt;

%MACRO update;
%LET COUNT=0;
DATA TEMPLC(KEEP=UNIT COST PREDICT AREA);
FORMAT UNIT 9.2 COST 9.2 PREDICT 9.2 AREA 9.2;
SET TEMPLC ENO=DONE;
IF _N_=1 THEN DO; LOW=9E9; HIGH=-9E9; RETAIN HIGH LOW; END;
If UNIT=THEN DO; COUNT=1; PREDICT=AREA;
 IF UNIT>HIGH THEN H=UNIT;
 IF UNIT<LOW THEN L=UNIT;
 OUTPUT;
END;
IF DONE=1 THEN DO;
SLOPE=10**((-EQU_B)/(EQU_A)*UNIT);
EQU_A=(K-EQU_B*J)/COUNT;
J=EQU_B*(R2-J*K/COUNT);
%END;
%MEND update;

%IF &DEVICE=HP7550A %THEN %DO;
%LET HGT1=1.0; %LET HGT2=1.6; %LET FONT=None; %LET WIDTH=1; %LET COLOR=BLACK;

GOPTIONS DEVICE = HP7550A
COLORS = (BLACK RED GREEN BLUE ORANGE LIME GOLD CYAN)
PENMOUNTS = D
CHY = BLACK
FIB = NONE
CTITLE = BLACK
VPOS = 40
HPOS = 100
VSIZE = D
HSIZE = D
BAND = 6600
HANDSHAKE = QMODE
GACCESS = GSASFILE
GPROTOCOL = GSASFILE
PROMPTCHARS = '0'DO04000000X
GOUTMODE = REPLACE
GOUTTYPE = INDEPENDENT
AUTOFEED = NOPROMPT
NOFILL = NOSPHERE
NOPOLYGON = NOGRAPH
NOTEXT = 82;

%END;

%MACRO PREDICT(UNIT);
PRE=predict(UNIT); AREA=predict(UNIT); MIDPT=predict(UNIT);
%MEND PREDICT;
\[ M = \frac{(K*K)}{COUNT}; \]

\[ CONFID = \text{SQRT}(J/M) \times 100; \]

\[ \text{CALL SYMPUT('EQN_A', 'EQN_A');} \]

\[ \text{CALL SYMPUT('EQN_B', 'EQN_B');} \]

\[ \text{CALL SYMPUT('CONFID', 'CONFID', 9.2);} \]

\[ \text{CALL SYMPUT('SLOPE', 'SLOPE', 9.2);} \]

\[ \text{END;} \]

\[ \text{RUN;} \]

\[ \text{------------------ CALCULATE PREDICTION COST FOR ALL KNOWN UNITS -------------;} \]

\[ \text{DATA TEMPLC(KEEP=UNIT COST PREDICT AREA);} \]

\[ \text{FORMAT UNIT 9.2 COST 9.2 PREDICT 9.2 AREA 9.2;} \]

\[ \text{SET TEMPLC;} \]

\[ \text{IF N = 1 THEN DO;} \]

\[ \text{CALL SYMPUT('SLOPE', 'SLOPE', 9.2);} \]

\[ \text{CALL SYMPUT('CONFID', 'CONFID', 9.2);} \]

\[ \text{END;} \]

\[ \text{PREDICT(UNIT);} \]

\[ \text{PREDICT=PRE;} \]

\[ \text{SAREA(U埃尔,UNIT);} \]

\[ \text{RUN;} \]

\[ \text{SEND;} \]

\[ \text{SEND UPDATE;} \]

\[ \text{******************************************************} \]

\[ \text{MACRO GRAPH;} \]

\[ \text{PROC PLOT DATA=PLOT;} \]

\[ \text{AXIS1 LABEL=('+\text{UNIT}' \text{COLOR} \text{F=FONT} \text{UN} \text{IT});} \]

\[ \text{ORDER=('MIN TO ' \text{MAX BY 1});} \]

\[ \text{VALUES=('MIN TO ' \text{MAX BY 1});} \]

\[ \text{AXIS2 LABEL=('+\text{COLOR} \text{F=FONT} \text{A=90 R=0 'COST'});} \]

\[ \text{ORDER=('MIN TO ' \text{MAX BY 1});} \]

\[ \text{VALUES=('MIN TO ' \text{MAX BY 1});} \]

\[ \text{LEGEND LABEL=NONE} \]

\[ \text{FRAME;} \]

\[ \text{PLOT COST*UNIT/LINE/} \]

\[ \text{CAXIS=COLOR} \]

\[ \text{CITEXT=COLOR} \]

\[ \text{CCHRT=COLOR} \]

\[ \text{CQVIE=COLOR} \]

\[ \text{LINES=3} \]

\[ \text{EXE=5} \]

\[ \text{MIDPOINTS=\text{JOIN}} \]

\[ \text{SLOPES=\text{NONE}} \]

\[ \text{VALUES=\text{JOIN}} \]

\[ \text{TITLE1 \text{H=FONT \text{C=COLOR \text{F=FONT}}} \text{OPS. ADMIN}} \]

\[ \text{TITLE2 \text{H=FONT \text{C=COLOR \text{F=FONT}}} \text{DEPT. 245-01 \text{GENERAL DYNAMICS \text{CONVAIR DIVISION \text{L=LEARNING CURVE}}}} \]

\[ \text{FOOTNOTE1 \text{H=FONT \text{C=COLOR J=C \text{SYN-1=SLOPE \text{SLOPE CONFID=CONFID}}} \text{MIDPOINT UNIT=\text{JOIN}}} \]

\[ \text{FOOTNOTE2 \text{H=FONT \text{C=COLOR J=C \text{MIDPOINT VALUE=\text{JOIN}}} \text{MIDVAL}} \]

\[ \text{RUN;} \]

\[ \text{SEND GRAPH;} \]

\[ \text{******************************************************} \]

\[ \text{MACRO EXPAND;} \]

\[ \text{PROC DISPLAY CATALOG=AfSCRN01.Af.lCRT.PROGRAM;} \]

\[ \text{SEND;} \]

\[ \text{SEND UPDATE;} \]
DATA TEMP(KEEP=UNIT COST PREDICT AREA MIDPT);
  DATA(UNIT MIDPT); PREDICT=MIDPT;
AREA(UNIT MIDPT);
  PREDICT(MIDPT);
COST=MIDPT; PREDICT=PRE;
VREF=LOG10(PREDICT); HREF=LOG10(MIDPT);
  CALL SYMPUT('MloPT',PUT(MIDPT,.7)); CALL SYMPUT('MloPT',PUT(PREDICT,.7));
  CALL SYMPUT('VREF',VREF);
  CALL SYMPUT('HREF',HREF);
MIDPT=. COST=.
OUTPUT:
DO UNIT=UNIT1 TO UNIT2 BY INTERVAL;
  IF UNIT=MPOINT THEN DO;
  AREA(UNIT1,UNIT);
  PREDICT=PRE;
  MIDPT=.; COST=.; OUTPUT;
END;
END:
RUN;

PROC SORT DATA=TEMP; BY UNIT;
RUN;

DATA TEMP; UPDATE TEMP TEMPLC; BY UNIT;
IF UNIT<>UNIT1 OR UNIT<>UNIT2 THEN DELETE;
RUN;
%MEND EXPAND;

*************************************************
* MACRO
* LEARNCRV - MAIN PROGRAM MODULE
*************************************************
%MACRO LEARNCRV;
%LET RPTFLAG=NO;
%LET COUNT=0;
DATA TEMPLC(KEEP=UNIT COST PREDICT AREA);
  FORMAT UNIT 9.2 COST 9.2 PREDICT 9.2 AREA 9.2;
UNIT=.; COST=.; PREDICT=.; AREA=.;
RUN;

*************************************************
* MAIN MENU
*************************************************
%START:
%LET RPTFLAG=NO;
%LET COUNT=0;
DATA TEMPLC(KEEP=UNIT COST PREDICT AREA);
  FORMAT UNIT 9.2 COST 9.2 PREDICT 9.2 AREA 9.2;
UNIT=.; COST=.; PREDICT=.; AREA=.;
RUN;

%MENU:
%IF &SEL=1 THEN %GOTO EDIT;
%IF &SEL=2 THEN %GOTO VIEW;
%IF &SEL=3 THEN %GOTO GENDATA;
%IF &SEL=4 THEN %GOTO REPORT;
%IF &SEL=5 THEN %GOTO VIEWRPT;
%IF &SEL=6 THEN %GOTO EXIT;

%EDIT DATASETS (FUNCTION #1)
%EDIT:
PROC fSEDIT DATA=TEMPLC SCREEN=AFSCRN01.fSEdIT.learnCRV; RUN;
%UPDATE; %GOTO MENU;

%VIEW:
PROC fSPRINT DATA=TEMPLC; RUN;
%GOTO MENU;

* GENERATE LEARNING CURVE DATA (FUNCTION #3)
%GENDATA:
%LET UNIT1=&LOW;
%LET UNIT2=&HIGH;
%LET UNITX=1;
%LET COST=10;
%LET TITLE='SAMPLE TITLE';
%LET DEVICE=HP7550A;
%LET INTERVAL=1;
%LET AREA=1000;
%LET MEMBER=NDNE;
%LET NODENAME=NONE;
%CLEAR;
PROC DISPLAY CATALOG=AFSCRN01.AF.LCGEN,PROGRAM; RUN;
%IF &GEN=1 THEN %GOTO EDIT;
%IF &GEN=2 THEN %GOTO VIEW;
%IF &GEN=3 THEN %GOTO START; ** CLEAR ALL DATA **;

** GENERATE DATA - METHOD #1 **
* GIVEN: RANGE OF UNITS!
* AREA UNDER CURVE!
* PERCENT OF SLOPE!
%IF &GEN=1 THEN %GOTO EDIT;
%CLEAR;
PROC DISPLAY CATALOG=AFSCRN01.AF.LCGEN,PROGRAM; RUN;

** GENERATE DATA - METHOD #2 **
* GIVEN: RANGE OF UNITS
* AREA UNDER CURVE!
* PERCENT OF SLOPE!
%IF &GEN=2 THEN %GOTO EDIT;
%CLEAR;
PROC DISPLAY CATALOG=AFSCRN01.AF.LCGEN,PROGRAM; RUN;

** GENERATE DATA - METHOD #3 **
* GIVEN: RANGE OF UNITS
* AREA UNDER CURVE!
* PERCENT OF SLOPE!
%IF &GEN=3 THEN %GOTO EDIT;
%* GENERATE DATA - METHOD #2
%* GIVEN: RANGE OF UNITS
%* COST OF GIVEN UNIT
%* PERCENT OF SLOPE

%if &GEN=2 %then %do;
PROC DISPLAY CATALOG=AFSCRN01.AF.LCGEN2.PROGRAM; RUN;
DATA TEMPLC(KEEP=UNIT COST);
UNIT=&UNITX; COST=&COST; OUTPUT;
DO I=1 TO 8;
  UNIT=2*UNIT;
  COST=COST*(&CURVE/l00);
  OUTPUT;
END;
RUN;
%update;
DATA TEMPLC(KEEP=UNIT COST PREDICT AREA);
FORMAT UNIT 9.2 COST 9.2 PREDICT 9.2 AREA 9.2;
DO [=&UNIT1 TO &UNIT2 BY &INTERVAL;
  UNIT=I:
  %PREDICT(UNIT): COST=PR[; PREDICT=.;
  OUTPUT;
END;
RUN;
%update;
%goto MENU;
%end;
%goto MENU:

*----------------- PREDICT UNIT COST (FUNCTION #4) --------------------.
%PREDATA:
%let UNIT1 = &LOW; %let UNIT2 = &HIGH;
%PREDAT1: DATA _NULL_; RUN;
%let UNITA = &UNIT1; %let UNITB = &UNIT2;
%let PREDICT=; %let AREA=;
%PREDICT(MIDPT): CALL SYMPUT('MIDPRE', PREDICT, 9.2));
%AREA(MIDPT, UNIT2): CALL SYMPUT('MIDAREA', AREA, 9.2));
%PREDICT(MIDPT): CALL SYMPUT('MIDPRE', PREDICT, 9.2));
%AREA(MIDPT, UNIT2): CALL SYMPUT('MIDAREA', AREA, 9.2));
%let UNITA = &UNIT1; %let UNITB = &UNIT2;
%clear; PROC DISPLAY CATALOG=AFSCRN01.AF.LCPRE,PROGRAM; RUN;
%if UNITA=O OR UNITB=O %then %goto MENU;
%let UNIT1 = &UNIT1; %let UNIT2 = &UNIT2;
%goto PREDAT1;

*------------- DRAW LEARNING CURVE GRAPH (FUNCTION #5) ----------------;
%graf:
%expand(graph);
DATA PLOT(KEEP=UNIT LINE COST);
SET TEMPLC DROP=DDNE;
LENGTH LINE=15;
PLIT=(UNIT LINE COST);
IF &CONFID=100 AND COST=THEN DO;
  X_MAX=LOG10(&MAX); X_MIN=X_MIN;
  Y_MAX=LOG10(&MAX); Y_MIN=Y_MIN;
  COST=COST; XX=UNIT; DROP CC XX;
  COST=Y_MAX; UNIT=X_MAX; LINE='2.MID-POINT'; OUTPUT;
  COST=CC; UNIT=XX;
END;
IF UNIT=, THEN DELETE;
UNIT=ROUND(10**&UNIT, 0.1);
%if &PREDICT= THEN PREDICT=ROUND(LOG10(&PREDICT), 0.1);
%if &UNIT = &MAX THEN &UNIT=UNIT;
%if COST < MAX AND COST > = THEN &Y_MAX=COST;
%if PREDICT < MAX AND PREDICT > = THEN &Y_MAX=PREDICT;
%if &UNIT < &MIN THEN &UNIT=UNIT;
%if COST < MAX AND COST > = THEN &Y_MIN=COST;
%if PREDICT < MAX AND PREDICT > = THEN &Y_MIN=PREDICT;
%if &CONFID=100 AND COST= THEN DO;
  Line='3.ACTUAL COST'; OUTPUT;
  %if PREDICT= THEN DO;
  Line='1.PREDICTED COST'; COST=PREDICT; OUTPUT;
  %end;
  %if DONE=1 THEN DO;
  X_MAX=ROUND(&MAX); IF &XX_MAX=ROUND(X_MAX); IF &YY_MAX=ROUND(Y_MAX); IF &YY_MAX=YY_MAX THEN &YY_MAX=YY_MAX;
  %end;
  %if &UNIT < &UNIT < THEN ;
  %if &UNIT < &UNIT < THEN ;
  %if &UNIT < &UNIT < THEN ;
  %if &CONFID=100 AND COST= THEN DO;
  %end;
  %device (IBM3279,XXXXXXXX):
%graf;
%if &NODENAME=NONE %then %do;
%clear;
PROC DISPLAY CATALOG=AFSCRN01.AF.HARDCOPY,PROGRAM; RUN;
%device(&DEVICE, &NODENAME);
%close dev(&DEVICE, &NODENAME);
%end;
%end;
%goto MENU;
------ WRITE LEARNING CURVE REPORT (FUNCTION #6) -------

REPORT:
%EXTRACT(REPORT);
TITLE "&TITLE";
DATA NULL;
FILE OUT HEADER=NEWPAGE;
SET TEMP;
IF N=1 THEN DO;
%PREDICT(1); SYN_1=PRED;
SLOPE=SLOPE;
CONFID=CONFID;
RETAIN SYN_1 SLOPE CONFID;
END;
IF MIDPT=1 THEN LINK LINE;
PUT @ 1 'I' @ 3 UNIT COMMA8.2 @ 12 'I' @14 COST COMMA15.2 
@30 'I' @32 PREDICT COMMA15.2 @48 'I' @50 AREA COMMA15.2 
@66 'I';
CALL SYMPUT('RPTFLAG','YES');
IF MIDPT=1 THEN LINK LINE;
RETURN;
NEWPAGE:
@12 ',' @14 'SYN_1'
LINK LINE;
PUT @ 1 'I' @ 30 'SLOPE' @ 48 'I' @50 'CONFIDENCE'
@66 'I';
LINK LINE;
PUT @ 1 'I' @ 30 'SLOPE' @ 48 'I' @50 'CONFIDENCE'
@66 'I';
LINK LINE;
PUT @ 1 'I' @ 30 'SLOPE' @ 48 'I' @50 'CONFIDENCE'
@66 'I';
LINE:
PUT @ 66 '+' @ 1 '+' @ 66 '+' @ 66 '+' @ 66 '+';
RETURN;
RUN;
%GOTO MENU;

-------- LOAD DATASET (FUNCTION #7) ---------

%LOAD:
%IF MEMBER=NONE %THEN %GOTO NOFILE;
DATA TEMP(KEEP=UNIT COST PREDICT AREA);
FORMAT UNIT 9.2 COST 9.2 PREDICT 9.2 AREA 9.2;
SET DB.8eMEMBER END=DONE;
IF UNIT=. THEN DELETE:
OUTPUT: COUNT=1;
IF DONE=1 THEN DO;
IF COUNT=1 THEN COUNT=0;
CALL SYMPUT('COUNT',PUT(COUNT,5.));
END;
RUN;
%UPDATE;
%GOTO MENU;

-------- SAVE DATASET (FUNCTION #8) -------

%SAVE:
%IF MEMBER=NONE %THEN %GOTO NOFILE;

------ LIST DATASETS (FUNCTION #9) ------

%LISTFILE: 
%PROC DATASETS DNAME=OBS; RUN;
%GOTO MENU;

------ VIEW REPORT (FUNCTION #10) -------

%VIEWRT: 
%IF RPTFLAG=NO %THEN %GOTO NOFILE;
%PROC OSLIST DNAME=OUT; RUN;
%GOTO MENU;

------ NOTIFY USER THAT A DATASET NAME IS REQUIRED -------

%NOFILE: 
%CLEAR;
%PROC DISPLAY CATALOG=AFSCRD01.AF.LCNOFI LE.PROGRAM; RUN;
%GOTO MENU;

------ NOTIFY USER THAT NO REPORT HAS BEEN GENERATED -------

%NORPT: 
%CLEAR;
%PROC DISPLAY CATALOG=AFSCRD01.AF.LCNORPT.PROGRAM; RUN;
%GOTO MENU;

------ SINGLE POINT EXIT FROM MAIN PROGRAM -------

%EXIT: 
%MEND LEARNCRV;
%LEARNCRV;