**STATISTICAL ANALYTICAL SOFTWARE (SAS)**

**CREATING TYPES OF DATASET :**

***DATA WORK.A1;* => In Work Library, A1 DATASET is created.**

***DATA WORKA1;* =>In Work Library,WORKA1 DATASET created.**

***DATA WORK A2;* => In Work Library, WORK DATASET, A2 DATASET. 2 DATASETS will be created.**

**1. BASIC PROGRAM : (own datalines can be created)**

**DATA WORK.A1;**

**INPUT ENAME $ ENO ESAL; ($ is used for string)**

**DATALINES;**

**AA 30 240**

**BB 31 320**

**CC 32 500**

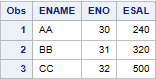
**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**2. PROGRAM WITH LONG DATALINES : (Incorrect datalines)**

**DATA WORK.A1;**

**INPUT ENAME $ ENO ESAL;**

**DATALINES;**

**TaylorAlisonSwift 1234567890123456789 198,345,678,123.356**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**(Space should not be given in between names. If space is given, it considers the rest of the name as for the next variable)**

**OP:**

****

**(Only 8 characters will be shown. ENO is shorten, E18 => 10^18. ESAL is shown as '.' as ESAL value is with included with comma(,) in datalines. If given only numeric without any special characters, OP will be shown.)**

**READ AND WRITE DATALINES**

**READ => INFORMAT WRITE => FORMAT**

**Default Width = 8.**

**MINIMUM WIDTH = 3.**

**MIDDLE WIDTH = 18.**

**MAXIMUM WIDTH =37.**

**3. TO READ DATALINES WITH COMMA AND DECIMAL :**

**(In the previous program OP was not shown for ESAL as it was included with comma. To show the OP, Width should be given accordingly.**

**[NOTE: Comma and Decimal Point won't be shown in the OP as this program only reads the Datalines.So,width is given only in the variable,not in the datalines.]**

**WIDTH should be included along with the comma's,decimal point and number of decimal numbers after the point.)**

**DATA WORK.A1;**

**INFORMAT ENAME $12. ENO 16. ESAL COMMA19.3;**

**INPUT ENAME ENO ESAL;**

**DATALINES;**

**TaylorAlisonSwift 1234567890123456789 198,345,678,123.356**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**4. TO WRITE DATALINES WITH COMMA AND DECIMAL :**

**( Width should be included both in INFORMAT STATEMENT [to read] and in FORMAT STATEMENT [to write]).**

**Firstly, the Datalines must be made to read for writing.Then only can write or make changes/modify in writing the Program Datalines.**

**DATA WORK.A1;**

**INFORMAT ENAME $12. ENO 16. ESAL COMMA19.3;**

**INPUT ENAME ENO ESAL;**

**FORMAT ENAME $12. ENO 16. ESAL COMMA19.3;**

**DATALINES;**

**TaylorAlisonSwift 1234567890123456789 198,345,678,123.356**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**5. TO MODIFY AND WRITING DATALINES WITH COMMA AND DECIMAL :**

**(A dataline can be modified only once in one program.If the datalines has to be modified and shown multiple times, multiple datalines programs has to be written.)**

**DATA WORK.A1;**

**INFORMAT ENAME $18. ENO 16. ESAL COMMA19.3;**

**INPUT ENAME ENO ESAL;**

**FORMAT ENAME $12. ENO 21. ESAL COMMA19.3;**

**DATALINES;**

**TaylorAlisonSwift 1234567890123456789 198,345,678,123.356**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**ENO WIDTH was increased in FORMAT STATEMENT which shown the full number in OP.**

**FORMAT ENAME $12. ENO COMMA27. ESAL COMMA19.3;**

**OP:**

****

**COMMA27. has included comma and decimal in OP.**

**TO INCLUDE *CURRENCY SYMBOLS* IN WRITING DATALINES**

**DOLLARS ($) :**

**FORMAT ENAME $12. ENO 21. ESAL DOLLAR17.3; (only $)**

**OP:**

****

**To include both dollars and comma increase width accordingly.**

**FORMAT ENAME $12. ENO 21. ESAL DOLLAR20.3;**

**OP:**

****

**OTHER CURRENCY SYMBOLS :**

**DOLLAR ($) => +1 WIDTH**

**OTHER CURRENCY SYMBOLS => Add 'NLMNL' respective currency shortform +3.WIDTH**

**INR (Rs) :**

**FORMAT ENAME $12. ENO 21. ESAL NLMNLINR23.3;**

**OP:**

****

**EURO (€):**

**FORMAT ENAME $12. ENO 21. ESAL NLMNLEUR23.3;**

**OP:**

****

**YEN (¥) :**

**FORMAT ENAME $12. ENO 21. ESAL NLMNLJPY23.3;**

**OP:**

****

**SAS DATE**

**SAS Date is a numeric value that is equavelent to number of days that occur between 1.1.1960 and a particular date.**

**The dates before 1.1.1960 - stored as NEGATIVE INTEGER.**

**30.12.1959 => (-2)**

**31.12.1959 => (-1)**

**The dates after 1.1.1960 - stored as POSITIVE INTEGER.**

**2.1.1960 => (1)**

**3.1.1960 => (2)**

**Referance Date of SAS 1.1.1960 - the value is 0.**

**1.1.1960 => (0)**

**TO READ DATE DATA :**

**DATA WORK.A1;**

**INFORMAT EDATE DDMMYY10.;**

**INPUT EDATE;**

**DATALINES;**

**15/08/1947**

**26/01/1950**

**01/01/1960**

**26/06/2002**

**27/01/2003**

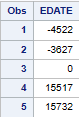
**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**DATES will be shown in numbers, not in DATE FORMAT.**

**TO WRITE DATE DATA :**

**DATA WORK.A1;**

**INFORMAT EDATE DDMMYY10.;**

**INPUT EDATE;**

**FORMAT EDATE DDMMYY10.;**

**DATALINES;**

**15/08/1947**

**26/01/1950**

**01/01/1960**

**26/06/2002**

**27/01/2003**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**While writing, OP will be in DDMMYY FORMAT.**

**DATE FORMATS:**

**The DEFAULT DATE FORMAT is (/) slash for DDMMYY and MMDDYY.**

**For YYMMDD => (-) ifan(dash)**

**DELIMITERS/SEPARATORS:**

**B - BLANK**

**C - COLON (:)**

**D - DASH (-) Ifan**

**P - PERIOD (.)**

**S - SLASH (/)**

**WEEKDATE - Monday, January 27,2003**

**MIN WIDTH - 3**

**MIDDLE WIDTH - 18**

**MAX WIDTH - 37**

**WORDDATE - January 27,2003**

**MIN WIDTH - 3**

**MIDDLE WIDTH - 16**

**MAX WIDTH - 32**

**JULIAN - NO OF DAYS from 01/01/1960 till given date.**

**MIN - 3**

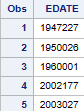
**MAX - 365,366**

**Leading 0 - zero before number**

**Trailing 0 - zero after number**

**FORMAT EDATE JULIAN7.;**

**OP:**

****

**DATA WORK.A1;**

**INFORMAT EAGE 3. ECOMM 7.2;**

**INPUT EAGE ECOMM;**

**FORMAT EAGE 3. ECOMM 9.4;**

**DATALINES;**

**8 9876.54**

**16 8765.43**

**32 7654.32**

**64 6543.21**

**102 5432.01**

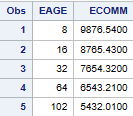
**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

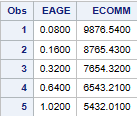
**In INFORMAT STATEMENT, 3. WIDTH is given for EAGE. If WIDTH not given, then DATALINE will be shown in OP with LEADING ZEROS. 2. WIDTH can also be given. But, 3. Is the minimum WIDTH.**

**INFORMAT EAGE ECOMM 7.2;**

**INPUT EAGE ECOMM;**

**FORMAT EAGE ECOMM 9.4;**

**OP:**

****

**This is the OP when WIDTH is not given in INFORMAT STATEMENT(reading)**

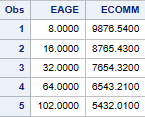
**In FORMAT STATEMENT (writing), WIDTH is not given. This results to TRAILING ZEROS.**

**INFORMAT EAGE 3. ECOMM 7.2;**

**INPUT EAGE ECOMM;**

**FORMAT EAGE ECOMM 9.4;**

**OP:**

****

**COPYING**

**COPYING DATASETS FROM *DIFFERENT* LIBRARIES**

**SET - Taking from Datalines from a specific Library.**

**DATA WORK.A1;**

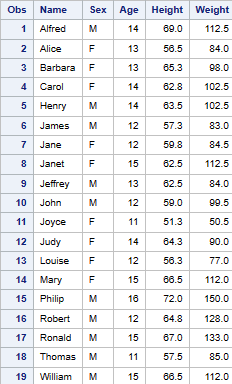
**SET SASHELP.CLASS;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**COPYING DATASETS FROM *SAME* LIBRARIES**

**DATA WORK.A2;**

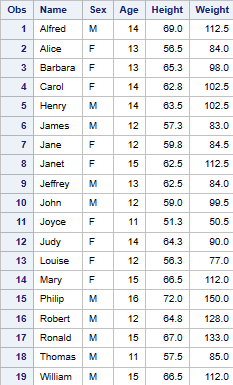
**SET WORK.A1;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**OP:**

****

**SELECTING *VARIABLES* TO *COPY***

**DATA WORK.A3;**

**SET SASHELP.CLASS;**

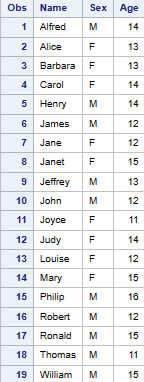
**KEEP NAME SEX AGE;**

**RUN;**

**PROC PRINT DATA = WORK.A3;**

**RUN;**

**OP:**

** Only variables (heading) are copied from SASHELP CLASS. Datalines are from A3 Dataset.**

**TO COPY *NUMERIC VARIABLES* FROM DIFFERENT LIBRARIES :**

**DATA WORK.A33;**

**SET SASHELP.CLASS;**

**KEEP \_numeric\_;**

**RUN;**

**PROC PRINT DATA = WORK.A33;**

**RUN;**

**OP:**

****

**TO COPY *CHARACTER VARIABLES* FROM DIFFERENT LIBRARIES :**

**DATA WORK.A33;**

**SET SASHELP.CLASS;**

**KEEP \_character\_;**

**RUN;**

**PROC PRINT DATA = WORK.A33;**

**RUN;**

**OP :**

****

**SELECTING *VARIABLES* TO *DROP***

**DATA WORK.A4;**

**SET SASHELP.CLASS;**

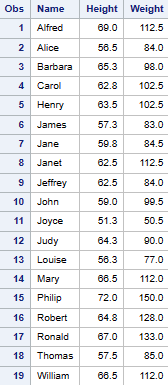
**DROP SEX AGE;**

**RUN;**

**PROC PRINT DATA = WORK.A4;**

**RUN;**

**OP:**

****

**MERGING**

**MERGING 2 DATASETS**

**There must be one variable in common among the datasets to be merged.**

**The common variable must be unique variables.**

**The datasets must be sorted in ascending order of the common variable.**

**SORTING**

**Before Merging, look for Sorting.**

**The below program is done in the same Library’s Dataset. Do it accordingly when in need.**

**/\* SORTING AND REMOVING DUPLICATES \*/**

**/\* \*/ => Used as Command Prompt.**

**DATA WORK.A3;**

**SET SASHELP.CLASS;**

**KEEP NAME AGE SEX;**

**RUN;**

**PROC SORT DATA = WORK.A3 OUT = WORK.A5 NODUPKEY;**

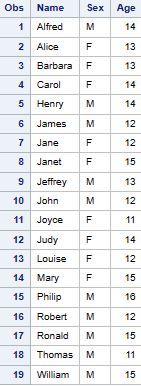
**BY NAME;**

**RUN;**

**/\* PROC PRINT DATA = WORK.A5; RUN; \*/**

**(If PROC PRINT is not given Result will open in OUTPT DATA Window. If you want result to be shown in RESULTS Window, remove command prompt from PROC PRINT STATEMENT)**

**OP:**

****

**DATA WORK.A4;**

**SET SASHELP.CLASS;**

**DROP SEX AGE;**

**RUN;**

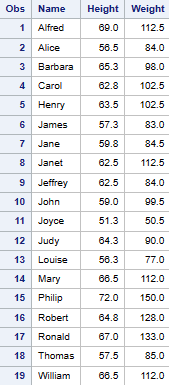
**/\* PROC PRINT DATA = WORK.A4; RUN; \*/**

**PROC SORT DATA = WORK.A4 OUT = WORK.A6 NODUPKEY;**

**BY NAME;**

**RUN;**

**OP:**

****

**MERGING**

**DATA WORK.A7;**

**MERGE WORK.A5 WORK.A6;**

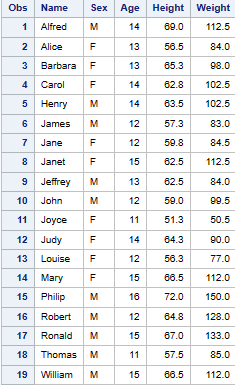
**BY NAME;**

**RUN;**

**PROC PRINT DATA = WORK.A7;**

**RUN;**

**OP:**

****

**MERGING – CARTESIAN PRODUCT (in between like comb)**

**DATA WORK.A8;**

**SET WORK.A5 WORK.A6;**

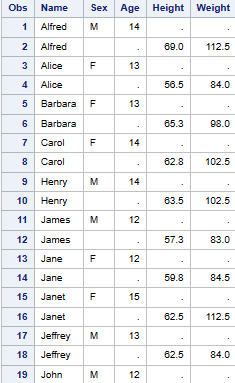
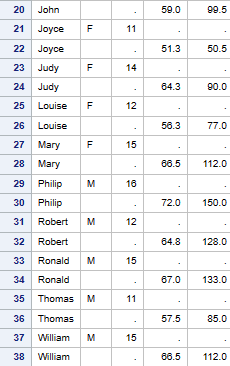
**BY NAME;**

**RUN;**

**PROC PRINT DATA = WORK.A8;**

**RUN;**

**OP:**

** **

**FUNCTIONS**

**DATA WORK.B1;**

**SET SASHELP.CLASS;**

**A1 = NAME;**

**A2 = AGE;**

**A3 = 'INDiA';**

**A4 = 20;**

**B1 = AGE + 40;**

**B2 = AGE + A4;**

**B3 = AGE \*\* 2;**

**C1 = SUM(B1,B2,B3,1000);**

**C2 = MIN(B1,B2,B3,1000);**

**C3 = MAX(B1,B2,B3,1000);**

**C4 = STD(B1,B2,B3,1000);**

**D1 = UPCASE(NAME);**

**D2 = LOWCASE(NAME);**

**D3 = REVERSE(NAME);**

**D4 = (NAME)||(SEX);**

**E1 = SIN(AGE);**

**E2 = LOG(AGE);**

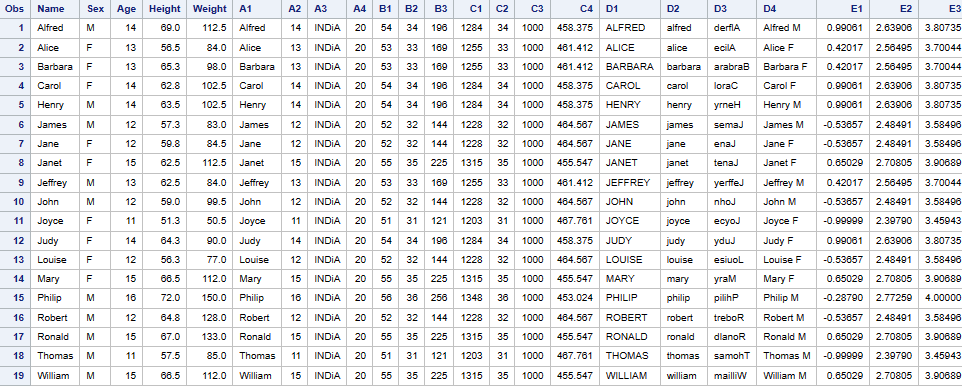
**E3 = LOG2(AGE);**

**RUN;**

**PROC PRINT DATA = WORK.B1;**

**RUN;**

**OP:**

****

**TO ADD PERIOD OF DAYS – DATE:**

data temp2;

input id 1@3 date mmddyy11.; (date format can be changed, this is the format of the datalines)

new\_month = intnx('month',date,1); (1 implies no:months to be added)

two\_days = intnx('day',date,2); (2 implies no:days to be added. For Subtraction, negative integers should be used)

cards;

1 01/27/2003

2 06/26/2002

3 03/30/2030

;

run;

proc print data = temp2;

format date date9.;

format date new\_month date9.;

format two\_days date9.;

run;

**OP:**



**CONDITIONS**

**/\* CONDITIONAL PROCESSING \*/**

**DATA WORK.D1;**

**SET SASHELP.CLASS;**

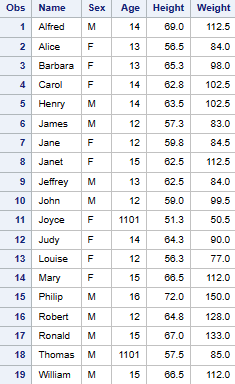
**IF AGE = 11 THEN AGE = 1101;**

**RUN;**

**PROC PRINT DATA=WORK.D1;**

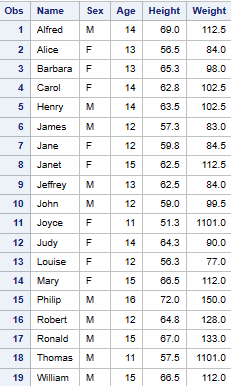
**RUN;**

**OP:**

****

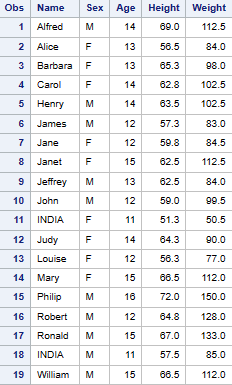
**IF AGE = 11 THEN weight = 1101;**

**OP:**

****

**IF AGE = 11 THEN NAME = 'INDIA';**

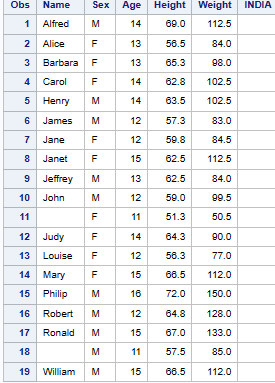
**OP:**

****

**IF AGE = 11 THEN NAME = INDIA;**

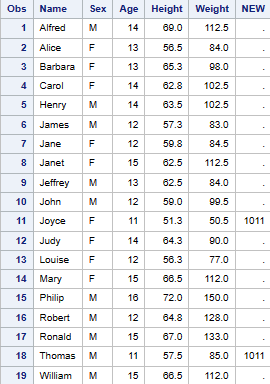
**(when INDIA is not given within quotations, NAME of age 11 is removed and new variable INDIA is created) [NOT USED]**

**OP:**

****

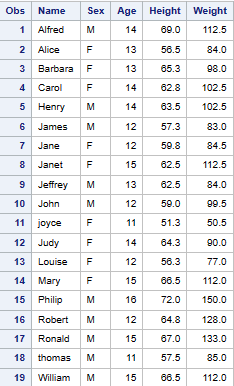
**IF AGE = 11 THEN NEW = AGE+1000;**

**OP:**

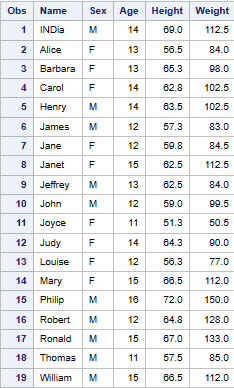
****

**IF AGE = 11 THEN NAME = LOWCASE(NAME);**

**OP:**

****

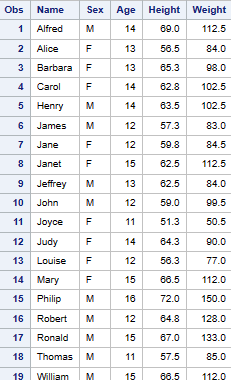
**IF NAME = 'Alfred' THEN NAME = 'INDia';**

****

**IF NAME = 'ALFRED' THEN NAME = 'INDia';**

**(there is no ALFRED in Dataset. So, there is no change in the OP)**

**OP:**

****

**OPERATORS**

**(>) – GT(Greater Than) (<) – LT(Lesser Than)**

**>= - GE(Greater Than or Equal to) <= - LE(Lesser Than or Equal to)**

**= - EQ(Equal to) ~= - NE(Not Equal to)**

**DATA WORK.D1;**

**SET SASHELP.CLASS;**

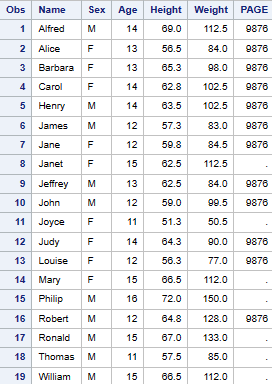
**IF AGE GT 11 AND AGE <=14 THEN PAGE = 9876;**

**RUN;**

**PROC PRINT DATA = WORK.D1;**

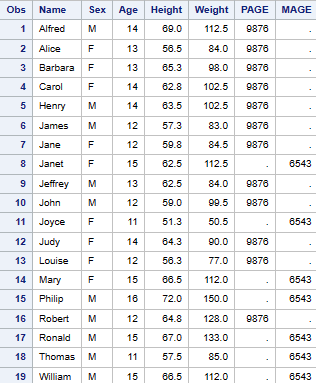
**RUN;**

**OP:**

****

**IF AGE GT 11 AND AGE <=14 THEN PAGE = 9876;**

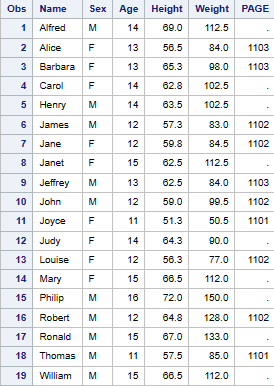
**ELSE MAGE = 6543;**

****

**IF AGE = 11 THEN PAGE = 1101;**

**ELSE IF AGE = 12 THEN PAGE = 1102;**

**ELSE IF AGE = 13 THEN PAGE = 1103;**

****

**FILTERS**

**(To create more separate Datasets with filters using conditional statements)**

**PROGRAM for create 2 Datasets for SEX – M & F**

**DATA WORK.D1 WORK.D2 WORK.D3;**

**SET SASHELP.CLASS;**

**IF SEX = 'M' THEN OUTPUT WORK.D1;**

**ELSE IF SEX = 'F' THEN OUTPUT WORK.D2;**

**ELSE OUTPUT WORK.D3;**

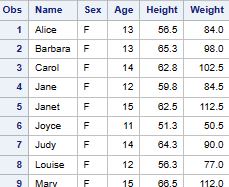
**RUN;**

**PROC PRINT DATA = WORK.D1;RUN;**

**PROC PRINT DATA = WORK.D2;RUN;**

**PROC PRINT DATA = WORK.D3;RUN;**

**OP:**

** **

**WHERE CONDITIONS:**

**WHERE condition can be given in both DATA and PROC Statements.**

**When it is given in DATA, it’ll go permanently. [NOTE: not to be used]**

**When it is given in PROC, it’ll go only in Output.**

**DON’T TRY THE BELOW PROGRAM**

**PROGRAM for creating Dataset with AGE 12 only. [don’t try as data will be lost permanently]**

**(WHERE given in DATA. So, except age 12, rest are deleted permanently)**

**DATA WORK.D1;**

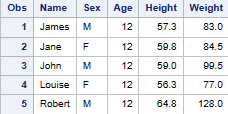
**SET SASHELP.CLASS;**

**WHERE AGE = 12;**

**RUN;**

**PROC PRINT DATA = WORK.D1;RUN;**

**OP:**

****

**PROGRAM for creating Dataset with AGE 12 only.**

**(WHERE given in PROC. Only filtered data will be shown in Output)**

**data work.d1;**

**set sashelp.class;**

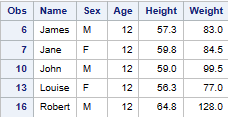
**run;**

**proc print data = work.d1;**

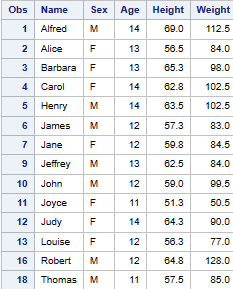
**where age = 12;**

**run;**

**OP:**

****

**where age between 11 and 14;**

****

**% is used as PREFIX or SUFFIX**

**A% => Word that starts with Uppercase A.**

**%e => Word that ends with Lowercase e.**

**%u% => Word that has u in between.**

**PROGRAM for WORDS containing specific alphabets used as filters.**

**DATA WORK.D1;**

**SET SASHELP.CLASS;**

**RUN;**

**PROC PRINT DATA = WORK.D1;**

**WHERE NAME LIKE 'A%';**

**RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (92).png**

**LOOPS**

**DO LOOP:**

**DATA WORK.L1;**

**A = 0;**

**DO X = 1 TO 10 BY 1;**

**A = A+5;**

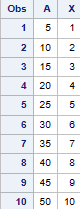
**OUTPUT; (OUTPUT Statement shows the entire loop process in OP)**

**END;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

****

**PROGRAM without OUTPUT Statement.**

**(So, doesn’t show the entire loop process. Shows only the SUM)**

**DATA WORK.L1;**

**A = 0;**

**DO X = 1 TO 10 BY 1;**

**A = A+5;**

**END;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

**(In the Loop of X, 1 is always added to the existing sum of X. So, consider the SUM as Total X – 1.)**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (94).png**

**DATA WORK.L1;**

**A = 0;**

**DO X = -10 TO 10 BY 2;**

**A = A+5;**

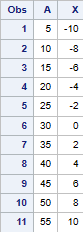
**OUTPUT;**

**END;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

****

**DO WHILE LOOP: [ = cannot be used for DO WHILE]**

**(Runs when True, stop when False)**

**DATA WORK.L1;**

**A = 0;**

**DO WHILE (A < 25);**

**A = A+5;**

**OUTPUT;**

**PUT A=;**

**END;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP: C:\Users\DELL\Pictures\Screenshots\Screenshot (96).png**

**DO WHILE (A <= 25); [ LE - <= ]**

**(when = is given, 30 gets included)**

**[1 loop is included additionally when <= is given]**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (99).png**

**A = 100; [ GE - >= ]**

**DO WHILE (A >= 50); (45 gets included) [1 additional loop is included]**

**A = A-5;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (100).png**

**DO UNTIL LOOP: [ = condition is preffered to use ]**

**(Runs when False, stops when True)**

**DATA WORK.L1;**

**A = 0;**

**DO UNTIL (A = 50);**

**A = A+5;**

**OUTPUT;**

**PUT A=;**

**END;**

**RUN;**

**PROC PRINT DATA = WORK.L1;**

**RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (97).png**

**ARRAYS**

**Arrays are VARIABLES having multiple values.**

**INDEXED ARRAY:**

**One ARRAY can have multiple loops as it is indexed.**

**DATA WORK.L1;**

**SET SASHELP.CLASS;**

**ARRAY H{3} AGE HEIGHT WEIGHT;**

**H(1) = H(1)+5000;**

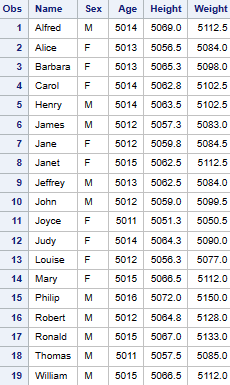
**H(2) = H(2)+5000;**

**H(3) = H(3)+5000;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

****

**INDEXED ARRAY – NUMERIC:**

**DATA WORK.L1;**

**SET SASHELP.CLASS;**

**ARRAY H{3} AGE HEIGHT WEIGHT;**

**DO X = 1 TO 3;**

**H(X) = H(X)+5000;**

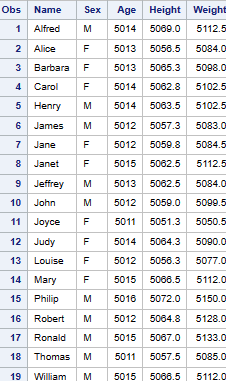
**END;**

**DROP X; (DROP statement can be used to remove any column)**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

****

**INDEXED ARRAY – STRING:**

**(For Array, $ position used for string should be before Variable name)**

**DATA WORK.L1;**

**SET SASHELP.CLASS;**

**ARRAY H{2} $NAME SEX;**

**DO X = 1 TO 2;**

**H(X) = LOWCASE(H(X));**

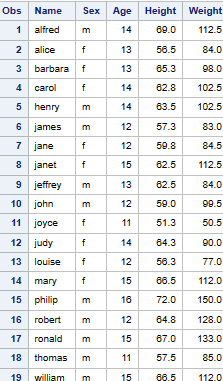
**END;**

**DROP X;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

****

**NON-INDEXED ARRAY – NUMERIC:**

**One array can have only one loop.**

**DATA WORK.L1;**

**SET SASHELP.CLASS;**

**ARRAY SP AGE HEIGHT WEIGHT;**

**DO OVER SP;**

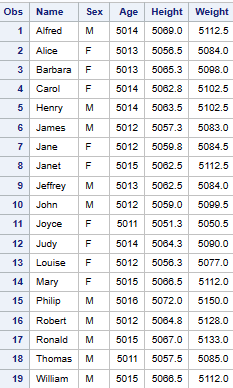
**SP = SP+5000;**

**END;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

****

**NON-INDEXED ARRAY – STRING:**

**(For Array, $ position used for string should be before Variable name)**

**DATA WORK.L1;**

**SET SASHELP.CLASS;**

**ARRAY SP $NAME SEX;**

**DO OVER SP;**

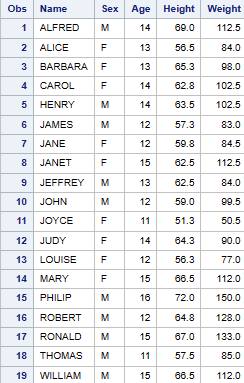
**SP = UPCASE(SP);**

**END;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**OP:**

****

**FIBONACCI SERIES**

**(To generate FIBONACCI SERIES)**

**FIBONACCI SEQUENCE is a CUMMULATIVE SEQUENCE**

**DATA WORK.A1;**

**A = 0;**

**B = 1;**

**DO E = 1 TO 20 BY 1;**

**C = A+B;**

**A = B;**

**B = C;**

**OUTPUT;**

**END;**

**DROP E;**

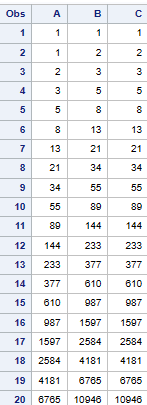
**RUN;**

**PROC PRINT DATA = WORK.A1; RUN;**

**This is how FIBONACCI series is generated.**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| **0** | **1** | **1** |
| **A = B** | **B = C** | **A+B** |
| **1** | **1** | **2** |
| **1** | **2** | **3** |

**OP:**

****

**LIST INPUT**

**(;) SEMICOLON is not given for each DATALINE. Instead, given in end.**

**DATA WORK.A1;**

**INPUT ENAME$ ENO ESAL;**

**DATALINES;**

**SAKURA 15 80000**

**WOLFY 18 150000**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1; RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (106).png**

**SPECIAL INPUT:**

**DATA WORK.A1;**

**INPUT ENAME$ ENO ESAL;**

**DATALINES4; (DATALINES4 is FORMAT)**

**SAKURA 15 80000**

**WOLFY 18 150000**

**;;;; (4 semicolon are FORMAT)**

**RUN;**

**PROC PRINT DATA = WORK.A1; RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (106).png**

**TRAILING INPUT: (when DATALINES are given in a single line)**

**DATA WORK.A1;**

**INPUT ENAME$ ENO ESAL @@; (@@ FORMAT)**

**DATALINES;**

**SAKURA 15 80000 WOLFY 18 150000**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1; RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (106).png**

**MIXED INPUT: (when DATALINES are in mixed lines)**

**DATA WORK.A1;**

**INPUT ENAME$ ENO ESAL;**

**DATALINES;**

**SAKURA 15**

**80000**

**WOLFY 18**

**150000**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1; RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (106).png**

**COLUMN INPUT: (when DATALINES are given without spaces)**

**DATA WORK.A1;**

**INPUT ENAME$ 1-6 ENO 7-8 ESAL 9-14; (width should given properly)**

**DATALINES;**

**SAKURA1580000**

**WOLFY 18150000**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1; RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (106).png**

**POINTER INPUT: (when DATALINES are given without spaces)**

**(starting position is given in @no and width is given)**

**DATA WORK.A1;**

**INPUT @1 ENAME $6. @7 ENO 2. @9 ESAL 6.;**

**DATALINES;**

**SAKURA1580000**

**WOLFY 18150000**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1; RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (106).png**

**READING FROM EXTERNAL FILE**

**DATASET OPTIONS**

**DATA WORK.SP1;**

**SET SASHELP.CLASS;**

**KEEP NAME SEX AGE;**

**RUN;**

**PROC PRINT DATA=WORK.SP1; RUN;**

**(OR)**

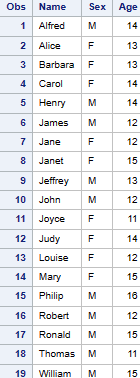
**DATA WORK.SP1 (KEEP=NAME SEX AGE);**

**SET SASHELP.CLASS;**

**RUN;**

**PROC PRINT DATA=WORK.SP1; RUN;**

**OP:**

****

**DATA WORK.SP1;**

**SET SASHELP.CLASS;**

**DROP SEX AGE;**

**RUN;**

**PROC PRINT DATA=WORK.SP1; RUN;**

**(OR)**

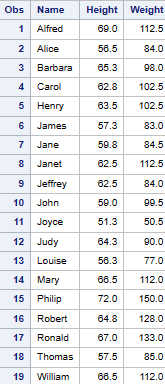
**DATA WORK.SP1 (DROP = SEX AGE);**

**SET SASHELP.CLASS;**

**RUN;**

**PROC PRINT DATA=WORK.SP1; RUN;**

**OP:**

****

**TO CHANGE VARIABLE NAME THROUGH DATASET OPTION – RENAME:**

**DATA WORK.SP1;**

**SET SASHELP.CLASS;**

**RENAME NAME = ENAME;**

**RENAME SEX = GENDER;**

**RUN;**

**PROC PRINT DATA=WORK.SP1; RUN;**

**(OR)**

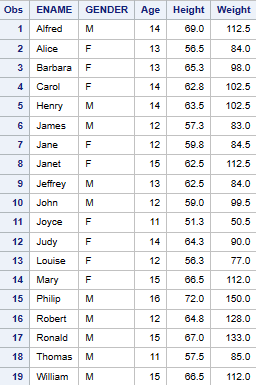
**DATA WORK.SP1 (RENAME = NAME = ENAME RENAME = SEX = GENDER);**

**SET SASHELP.CLASS;**

**RUN;**

**PROC PRINT DATA=WORK.SP1; RUN;**

**OP:**



**LABEL: (Label creates a name to the DATASET TABLE in the properties)**

**DATA WORK.L1 (LABEL = 'DETAILS');**

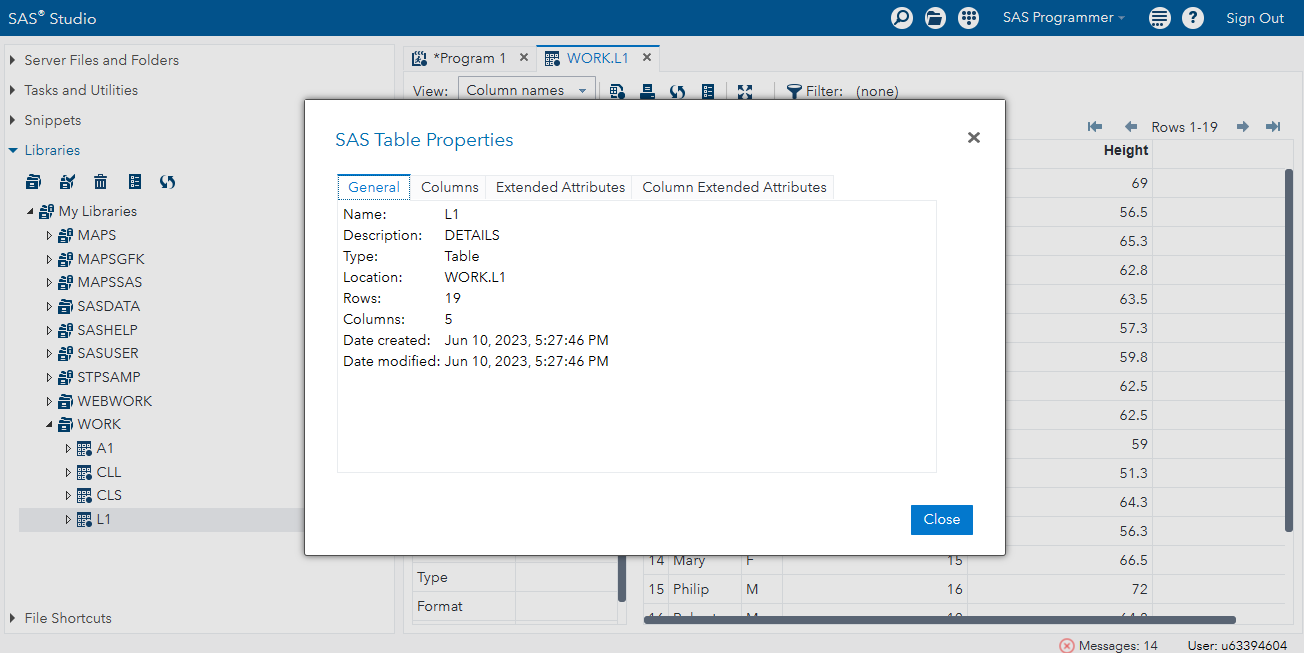
**SET SASHELP.CLASS;**

**RUN;**

**PROC PRINT DATA = WORK.L1;**

**RUN;**

**Libraries -> Work -> Right click L1 -> Properties -> DESCRIPTION NAME**

****

**PASSWORD:**

**DATA WORK.L1 (PW = AAA); (PW => Password)**

**SET SASHELP.CLASS;**

**RUN;**

**PROC PRINT DATA = WORK.L1; RUN;**

**To bring DATASET through PROC STATEMENT, without using DATA STEP**

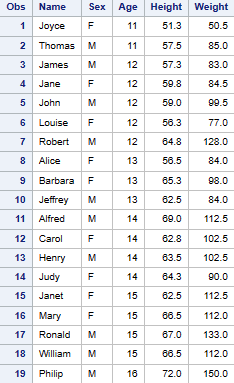
**PROC SORT DATA = SASHELP.CLASS OUT = WORK.CLS;**

**BY AGE;**

**RUN;**

**proc print data = work.cls; run;**

**OP:**

****

**DATA WORK.CLL; (FIRST DOT & LAST DOT)**

**SET WORK.CLS;**

**BY AGE;**

**FIRAGE = FIRST.AGE;**

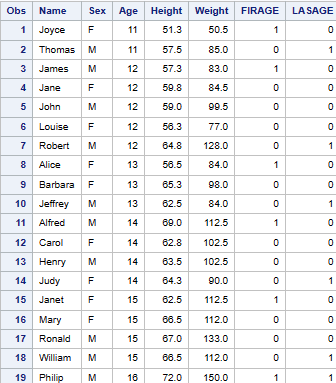
**LASAGE = LAST.AGE;**

**RUN;**

**PROC PRINT DATA = WORK.CLL;**

**RUN;**

**OP:**

****

**PROCEDURE – FOR ALL PROCS (format)**

**PROC PROCEDURENAME DATA=LIBRARY.DATASET PROC\_OPTIONS;**

**STATEMENT1 VARIABLES/STATEMENT1\_OPTIONS;**

**STATEMENT2 VARIABLES/STATEMENT2\_OPTIONS;**

**STATEMENT3 VARIABLES/STATEMENT3\_OPTIONS;**

**-------------------------------------------**

**STATEMENTN VARIABLES/STATEMENTN\_OPTIONS;**

**RUN;**

**QUIT;**

**PROCEDURE PRINT\_OPTIONS**

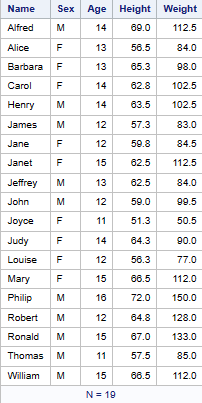
**NOOBS – No Observation column**

**N – Gives the No of Observation in bottom**

**PROC PRINT DATA = SASHELP.CLASS NOOBS N;**

**RUN;**

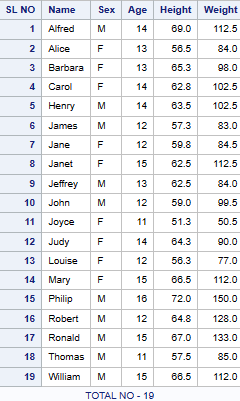
**OP:**

****

**PROC PRINT DATA = SASHELP.CLASS OBS='SL NO' N='TOTAL NO - ';**

**RUN;**

**OP:**

****

**WIDTH OPTIONS – FULL UNIFORM MIN**

**(DOUBLE – Double Line Spacing WIDTH – Column Width)**

**FORMAT:**

**PROC PRINT DATA = LIBRARYNAME.DATASET NAME**

**DOUBLE WIDTH = FULL/UNIFORM/MIN;**

**RUN;**

**FULL: (FULL WIDTH shows the entire decimal number)**

**DATA WORK.A1;**

**INPUT ENAME $ NO;**

**DATALINES;**

**SAKURA 1234.567**

**WOLFY 123.456**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1**

**DOUBLE WIDTH = FULL;**

**RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (118).png**

**UNIFORM: (UNIFORM WIDTH shows the round off of decimal number)**

**DATA WORK.A1;**

**INPUT ENAME $ NO;**

**DATALINES;**

**SAKURA 1234.567**

**WOLFY 123.456**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1**

**DOUBLE WIDTH = UNIFORM;**

**RUN;**

**OP:**

**C:\Users\DELL\Pictures\Screenshots\Screenshot (119).png**

**MIN: (MIN WIDTH shows the round off of decimal number)**

**DATA WORK.A1;**

**INPUT ENAME $ NO;**

**DATALINES;**

**SAKURA 1234.567**

**WOLFY 123.456**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1**

**DOUBLE WIDTH = MIN;**

**RUN;**

**OP:**

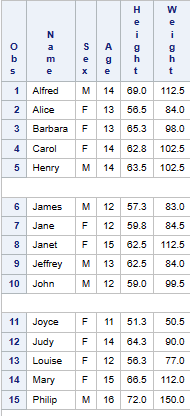
**C:\Users\DELL\Pictures\Screenshots\Screenshot (119).png**

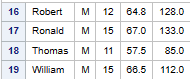
**PROC PRINT DATA = SASHELP.CLASS**

**BLANKLINE = 5 HEADING =V; (V – Vertical)**

**RUN;**

**OP:**





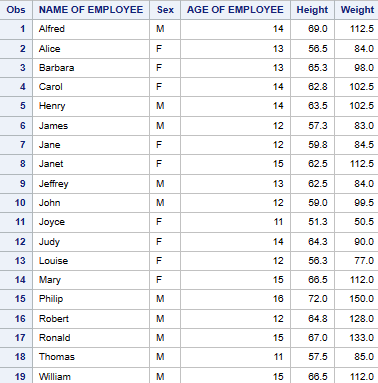
**TO CHANGE VARIABLE NAME THROUGH PROC OPTION – LABEL:**

**PROC PRINT DATA = SASHELP.CLASS LABEL;**

**LABEL NAME = 'NAME OF EMPLOYEE' AGE = 'AGE OF EMPLOYEE';**

**RUN;**

**OP:**

****

**PROCEDURE PRINT – STATEMENTS**:

**VAR – To print the list of Variables in a specific order.**

**ID – Replaces OBS.**

**SUM – Works only with Numeric Data.**

**PROC PRINT DATA = SASHELP.CLASS;**

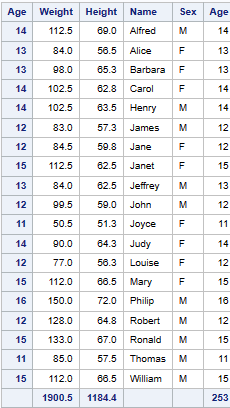
**VAR WEIGHT HEIGHT NAME SEX AGE;**

**ID AGE;**

**SUM AGE HEIGHT WEIGHT;**

**RUN;**

**OP:**

****

**PROCEDURE PRINT – STATEMENTS CONTINUES**

**PROC SORT DATA=SASHELP.CLASS OUT=WORK.CLS;**

**BY AGE;**

**RUN;**

**PROC PRINT DATA = WORK.CLS; (To print the entire Dataset once)**

**RUN;**

**PROC PRINT DATA=WORK.CLS;**

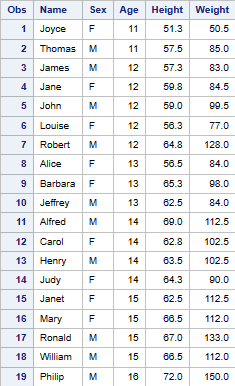
**BY AGE;**

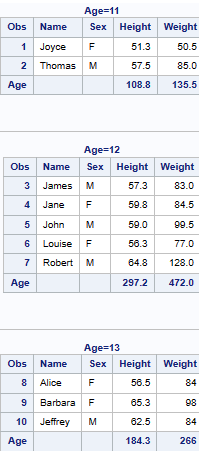
**PAGEBY AGE; (PAGEBY - To print in Different PAGES)**

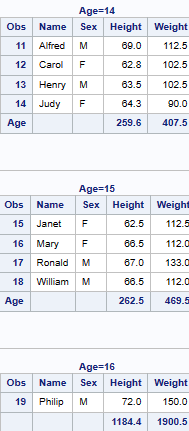
**SUMBY AGE;**

**RUN;**

**OP:**

****

****

****

**PROCEDURE SORT**

**NODUP – Removes Entire KEY if there is DUPLICATES.**

**NODUPKEY – Checks for and eliminates observations with duplicate BY values.**

**It must be a primary key – BY KEY is the PRIMARY KEY(BY KEY must be sorted)**

**PROC SORT DATA = SASHELP.CLASS**

**NODUP NODUPKEY OUT = WORK.CLS;**

**BY DESCENDING AGE;**

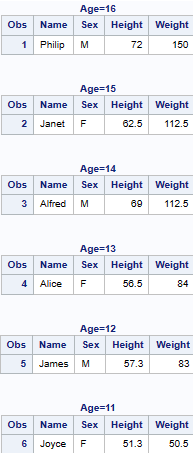
**RUN;**

**PROC PRINT DATA = WORK.CLS;**

**BY descending AGE;**

**RUN;**

**OP:**

****

**PROCEDURE CONTENTS**

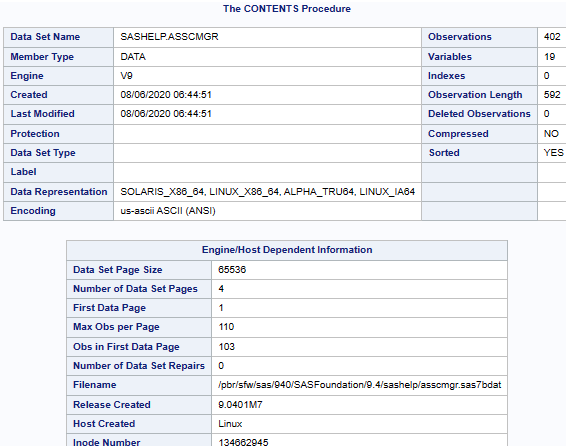
**The CONTENTS procedure shows the contents of a SAS dataset and prints the directory of the SAS library.**

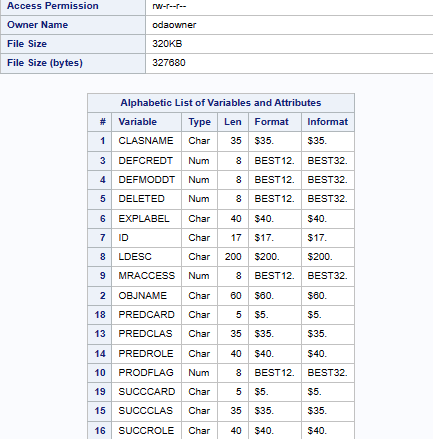
**­SHOWS ALPHABETICAL ORDER OF VARIABLES:**

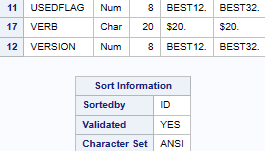
**PROC CONTENTS DATA = SASHELP.ASSCMGR;**

**RUN;**

**OP:**

****

****

****

**VARNUM: SHOWS CREATION ORDER OF VARIABLES**

**PROC CONTENTS DATA = SASHELP.ASSCMGR VARNUM;**

**RUN;**

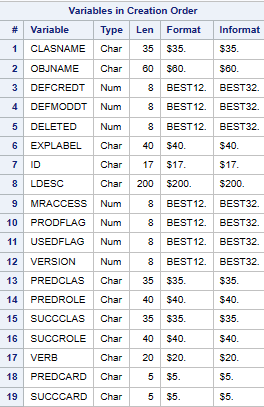
**OP:**

**The Contents procedure**

**Engine/Host Dependent Information**

**Sort Information**

**All the above 3 are common in the Procedure Content.**

****

**POSITION: SHOWS ALPHABETICAL AND CREATION ORDER OF VARIABLES**

**PROC CONTENTS DATA = SASHELP.ASSCMGR POSITION;**

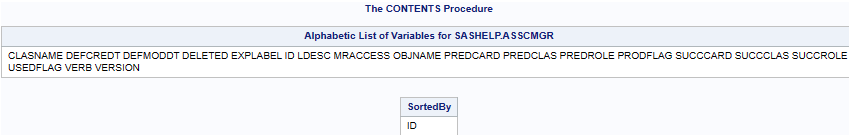
**RUN;**

**SHORT: SHOWS ONLY VARIABLES**

**PROC CONTENTS DATA = SASHELP.ASSCMGR SHORT;**

**RUN;**

**OP:**

****

**NOPRINT:**

**PROC CONTENTS DATA = SASHELP.ASSCMGR NOPRINT OUT=WORK.LL1;**

**RUN;**

**PROC PRINT DATA=WORK.LL1;**

**RUN;**

**PROCEDURE COMPARE**

**There should be 2 Datasets to compare.**

**Only Numeric Values can be Compared.**

**DATASET 1:**

**DATA WORK.A1;**

**INPUT ENAME$ ENO ESAL;**

**DATALINES;**

**SAKURA 15 35000**

**WOLFY 18 50000**

**;**

**RUN;**

**PROC PRINT DATA = WORK.A1;**

**RUN;**

**DATASET 2:**

**DATA WORK.A2;**

**INPUT ENAME$ ENO ESAL;**

**DATALINES;**

**BLAKLEY 1 80000**

**STARLEY 2 60000**

**;**

**RUN;**

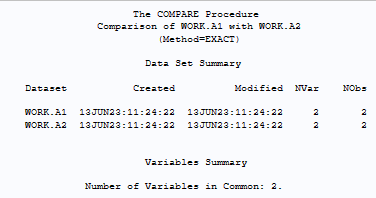
**PROC PRINT DATA = WORK.A2;**

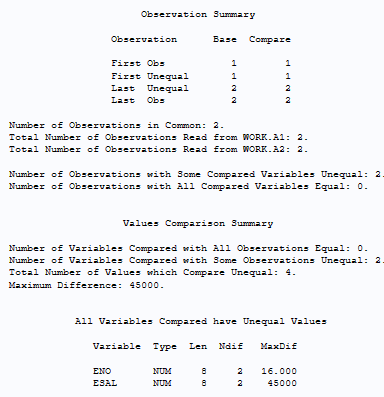
**RUN;**

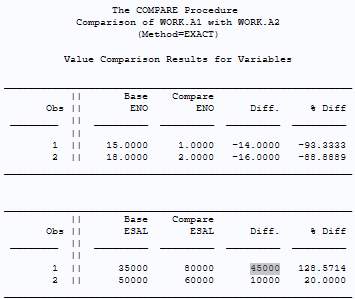
**PROC COMPARE BASE = WORK.A1 COMPARE = WORK.A2;**

**RUN;**

**OP:**

****

****



**PROCEDURE TRANSPOSE (VICE-VERSA)**

**TRANSPOSE generally transposes Numeric Data when no VAR STATEMENT is given.**

**Without VAR STATEMENT:**

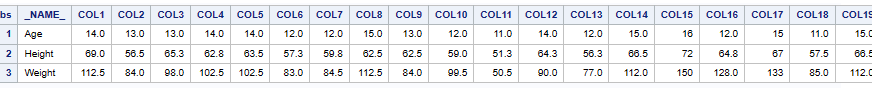
**PROC TRANSPOSE DATA = SASHELP.CLASS OUT = WORK.TR1;**

**RUN;**

**PROC PRINT DATA = WORK.TR1;**

**RUN;**

**OP:**



**TRANSPOSE – CHARACTER [ \_CHARACTER\_ ]**

**PROC TRANSPOSE DATA = SASHELP.CLASS OUT = WORK.TR1;**

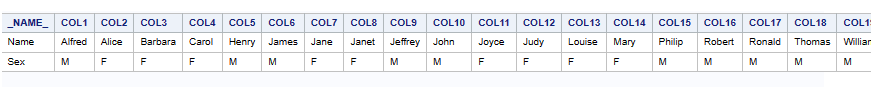
**VAR \_CHARACTER\_;**

**RUN;**

**PROC PRINT DATA = WORK.TR1;**

**RUN;**

**OP:**

****

**TRANSPOSE – CHARACTER [ \_NUMERIC\_ ]**

**PROC TRANSPOSE DATA = SASHELP.CLASS OUT = WORK.TR1;**

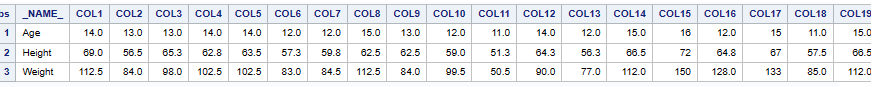
**VAR \_NUMERIC\_;**

**RUN;**

**PROC PRINT DATA = WORK.TR1;**

**RUN;**

**OP:**

****

**TRANSPOSE – ALL: [ \_ALL\_ ]**

**PROC TRANSPOSE DATA = SASHELP.CLASS OUT = WORK.TR1;**

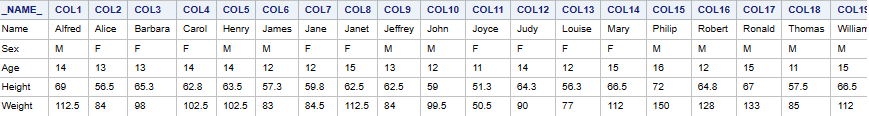
**VAR \_ALL\_;**

**RUN;**

**PROC PRINT DATA = WORK.TR1;**

**RUN;**

**OP:**

****

**TRANSPOSE – TO SELECTED VARIABLES:**

**PROC TRANSPOSE DATA = SASHELP.CLASS OUT = WORK.TR1;**

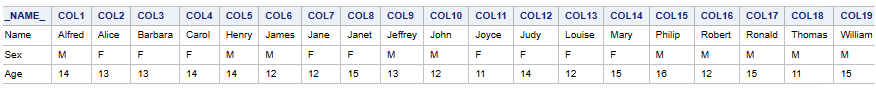
**VAR NAME SEX AGE;**

**RUN;**

**PROC PRINT DATA = WORK.TR1;**

**RUN;**

**OP:**

****

**PROCEDURE RANK**

**PROC RANK DATA = SASHELP.CLASS OUT = WORK.RANK1;**

**VAR AGE;**

**RANKS AR;**

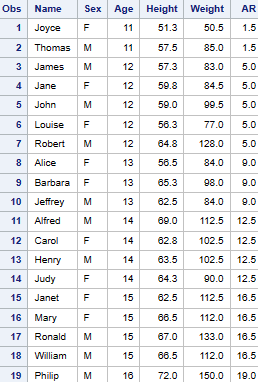
**RUN;**

**PROC SORT DATA = WORK.RANK1; BY AR;**

**RUN;**

**PROC PRINT DATA = WORK.RANK1; RUN;**

**OP:**

****

**TIES:**

**TIES => MEAN TIES => HIGH**

**PROC RANK DATA = SASHELP.CLASS OUT = WORK.RANK1 TIES = LOW DESCENDING;**

**VAR AGE;**

**RANKS AR;**

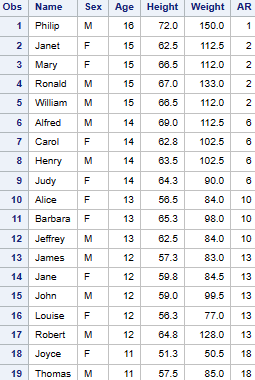
**RUN;**

**PROC SORT DATA = WORK.RANK1; BY AR;**

**RUN;**

**PROC PRINT DATA = WORK.RANK1; RUN;**

**OP:**

****

**STATISTICS**

**MEAN:**

**PROC MEANS DATA = SASHELP.CLASS;**

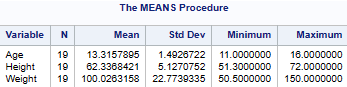
**RUN;**

**PROC SUMMARY DATA = SASHELP.CLASS PRINT;**

**VAR \_NUMERIC;**

**RUN;**

**OP:**

****

**MEAN FOR SPECIFIC VARIABLE:**

**PROC MEANS DATA = SASHELP.CLASS;**

**VAR AGE HEIGHT;**

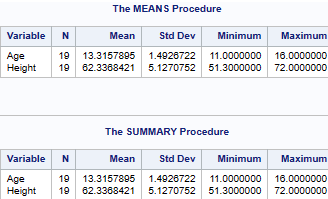
**RUN;**

**PROC SUMMARY DATA = SASHELP.CLASS PRINT;**

**VAR AGE HEIGHT;**

**RUN;**

**OP:**



**PROC MEANS DATA = SASHELP.CLASS;**

**CLASS AGE;**

**VAR AGE HEIGHT;**

**FREQ AGE;**

**WEIGHT AGE;**

**OUTPUT OUT = WORK.MN;**

**RUN;**

**PROC SUMMARY DATA = SASHELP.CLASS PRINT;**

**CLASS AGE;**

**VAR AGE HEIGHT;**

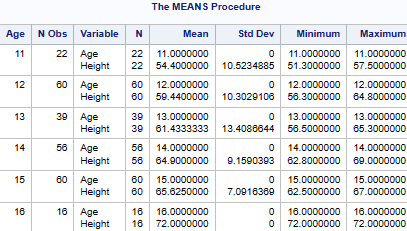
**FREQ AGE;**

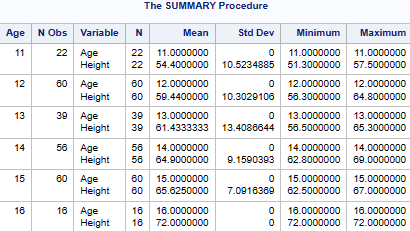
**WEIGHT AGE;**

**OUTPUT OUT = WORK.MN;**

**RUN;**

**OP:**

****

****

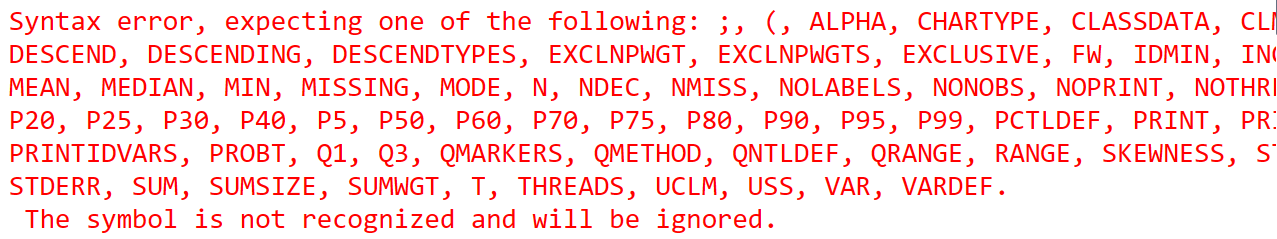
**When you want to know all the Statistical Options**

**PROC MEANS DATA = SASHELP.CLASS ?; (Type ?)**

**RUN;**

**OP:**

**? gives you all the Statistical Options. You can choose whatever you’re in need.**

****

**NMISS – NO OF OBSERVATIONS THAT ARE MISSING**

**SUMWGT – SUM OF WEIGHTS**

**USS – UNCORRECTED SUM OF SQUARES**

**CSS - SUM OF SQUARES CORRECTED FOR THE MEAN**

**UCLM - ONE-SIDED CONFIDENCE LIMIT ABOVE THE MEAN**

**SKEW / SKEWNESS - MEASURES THE TENDENCY OF THE DEVIATIONS TO BE LARGER IN ONE DIRECTION THAN IN THE OTHER.**

**KURT - MEASURES HEAVINESS OF TAILS.**

**There are so many options like these in the above.**

**PROGRAM:**

**PROC MEANS DATA = SASHELP.CLASS N NMISS SUM MIN MAX MEAN STD SUMWGT MEDIAN USS CSS UCLM SKEW KURT P10 P25 P50 P75 P90 P95 P99Q1 Q3;**

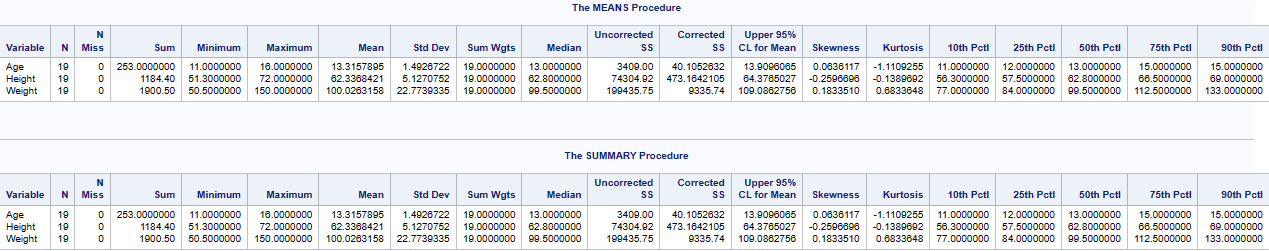
**RUN;**

**PROC SUMMARY DATA = SASHELP.CLASS PRINT N NMISS SUM MIN MAX MEAN STD SUMWGT MEDIAN USS CSS UCLM SKEW KURT P10 P25 P50 P75 P90 P95 P99Q1 Q3;**

**VAR \_NUMERIC\_;**

**RUN;**

**OP:**

****

**PROCEDURE – UNIVARIATE (Descriptive Statistics)**

**PROC UNIVARIATE DATA = SASHELP.CLASS;**

**RUN;**

**PROC UNIVARIATE DATA = SASHELP.CLASS;**

**CLASS AGE;**

**VAR AGE HEIGHT;**

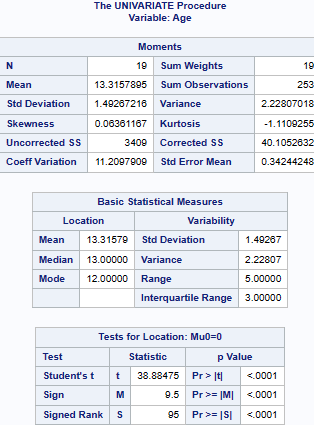
**FREQ AGE;**

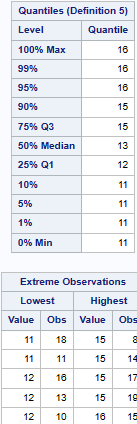
**WEIGHT AGE;**

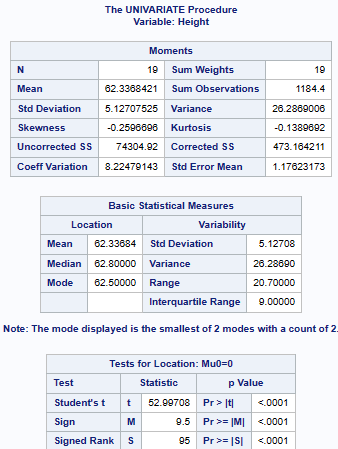
**OUTPUT OUT = WORK.MN;**

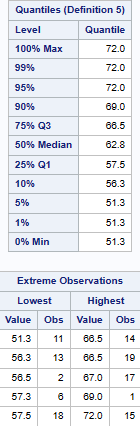
**RUN;**

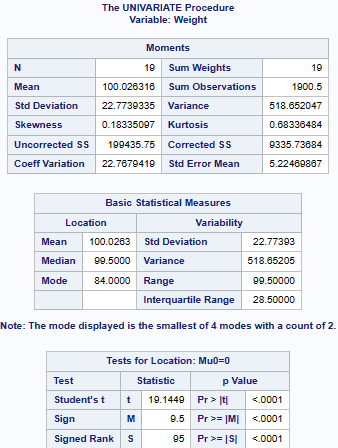
**OP:**

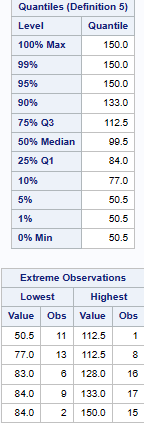












**Like the above,it gives detailed data for AGE,HEIGHT & WEIGHT**

**PROC – UNIVARIATE (GRAPHS):**

**NORMAL - Displays a normal density CURVES on the HISTOGRAM.**

**PROC UNIVARIATE DATA = SASHELP.CLASS;**

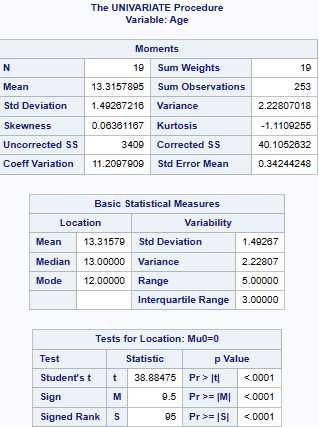
**HISTOGRAM AGE /NORMAL;**

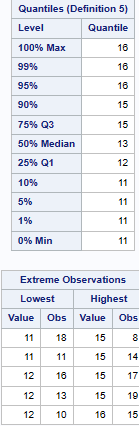
**PROBPLOT HEIGHT;**

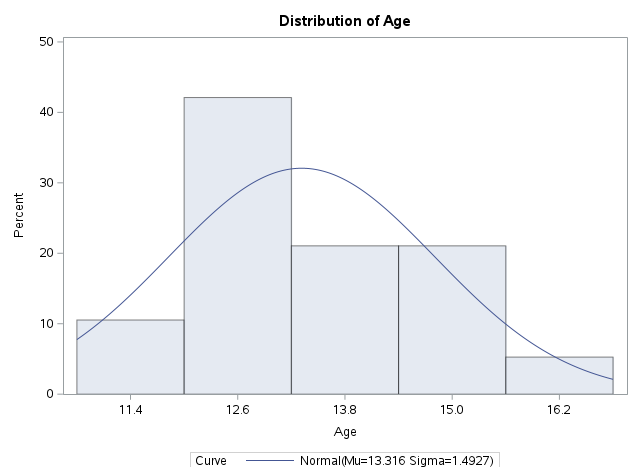
**QQPLOT WEIGHT;**

**RUN;**

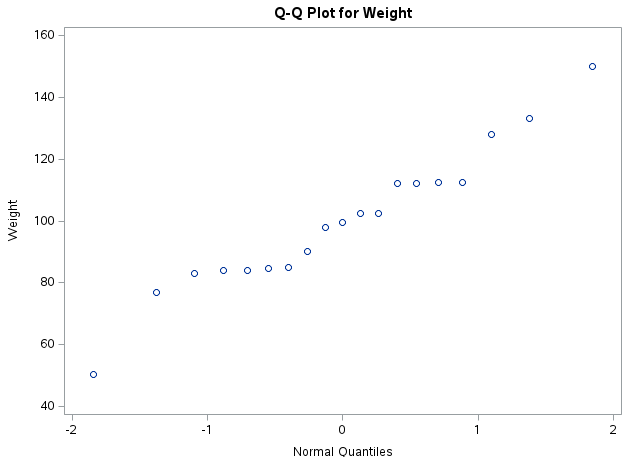
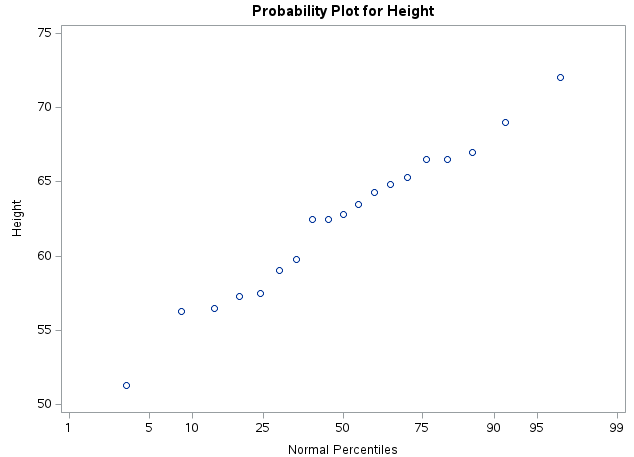
**OP:**

****

****

****

**Like the above, detailed descriptive statistics will be shown for each variable mentioned in the code. Graph will also be shown as it is included in the code.**

****

**PROCEDURE CORR (Correlation)**

**The CORR procedure computes Pearson correlation coefficients, three nonparametric measures of association, and the probabilities associated with these statistics.**

**PROC CORR DATA = SASHELP.CLASS;**

**VAR HEIGHT WEIGHT;**

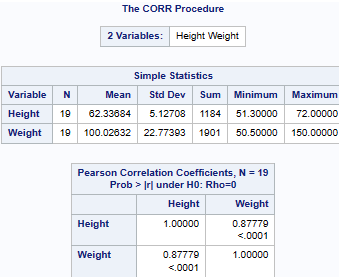
**RUN;**

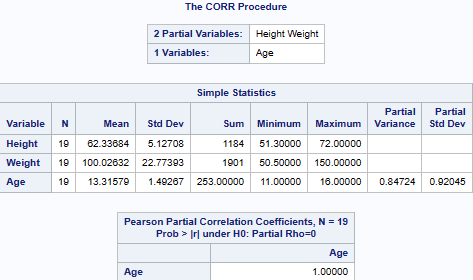
**PROC CORR DATA = SASHELP.CLASS;**

**PARTIAL HEIGHT WEIGHT;**

**RUN;**

**OP:**

****

****

**The correlation statistics include the following:**

**Pearson product-moment correlation:**

**Requests a table of Pearson product-moment correlations. The correlations range from 1 to 1.**

**Spearman rank-order correlation:**

**Requests a table of Spearman correlation coefficients based on the ranks of the variables.**

**Kendall’s tau-b coefficient:**

**Requests a table of Kendall’s τ-b coefficients based on the number of concordant and discordant pairs of observations.**

**Hoeffding’s measure of dependence:**

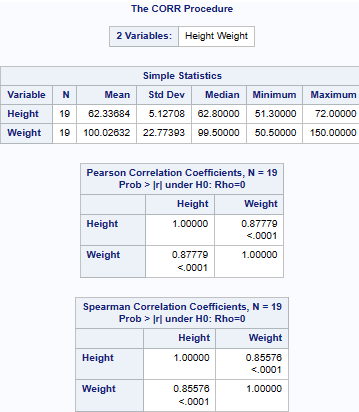
**Requests a table of Hoeffding’s statistics.**

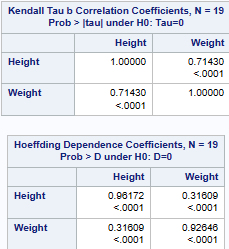
**PROC CORR DATA = SASHELP.CLASS PEARSON SPEARMAN KENDALL HOEFFDING;**

**VAR HEIGHT WEIGHT;**

**RUN;**

**OP:**

****

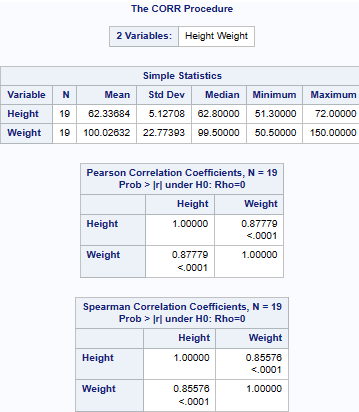
****

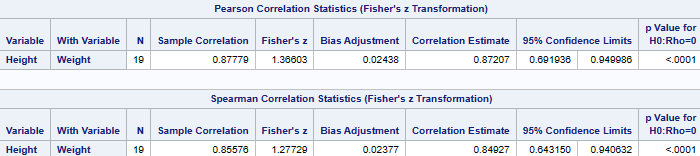
**PROC CORR DATA = SASHELP.CLASS PEARSON SPEARMAN FISHER;**

**VAR HEIGHT WEIGHT;**

**RUN;**

**OP:**

****

****

**MACROS FOR DESCRIPTIVE STATISTICS:**

**P => PEARSON**

**S =>SPEARMAN**

**K => KENDALL**

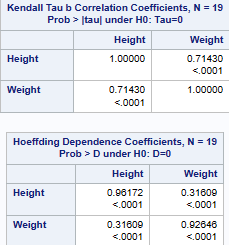
**H => HOEFFDING**

**PROC CORR DATA = SASHELP.CLASS OUT=P OUTS=S OUTK=K OUTH=H;**

**VAR HEIGHT WEIGHT;**

**RUN;**

**OP:**

****  ****

**PROCEDURE – TTEST**

**The T-TEST procedure performs T Tests and computes confidence limits for one sample, paired observations, two independent samples, and the AB/BA crossover design.**

**T-Test is a statistical test used to determine whether two population means are different when the variances are unknown and sample size is small.**

**T-Test is often used for Hypothesis testing with continuous data.**

**Population Variance is Unknown, and the sample size is Small. Population Size should be less than 30.**

**T-Test is based on T-Distribution.**

**When the Population Variance is Unknown, T-Test should be used.**

**T-Test = Population Mean – Sample Mean**

**Population Standard Deviation**

**T-Score is the result value that is derived from T-Test.**

**NULL HYPOTHESIS:**

**When the Population Means are Equal.**

**When T-Score is greater Critical Value**

**ALTERNATIVE HYPOTHESIS:**

**When the population Means are not Equal.**

**When T-Score is lesser than Critical Value.**

**PROC TTEST DATA = SASHELP.CLASS; (Gives Entire PROC)**

**RUN;**

**PROC TTEST DATA = SASHELP.CLASS; (Gives only VAR AGE HEIGHT PROC)**

**VAR AGE HEIGHT;**

**RUN;**

**OP:**

**T-Test Procedure details and Graph will be given.**

**(Age cannot be shown as age has more variables)**

**PROC TTEST DATA=SASHELP.CLASS;**

**CLASS SEX; (SEX has only 2 variables)**

**VAR AGE HEIGHT;**

**RUN;**

**OP:**

**T-Test Procedure details and Graph will be given.**

**CHI-SQUARE**

**Chi Square Test is a statistical test which is used to determine the independence between two categorical variables.**

**Chi-Square test is used to analyse data which involves Frequency counts of Categorical Variables.**

**CATEGORIAL VARIABLES:**

**Variables which can be categorized are called Categorical Variable.**

**EXAMPLE OF CATEGORIAL VARIABLE:**

**Gender**

**Age Group**

**Race**

**Educational Level**

**Categorical Data is Qualitative.**

χ2 = Σ(O− E)2

E

**Where**

**CHI SQUARE obtained = Sum of (Observed Score – Expected Score)**2

**Expected Score**

**The Hypothesis being tested for Chi-Square**

**NULL:**

**Variable A and Variable B are independent.**

**ALTERNATE:**

**Variable A and Variable B are not Independent.**

**PROC TTEST DATA = SASHELP.CLASS;**

**RUN;**

**PROC FREQ DATA = SASHELP.CLASS;**

**TABLES HEIGHT \*WEIGHT /CHISQ;**

**RUN;**

**OP:**

**T-Test Procedure details, FREQUENCY Table and Graph will be given.**

**(Type ?)**

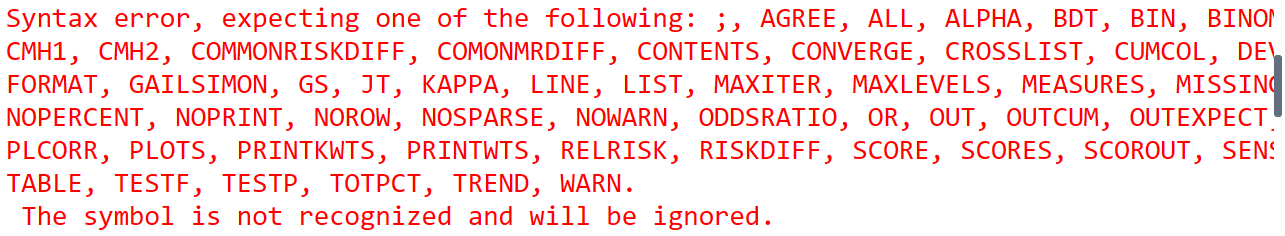
**PROC FREQ DATA = SASHELP.CLASS;**

**TABLES HEIGHT \*WEIGHT /?;**

**RUN;**

**OP:**

**? gives you all the Statistical Options. You can choose whatever you’re in need.**

****

**F-TEST**

**F-Test is a statistical test used to determine whether two population variances are equal.**

**F-Test is used to determine the equality in population variance.**

**F-Test is often used for Hypothesis testing of continuous data.**

**NULL HYPOTHESIS:**

**Population Variance are Equal.**

**ALTERNATIVE HYPOTHESIS:**

**Population Variances are not Equal.**

**CONTINUOUS DATA:**

**Continuous Data is the data that can have any value.**

**EXAMPLE:**

**Height – Height of Children**

**Weight**

**Temperature**

**Time**

**Speed**

**Length**

**F Ratio = Greater Sample Variance**

**Smaller Sample Variance**

**Critical Value = 0.05**

**F Ratio > Critical Value => Hypothesis Rejected.**

**Population Variance is not equal.**

**If the Hypothesis is rejected, it is concluded that the population variance is not equal.**

**F Ratio < Critical Value => Hypothesis Accepted.**

**Population Variance is equal.**

**If the Hypothesis is accepted, it is concluded that the population variance is equal.**

**Used 1 Tail or 2 Tail Hypothesis.**

**Z-TEST**

**Z-Test is a statistical test used to determine whether two population means are different when the variance are known and sample size is large.**

**Z-Test is often used for Hypothesis Testing with continuous data.**

**Population Variance is known and sample size is large. Population Size should be more than 30.**

**Z-Test is based on Standard Normal Distriution, which has Mean = 0 Standard Deviation = 1**

**Z-Test = Population Mean – Sample Mean**

**Sample Distribution Standard Deviation**

**Score is the result value derived from Z-Test.**

**NULL HYPOTHESIS:**

**When Z-Score is greater than Critical Value.**

**When Population Mean are Equal.**

**ALTERNATIVE HYPOTHESIS:**

**When Z-Score is lesser than Critical Value.**

**When Population Mean are not Equal.**

**PROCEDURE – REG (REGRESSION)**

**The REG procedure is one of many regression procedures in the SAS System. It is a general-purpose procedure for regression, while other SAS regression procedures provide more specialized applications.**

**REGRESSION is for Numerical and Continuous Variables.**

**MODEL**

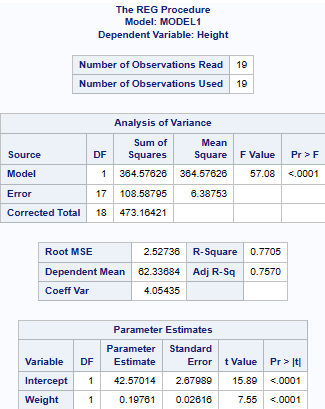
**The dependent (response) variables are specified, followed by an equal sign and the regressor variables. Variables specified in the MODEL statement must be numeric variables in the data set being analyzed.**

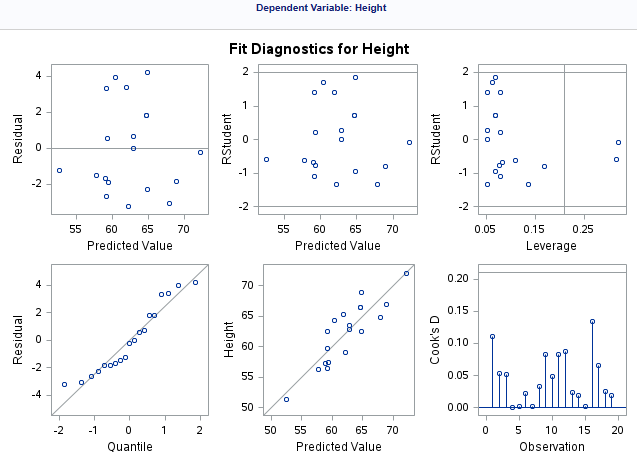
**PROC REG DATA=SASHELP.CLASS;**

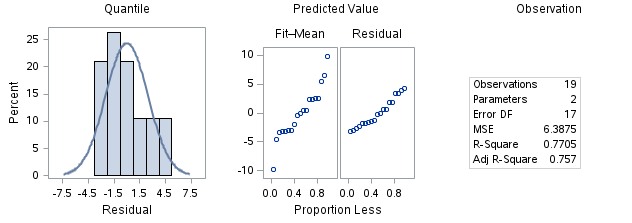
**MODEL HEIGHT=WEIGHT;**

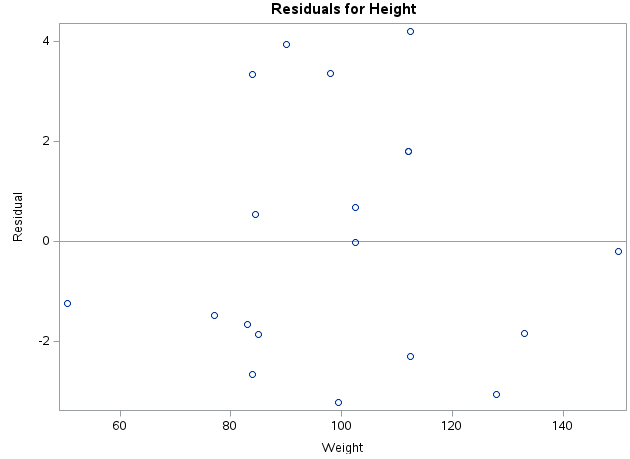
**RUN;**

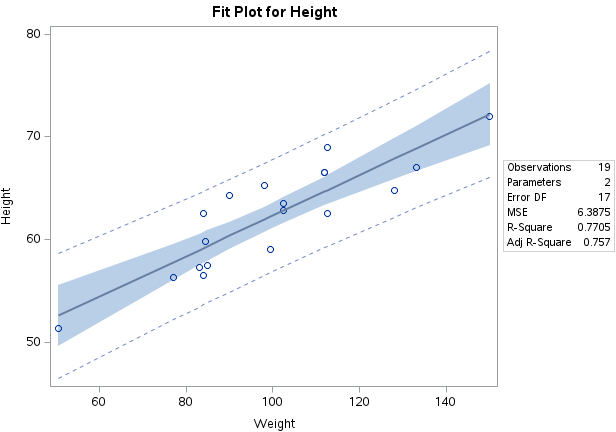
**OP:**

****

****

****



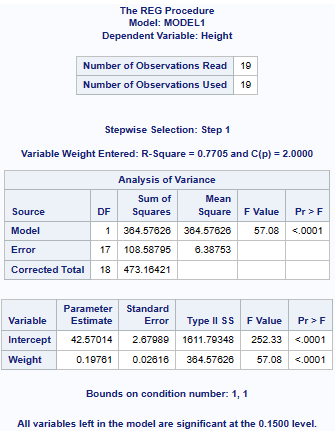
****

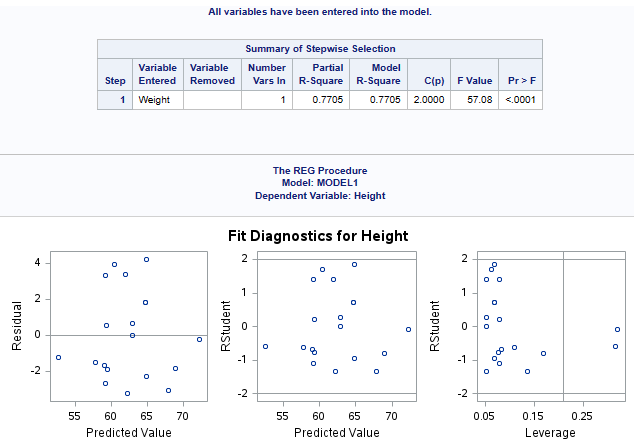
**PROC REG DATA=SASHELP.CLASS;**

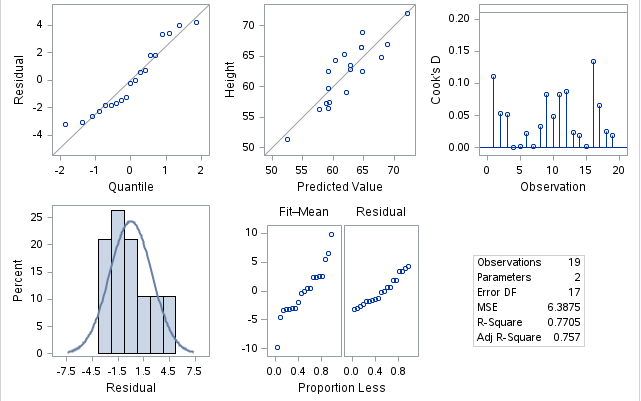
**MODEL HEIGHT=WEIGHT/SELECTION = STEPWISE;**

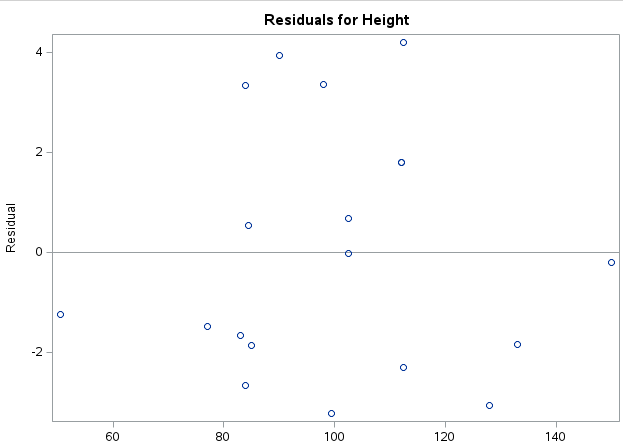
**RUN;**

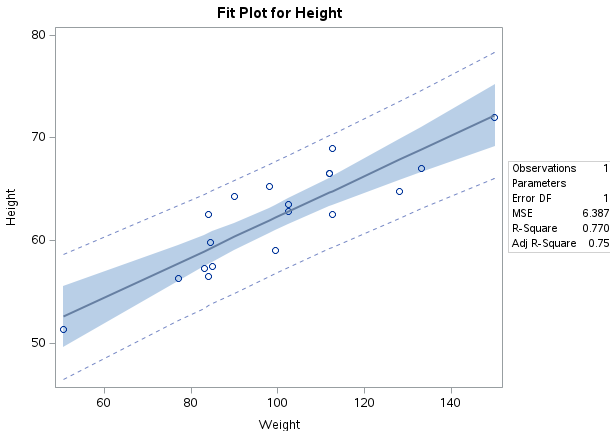
**OP:**

****

****

****

****

****

**(Type ?)**

**PROC REG DATA=SASHELP.CLASS;**

**MODEL HEIGHT=WEIGHT/SELECTION=?;**

**RUN;**

**OP:**

**? gives you all the Statistical Options. You can choose whatever you’re in need.**

****

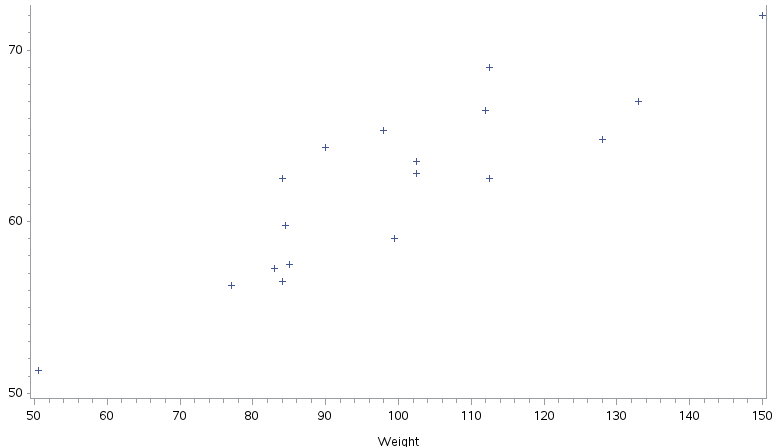
**GRAPH PLOTS**

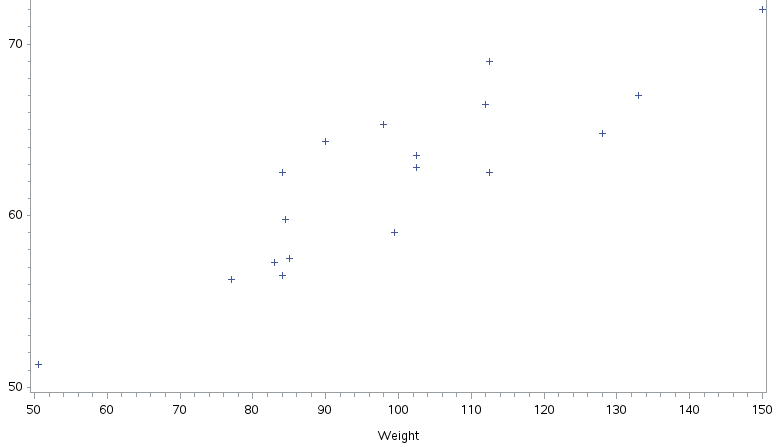
**PROC GPLOT DATA=SASHELP.CLASS;**

**PLOT HEIGHT \* WEIGHT;**

**RUN;**

**OP: (Default - +)**

****

****

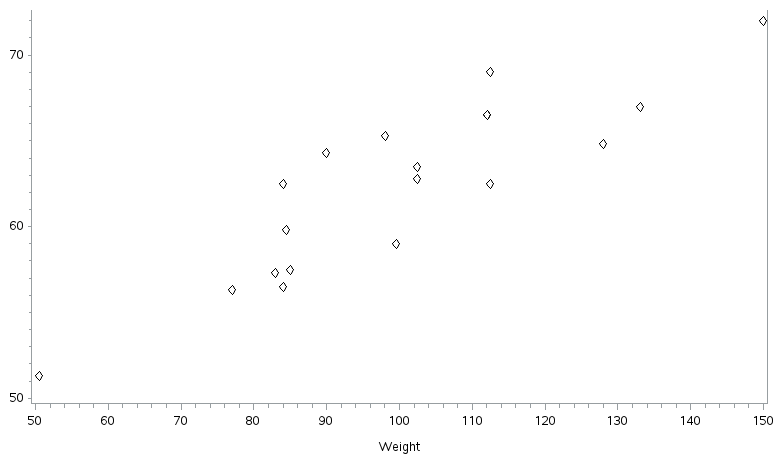
**$: [Diamond ♢]**

**PROC GPLOT DATA=SASHELP.CLASS;**

**PLOT HEIGHT \* WEIGHT = '$';**

**RUN;**

**OP:**

****

**#: [HEART ♡]**

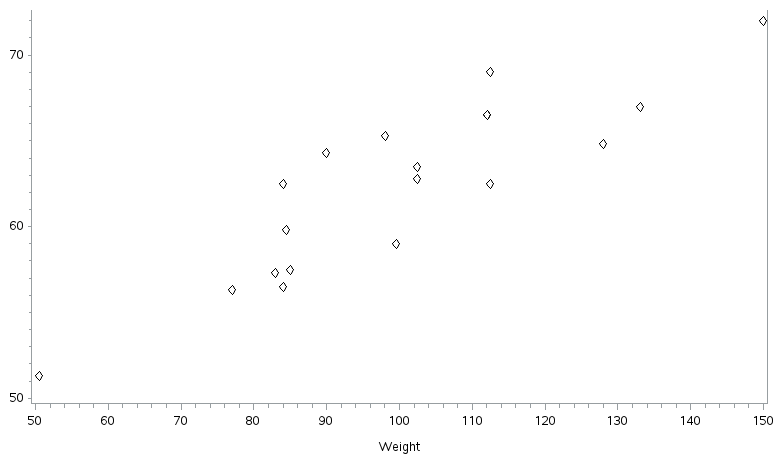
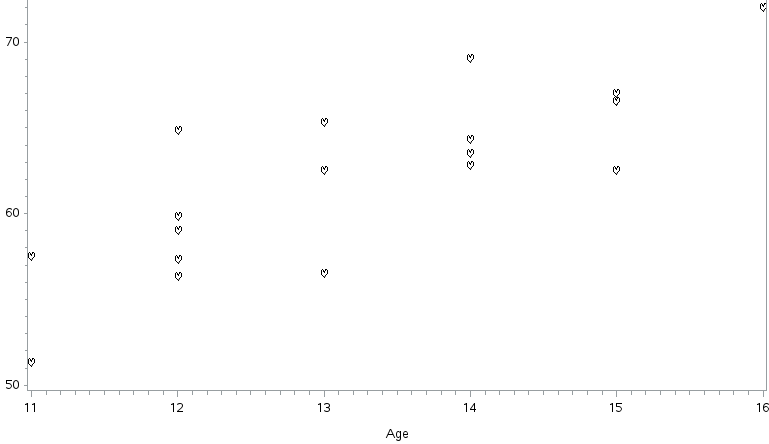
**PROC GPLOT DATA=SASHELP.CLASS;**

**PLOT HEIGHT \* WEIGHT = '$'**

**HEIGHT \* AGE = '#';**

**RUN;**

**OP:**

****

**& - CLOVER ☘**

**Square - ▢**

**OVERLAY:**

**Both Symbols will be shown in the same Axis.**

**Places all the plots that are generated by the PLOT statement on one set of axes.**

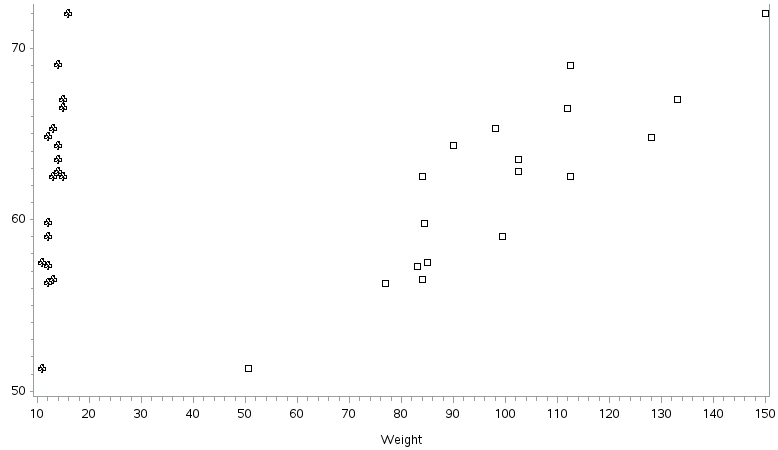
**PROC GPLOT DATA=SASHELP.CLASS;**

**PLOT HEIGHT \* WEIGHT = 'square'**

**HEIGHT \* AGE = '&'/OVERLAY;**

**RUN;**

**OP:**

****

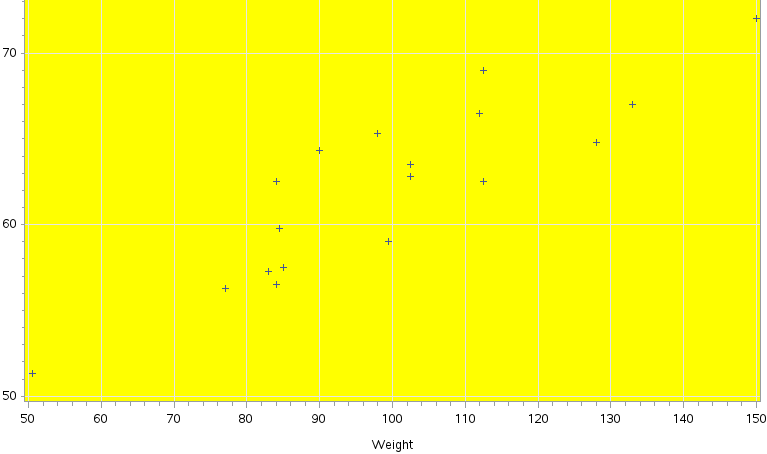
**TO CHANGE GRAPH BACKGROUND COLOUR:**

**PROC GPLOT DATA=SASHELP.CLASS;**

**PLOT HEIGHT \* WEIGHT/CFRAME=YELLOW AUTOHREF AUTOVREF;**

**RUN;**

**OP:**

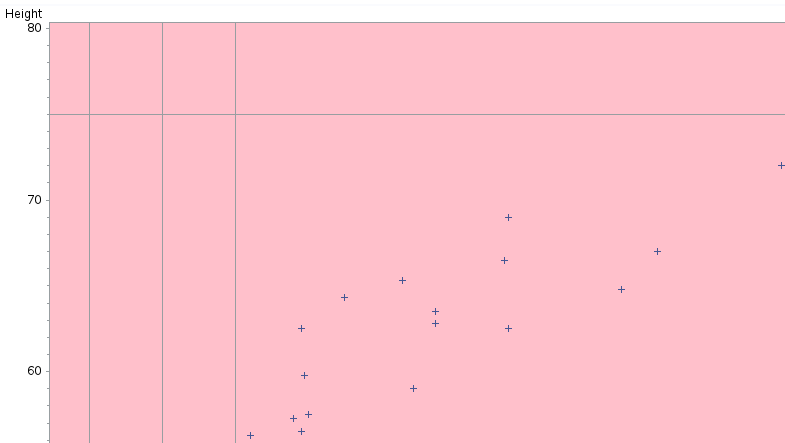
****

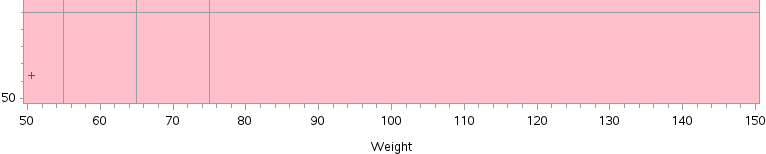
**PROC GPLOT DATA=SASHELP.CLASS;**

**PLOT HEIGHT \* WEIGHT/CFRAME=PINK HREF=55 65 75 VREF = 55 75;**

**RUN;**

**OP:**





**GRAPH WITH LINE:**

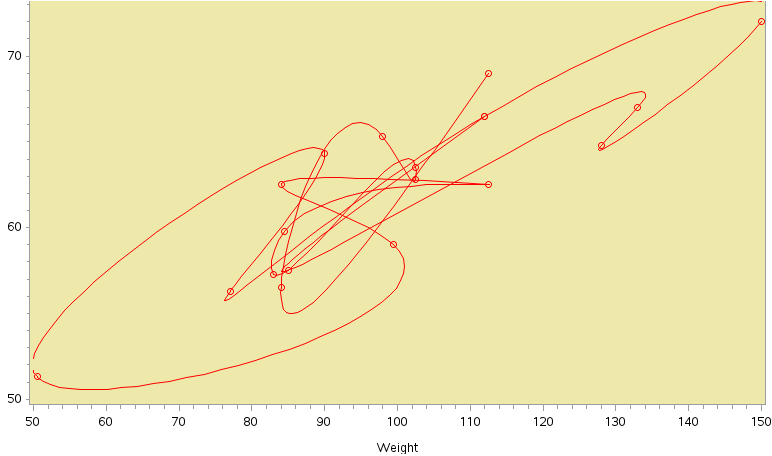
**PROC GPLOT DATA=SASHELP.CLASS;**

**PLOT HEIGHT \* WEIGHT/CFRAME=PINK CAUTOHREF=VIOLET;**

**SYMBOL C=RED I=SPLINE V=CIRCLE;**

**RUN;**

**OP:**

****

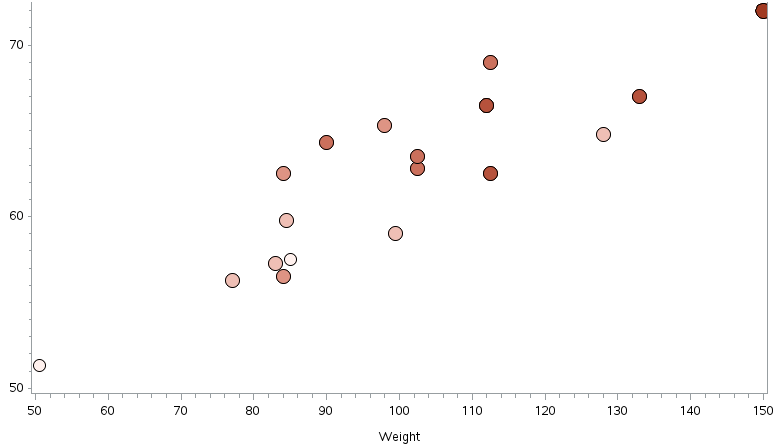
**BUBBLE CHART:**

**PROC GPLOT DATA=SASHELP.CLASS;**

**BUBBLE HEIGHT\*WEIGHT=AGE/BFILL=GRADIENT;**

**RUN;**

**OP:**



**GRAPHS:**

**PROC GCHART DATA=SASHELP.CLASS;**

**VBAR HEIGHT/SUBGROUP=WEIGHT;**

**HBAR HEIGHT/SUBGROUP=WEIGHT;**

**PIE HEIGHT/SUBGROUP=WEIGHT;**

**DONUT HEIGHT/SUBGROUP=WEIGHT;**

**BLOCK HEIGHT/SUBGROUP=WEIGHT;**

**RUN;**

**(Type ?)**

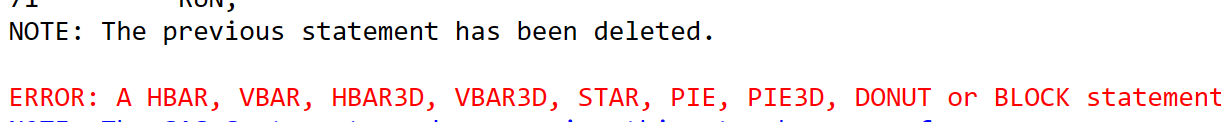
**PROC GCHART DATA=SASHELP.CLASS;**

**BLOCK HEIGHT/SUBGROUP=?;**

**RUN;**

**OP:**

**? gives you all the Statistical Options. You can choose whatever you’re in need.**

****

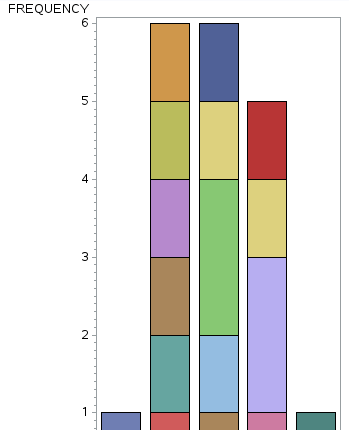
**VERTICAL BAR CHART:**

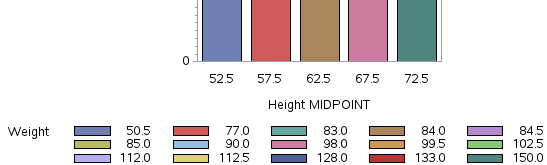
**PROC GCHART DATA=SASHELP.CLASS;**

**VBAR HEIGHT/SUBGROUP=WEIGHT;**

**RUN;**

**OP:**

****

****

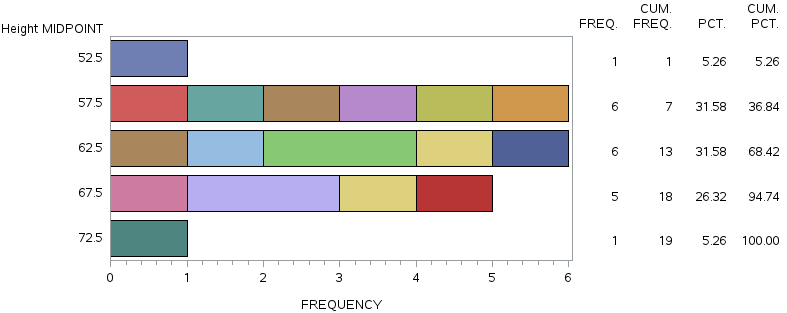
**HORIZONTAL BAR CHART:**

**PROC GCHART DATA=SASHELP.CLASS;**

**HBAR HEIGHT/SUBGROUP=WEIGHT;**

**RUN;**

**OP:**

****

**C:\Users\DELL\Pictures\Screenshots\Screenshot (212).png**

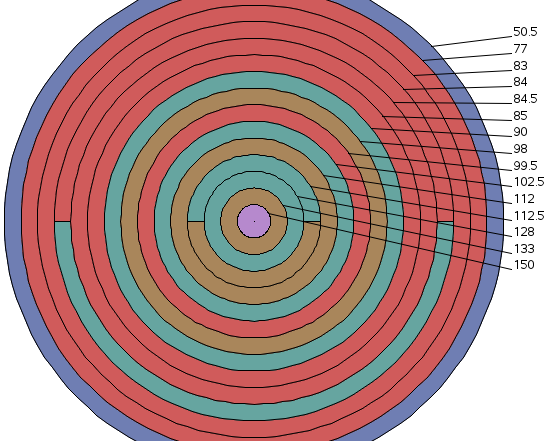
**PIE CHART:**

**PROC GCHART DATA=SASHELP.CLASS;**

**PIE HEIGHT/SUBGROUP=WEIGHT;**

**RUN;**

**OP:**

****

**C:\Users\DELL\Pictures\Screenshots\Screenshot (214).png**

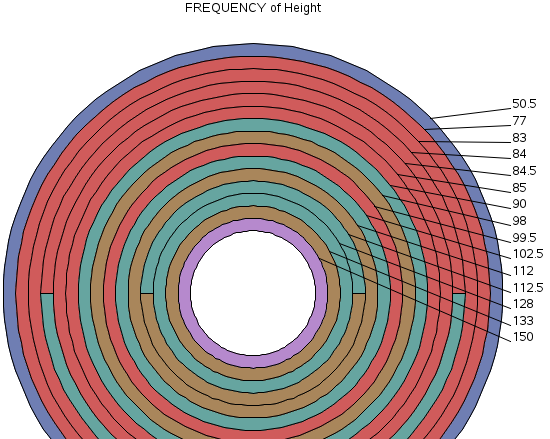
**DONUT CHART:**

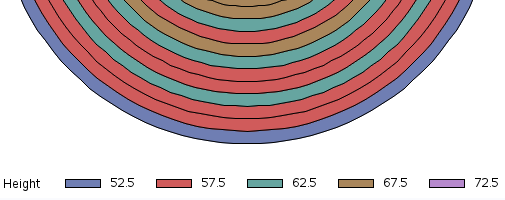
**PROC GCHART DATA=SASHELP.CLASS;**

**DONUT HEIGHT/SUBGROUP=WEIGHT;**

**RUN;**

**OP:**

****

****

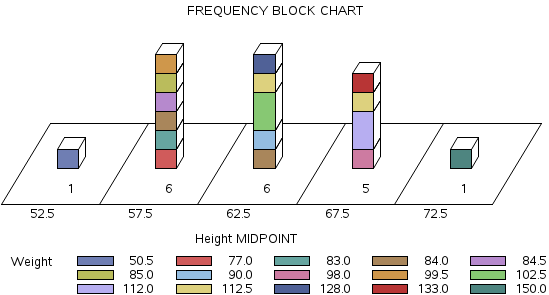
**BLOCK CHART:**

**PROC GCHART DATA=SASHELP.CLASS;**

**BLOCK HEIGHT/SUBGROUP=WEIGHT;**

**RUN;**

**OP:**

****

**3D GRAPHS [ VBAR3D | HBAR3D | PIE3D ]**

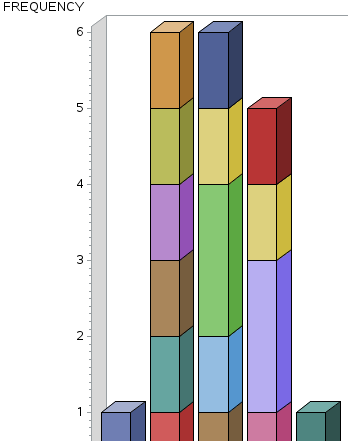
**VERTICAL 3D BAR:**

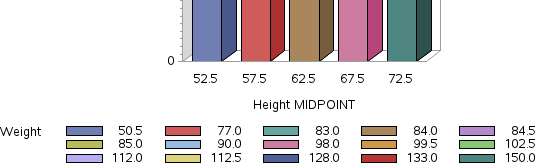
**PROC GCHART DATA=SASHELP.CLASS;**

**VBAR3D HEIGHT/SUBGROUP = WEIGHT;**

**RUN;**

**OP:**

****

****

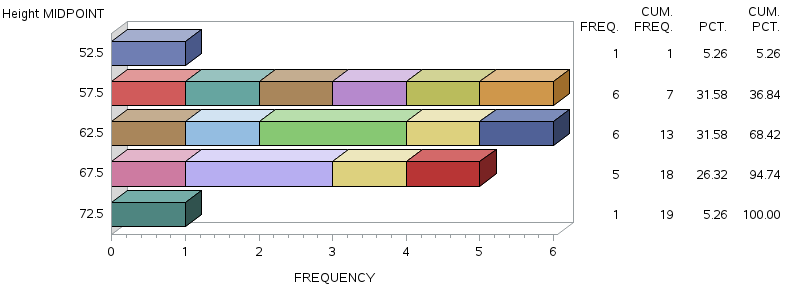
**HORIZONTAL 3D BAR:**

**PROC GCHART DATA=SASHELP.CLASS;**

**HBAR3D HEIGHT/SUBGROUP = WEIGHT;**

**RUN;**

**OP:**

** C:\Users\DELL\Pictures\Screenshots\Screenshot (226).png**

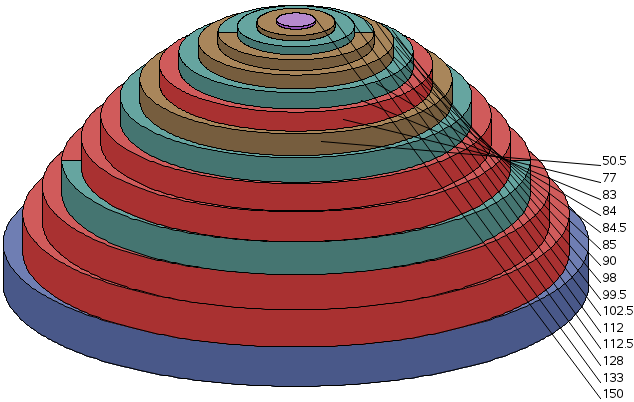
**PIECHART 3D BAR:**

**PROC GCHART DATA=SASHELP.CLASS;**

**PIE3D HEIGHT/SUBGROUP = WEIGHT;**

**RUN;**

**OP:**

****

**C:\Users\DELL\Pictures\Screenshots\Screenshot (230).png**

**PROC REPORT**

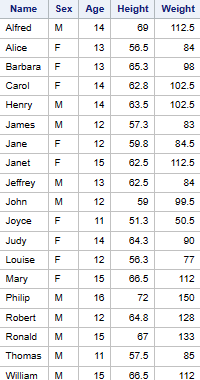
**BASIC REPORT:**

**(Report will be shown in Report Window, not in OP Window)**

**proc report data=sashelp.class;**

**run;**

**OP: [ Basic Report shows the Dataset ]**

****

**(OP will not be shown in report window. It’ll be shown in OP Window)**

**proc report data=sashelp.class nowd;**

**run;**

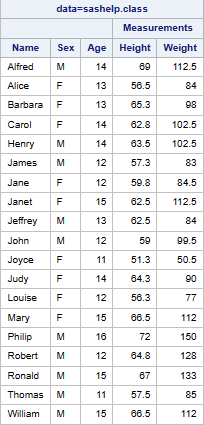
**REPORT WITH MODIFYING HEADINGS:**

**proc report data=sashelp.class;**

**column ('data=sashelp.class' name sex age('Measurements'height weight));**

**run;**

**OP:**

****

**REPORT WITH MODIFIED HEADING AND ORDERED(GROUP):**

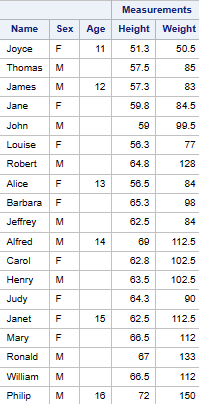
**proc report data=sashelp.class;**

**column name sex age('Measurements'height weight);**

**define age/order;**

**run;**

**OP:**

****

**proc report data=sashelp.class;**

**column age sex n;**

**define age/group;**

**define sex/group;**

**define N/'N' format=2.;**

**run;**

**OP:**

****

**proc report data=sashelp.class;**

**column age sex n height weight;**

**define age/group;**

**define sex/group;**

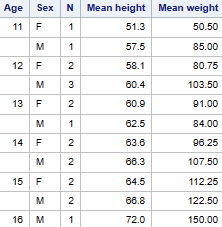
**define N/'N' format=2.;**

**define height/analysis mean 'Mean height'format=6.1;**

**define weight/analysis mean 'Mean weight' format=6.2;**

**run;**

**OP:**

****

**proc report data=sashelp.class;**

**column age N sex,weight,Mean;**

**define age/group;**

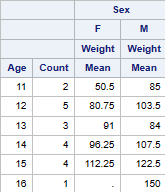
**define N/'Count';**

**define sex/across; [across – Transpose(Rotate)]**

**define weight/analysis;**

**run;**

**OP:**

****

**TO CREATE NEW VARIABLE:**

**proc report data=sashelp.class;**

**column name sex age height weight**

**mheight mweight; (To create new Variable)**

**define height/display; (DISPLAY must be there for new Variable)**

**define weight/display;**

**define mheight/computed format=6.2 'Height(cm)';**

**define mweight/computed format=5.2 'Weight(kg)';**

**compute mheight; (Compute Block is must for define statement)**

**mheight = height \*2.54;**

**endcomp;**

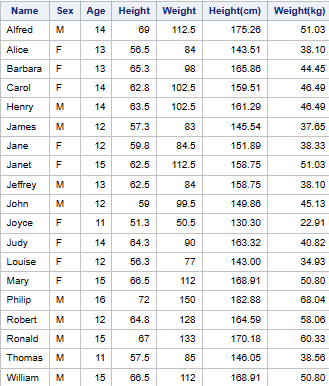
**compute mweight;**

**mweight = weight \*0.4536;**

**endcomp;**

**run;**

**OP:**

****

**MACROS**

**Macro is a facility inside SAS used for 3 purposes:**

1. **Reuse a particular code.**
2. **To increase code Complexity.**
3. **To decrease Coding Error.**

**Macro is split into 2 types:**

1. **Macro Variable - &**

* **System Defined Macro Variable**
* **User-defined Macro Variable**

1. **Macro Program - %**

**(% => Macro Program)**

**% PUT \_USER\_; (To Create Personalized)**

**% PUT \_GLOBAL\_; (To Create Common)**

**% PUT \_AUTOMATIC\_;**

**% PUT \_ALL\_;**

**MACROS FOR TITLE:**

**PROC PRINT DATA=SASHELP.CLASS;**

**TITLE 'THIS IS PRINTED ON' &SYSDATE 'ON'&SYSHOSTNAME;**

**RUN;**

**OP:**

****

**MACROS FOR DO LOOP:**

**%LET A=10;**

**%PUT &A;**

**DATA WORK.L1;**

**B=&A;**

**DO X=1 TO &A;**

**B = B+&A;**

**OUTPUT;**

**END;**

**RUN;**

**PROC PRINT DATA=WORK.L1;**

**OP:**

****

**MACROS FOR DO LOOP AND CREATING TITLE:**

**%LET A=10;%PUT &A;**

**%LET B=20;%PUT &B;**

**%LET C=30;%PUT &C;**

**DATA WORK.L1;**

**SP=&A;**

**DO X=1 TO &B;**

**SP = SP+&C;**

**OUTPUT;**

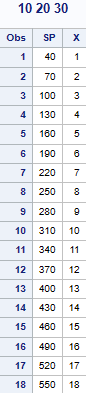
**END;**

**RUN;**

**PROC PRINT DATA=WORK.L1;**

**TITLE &A &B &C; RUN;**

**OP:**

****

**C:\Users\DELL\Pictures\Screenshots\Screenshot (244).png**

**MACRO WITH SET AND WHERE STATEMENT:**

**%LET S='F';**

**%PUT &S;**

**DATA WORK.P1;**

**SET SASHELP.CLASS;**

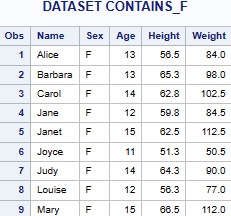
**WHERE SEX=&S;**

**RUN;**

**PROC PRINT DATA= WORK.P1;**

**TITLE 'DATASET CONTAINS\_' &S; RUN;**

**OP:**

****

**%MACRO PR(ABC);**

**PROC PRINT DATA=&ABC;**

**RUN;**

**%MEND;**

**%PR(SASHELP.CLASS);**

**%PR(WORK.P1);**

**%MACRO DR(A,B,C,D,E);**

**PROC PRINT DATA=&A &B;**

**VAR &C;**

**ID &D;**

**SUM &E;**

**RUN;**

**%MEND;**

**%DR(sASHELP.CLASS,N BLANKLINE=2,NAME SEX AGE HEIGHT,AGE,HEIGHT);**