|  |  |  |  |
| --- | --- | --- | --- |
| **Steps for Creating a SAS Datasets** | | | **Scope** |
| 1 | **LIBNAME** libref ‘<Path>’; | Reference a SAS data library | Global |
| 2 | **FILENAME** fileref ‘<Path>’; | Reference (Temp) an external file | Global |
| 3 | **DATA** ‘SASDataSetName’; | Name a SAS data set |  |
| 4 | **INFILE** ‘filename/fileref’  ***OBS***=10;  **FIRSTOBS=2;**  **Dlm=’,’ DSD;** | Identify an external file using INFILE statement  **OBS** mention the ***range till which data needs to be read.*** Can be used in data and proc print. Used to ***verify Data*** reading without affecting RAM space much. ***FIRSTOBS*** will start to read data from row2 of raw dataset, ***DLM/DSD*** is Delimiter and Delimiter sensitive data. |  |
| 5 | **INPUT** <informats>; | Describe data |  |
| 6 | Sum\_var **+** var2; | + is called accumulator variable.  ***Defaults to zero initially*** and in case if values are ***missing***. Values get summed as dataset is read. ***+ will automatically retain*** its value |  |
| 7 | **Retain** <Sum\_var> <val>; | ***Used to initializes Accumulator variable*** which is otherwise 0 by default. |  |
| 8 | **IF** <condition> then Vari=Val; | Condition can use any conditional operator:  ***=/eq,~=/^=/ne, >=/ge, <=/le, >/gt, </lt, in, &, |***  Character ***values need to be of same case*** in condition statements, enclosed in ‘‘  Condition inside ***parenthesis is given high importance***. ***BODMAS*** rule apply here.  **0/. = False**, that is 0 or missing is false  **1 = True** |  |
| 9 | **LENGTH** Var1 $ 10 Var2 20; | By default, SAS ***allocates the space of first value*** it encounters. Numeric variables have ***default size 8***.  This ***should be declared before value is set*** |  |
| 10 | If <condition> then <stmt>;  **Else** if <condi2> then <stmt>;  Else <final condition>; | Used for code optimization  Better to arrange else-if operation in ***decreasing probability*** to increase performance. |  |
| 11 | If <condition> then **DELETE**; | This is used to ***delete an observation using condition***. Used mostly ***along with IF*** |  |
| 12 | **DROP** **=** Var1 / **KEEP** **=** Var1; | This ***can be used in Data Step as well SAS procedures***.  Doesn’t apply to all output dataset that are named in Data statement. |  |
| 13 | **DROP** Var1 / **Keep** Var1; | ***Cannot be used in proc steps***  ***Applies to all o/p data sets***  Based on the cou of variables use Drop and Keep wisely. |  |
| 14 | **LABEL** Var=’Label Detail’;  **FORMAT** Var1 DOLLAR12; | Used to provide a ***permanent label/format*** to a variable. However, when ***used in Proc*** statement ***can override this behaviour***. |  |
| 15 | **SELECT** <Var>; **WHEN** (“Val”) stmt; **otherwise** <stmt>; **end**; | This is like a Switch-Case statement, this will use ***select – when – otherwise - end*** |  |
| 16 | **DO**; <SAS Statements>; **END**; | If loop or when can handle only one stmt, do can handle many statement in its block |  |

**Column Style:** **[Standard Data + *Well Ordered in Column]***

1--------10---------20---------30--------40---------50---------60---------70---------80--------90

124 61 Mod Male Pradeep United States

123 76 Ded Female Sruthi India

142 89 Reg Male Sathyamurthy United Kingdom

|  |  |  |
| --- | --- | --- |
| **PROC PRINT DATA=DATASETNAME**  **NOOBS \*used to avoid printing observation column while printing;**  **DOUBLE \*print double spacing in SAS Output and not in SAS Report;**  **(OBS=3) \* Print only the first 3 observation of the dataset in print;** | | **Scope** |
| **Sum** <Col Name>; | Calculate the sum of the column | Local |
| **VAR** <Col Name>; | Mention the variable and its ***order*** of printing | Local |
| **Label** <Col Name>=’’; | Define label name for a column  Can mention up to ***256 char***  ***Can be defined in single or multiple lines*** | **Local** |
| **Where** <column condi>  **CONTAINS** ‘str’;  **?** ‘str’;  **IN**(‘str1’,’str2’); | Defines the column condition  ***=, ^=, >, <, >=, <=***  ***CONTAINS*** *is string comparison*  ***AND, OR*** *operator used along with col name each time*  ***IN*** *operator is used as SQL style in comparison.* | Local |
| **ID** <Col Names>; | Act as a primary key, ***replace OBS*** column without explicitly mention of NOOBS.  ID used along with ***Var*** will ***display*** a ***column twice***. | Local |
| **SUM** <Col Name>; | Will provide the total of the column specified. | Local |
| **BY** <Col Name>; | Col Name should be ***same as one that is sorted before*** using this. ***Subset results***. | Local |
| **BY** <Col Name1>;  **ID** <Col Name1>; | When ID used along with BY it will:   1. ***Supress OBS*** column 2. ID/BY variable name is printed in left col 3. Each ***ID/BY value is printed only once*** at the start of each by group and on the line, that has group sub-total. | Local |
| **By** <Col Name1>;  **PAGEBY**<Col Name1>; | Mostly used along with sum-by-id.  ***Column used in PAGEBY should be same as one used in BY***.  Used to ***print each sub-total on a separate page***. | Local |
| **FORMAT** <Col Name>; | When defined ***inside PROC it scopes within it***. To make it ***permanent FORMAT or Labels*** need ***to be defined in DATA step*** | Local/Global |
| **TITLE** ‘str1’; | Generally, ***need to be defined outside a PROC step***.  However, it can be used inside PROC too  ***TITLE is global.*** Once defined will stay forever until title statement is modified, cancelled or end SAS session.  ***Cancel of title*** is done by ***title;*** | Global |
| **FOOTNOTE** ‘str2’; | Used to print note below a table/graph  It is same as TITLE function, up to ***10 footnotes can be defined*** in SAS.  ***Cancel of footnote*** is done by:  ***Footnote;*** | Global |

|  |  |  |
| --- | --- | --- |
| **PROC SORT** **DATA=DATASETNAME**  **OUT=DATASETNAME \*o/p SAS dataset** | |  |
| **by** <Col Name>;  **by descending** <col1> | Sorted by the column mentioned, sort takes place from right to left columns mentioned.  If used with descending it will apply to ***column which is immediately after it***, rest of the other columns will be sorted in ascending order. | Local |
| **NOTSORTED**; | To explicitly mention not to sort if the ***values are equal*** based on by condition. | Local |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **PROC FORMAT LIB=library**  **LIBRARY/LIB \*Defines the SAS library that needs to be referred;**  **FMTLIB \*print all the user defined format present in the Library mentioned;** | | | **Scope** |
| 1 | **LIBNAME** ***library*** ‘<Path>’; | Reference a SAS data library | Permanent |
| 2 | **PROC FORMAT LIB**=library  **FMTLIB**; | Library can be the SAS library referred above or it can be a catalog like ***library.catalog.***  ***FMTLIB*** will list all the user defined format present in the library. ***formats.sas7bcat*** file is created in the path mentioned in library. | Permanent |
| 3 | **Value** <format-name> | Format name must begin with ***$ for Char*** var  Cannot be > 8 char in length  Cannot be the name of existing SAS format  Cannot end with a number  Does not end with a period when defined | Permanent |
|  | Range1=’label1’ | Range1= Actual Column Data  Label1= Description of Range1  Numeric => 102=’Manager’  Character => ‘A’=’Good Performance’  Range => low-<12=’Not Teen Age’ | Permanent |
|  | Range2=’label2’; | Always the last Range must be ended with; which implies SAS that PROC FORMAT statement ends. | Permanent |
| 4 | **PROC FORMAT;** | This format will be created in the ***work directory*** which ***means temporary***. | Temporary |
|  | Value <format-name> | Scope within that SAS session only | Temporary |
|  | Range1=’label1’ | Scope within that SAS session only | Temporary |
|  | Range2=’label2’; | Scope within that SAS session only | Temporary |
| 5 | **PROC CATALOG;** | You can delete the user defined format | Permanent |

|  |  |  |  |
| --- | --- | --- | --- |
| **PROC REPORT DATA=<DATASETNAME>**  **WD/NOWD \*Decides should the o/p be printed in a dedicated report window;**  **DOUBLE \*print double spacing in SAS Output and not in SAS Report;**  **SPLIT=’<symbol>’ \* Symbol can be \*, # $ etc., Used to define the label split in reporting;** | | | **Scope** |
| 1 | **COLUMN** <Col Names> | Used ***to subset the column*** that is needed to be displayed in the report. | Local |
| 2 | **WHERE** <Col Condi/Name>  **In** (‘value1’,’value2’) | Used to ***filter out the data*** required  In ***used along with where to filter the data*** based on values provided, SQL style usage. | Local |
| 3 | **DEFINE** <Col1>/<***usage***>  DEFINE <Col2>/<***attribute***>  DEFINE <Col3>/<***options***>  DEFINE <Col4>/<**Justify**>  DEFINE <Col5>/<***Col*** Heading>    \* Column definition;  **PROC** **REPORT** DATA=CARS\_SAMPLE NOWD SPLIT='\*' HEADLINE HEADSKIP;  define Make/format=$CHAR8. width=**3** spacing=**10**;  define Type/'Car\*Type';  define Model/center;  define Cylinders/order DESCENDING;  define Cylinders/group;  **RUN**;  \* Column definition - usage of group definition;  **PROC** **REPORT** DATA=CARS\_SAMPLE NOWD SPLIT='\*' HEADLINE HEADSKIP;  column cylinders MSRP;  define cylinders/group;  **RUN**;  \* Specifying statistics;  **PROC** **REPORT** DATA=CARS\_SAMPLE NOWD SPLIT='\*' HEADLINE HEADSKIP;  column cylinders MSRP;  define cylinders/group;  define MSRP/mean 'Average of MSRP';  **RUN**;  \* Column definition - usage of across definition;  **PROC** **REPORT** DATA=CARS\_SAMPLE NOWD SPLIT='\*' HEADLINE HEADSKIP;  column cylinders type MSRP;  define cylinders/across;  define type/across;  **RUN**; | Used to ***build column definitions*** in report like column space and width, etc.,  Let to ***define more than one column attribute*** at a time.  Column can be defined ***in any order*** and list ***options within it in any order as well***.  ***Usage*** specifies ***how to use the variables***:  By ***default***, **Char** Variable defined as ***Display***  And ***Numeric*** variables defined as ***Analysis***   1. **Across** – Displays variable ***horizontally*** rather vertically 2. **Analysis** - Default ***SUM*** analysis. 3. **Computed** – ***position*** of compute variable is ***very important. Use compute*** *and* ***endcomp*** *and* ***derive the value*** with some formula 4. ***Display*** – This is for ***Char*** variables 5. ***Group*** – to create ***summary report***. To get a proper result, display/character variables need to be grouped properly. 6. **Order** – This is like Grouping and Order, by ***default it is ordered in ascending***, if needed we need explicit mention of value ***DESCENDING***.   ***Attributes*** specifies the ***look*** of each column:  Width and spacing has its ***effect only in o/p window*** and doesn’t affect HTML window.   1. **Format** – define SAS/user format, default is ***its variable type*** 2. **Width** – width of col, default is ***Max*** 3. **Spacing** – No of blank char, default is ***2***   ***Options*** specifies the ***further formatting*** option:   1. **DESCENDING** 2. **NOPRINT** 3. **NOZERO** 4. **PAGE**   ***Justification*** specifies ***arrangements*** of column:   1. **Center** – Justify the char in centre 2. **Left** – ***default for*** ***chars*** n left justify 3. **Right** – ***default for*** ***num*** n right justify | Local |
| ***Column Heading*** is the ***label definition***. ***Split*** in report definition is used to ***split the column label*** as needed. (e.g. ***SPLIT=’\*’;***) define col/c\*t; | | | |

|  |  |  |
| --- | --- | --- |
| SI.NO | Statistics | Definition |
| 1 | **CSS** | Corrected sum of squares |
| 2 | **USS** | Uncorrected sum of squares |
| 3 | **CV** | Coefficient of variation |
| 4 | **MAX** | Maximum value |
| 5 | **MEAN** | Average |
| 6 | **MIN** | Minimum Value |
| 7 | **N** | Number of observations with non-missing values |
| 8 | **NMISS** | Number of observations with missing values |
| 9 | **RANGE** | Range |
| 10 | **STD** | Standard deviation |
| 11 | **STDERR** | Standard error of the mean |
| 12 | **SUM** | Sum |
| 13 | **SUMWGT** | Sum of the Weight variable values |
| 14 | **PCTN** | Percentage of a cell or row frequency to a total frequency |
| 15 | **PCTSUM** | Percentage of a cell or row sum to a total sum |
| 16 | **VAR** | Variance |
| 17 | **T** | Student's *t* for testing the hypothesis that the population mean is 0 |
| 18 | **PRT** | Probability of a greater absolute value of student's *t* |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Computing Statistics for Numeric Variable*** | | | |
| **PROC MEAN DATA=<DATASETNAME>**  **\*By default gives descriptive statistics, with n-count of all non-missing values;**  **<STATS KEYWORDS> \*To suppress default o/p and choose what stats is required for o/p;**  **MAXDEC=2 \*To set the decimal point;**  **NOPRINT \*Supress the result being printed;** | | | **Scope** |
| 1 | **VAR** <Col Names>; | Used to display the ***variables for which the statistics are required*** | Local |
| 2 | **CLASS** <Col Names>; | Specifies categorical variables which needed ***group processing*** | Local |
| 3 | **OUTPUT**  <***STATS***>=<Col Names>  ***OUT*** = <O/p dataset> | ***Output*** is used to ***structure the final output of the PORC MEAN*** above the segregation done based on a class variable.  <***STATS***> can be any ***statistic key-word*** and col name specifies on which columns it needs to be applied.  If <STATS> keywords are ***not mentioned***, then ***SAS will produce*** whole statistics and add ***\_STAT\_ variable along with \_TYPE\_ and \_FREQ\_***  ***\_TYPE\_*** is a ***simple binary pattern*** ***to summarise the CLASS variable***.    ***\_FREQ\_*** is the count of class variable occurrence  ***OUT*** specifies the ***output dataset*** in which the final ***statistic result needs to be stored***. | Local |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Computing Statistics for Numeric Variable*** | | | |
| **PROC SUMMARY DATA=<DATASETNAME> PRINT;** | | | **Scope** |
| 1 | **VAR** <Col Names>; | Used to display the ***variables for which the statistics are required*** |  |
| 2 | **CLASS** <Col Names>; | Specifies categorical variables which needed ***group processing*** |  |
| 3 | **OUTPUT**  <***STATS***>=<Col Names>  ***OUT*** = <O/p dataset> | ***Output*** is used to ***structure the final output of the PORC MEAN*** above the segregation done based on a class variable. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Descriptive Statistics | | | |
| SI.NO | **Keywords** | **Definition** |
| 1 | **CLM** | Two-sided confidence limit for the mean |
| 2 | **CSS** | Corrected sum of squares |
| 3 | **CV** | Coefficient of variation |
| 4 | **KURTOSIS / KURT** | Kurtosis |
| 5 | **LCLM** | One-sided confidence limit below the mean |
| 6 | **MAX** | Maximum value |
| 7 | **MEAN** | Average |
| 8 | **MIN** | Minimum value |
| 9 | **N** | Number of observations with non-missing values |
| 10 | **NMISS** | Number of observations with missing values |
| 11 | **RANGE** | Range |
| 12 | **SKEWNESS / SKEW** | Skewness |
| 13 | **STDDEV / STD** | Standard deviation |
| 14 | **STDERR / STDMEAN** | Standard error of the mean |
| 15 | **SUM** | Sum |
| 16 | **SUMWGT** | Sum of the Weight variable values |
| 17 | **UCLM** | One-sided confidence limit above the mean |
| 18 | **USS** | Uncorrected sum of squares |
| 19 | **VAR** | Variance |

|  |  |  |  |
| --- | --- | --- | --- |
| Quantile Statistics | | | |
| SI.NO | **Keywords** | **Definition** |
| 1 | **MEDIAN / P50** | Median or 50th percentile |
| 2 | **P1** | 1st percentile |
| 3 | **P5** | 5th percentile |
| 4 | **P10** | 10th percentile |
| 5 | **Q1 / P25** | Lower quartile or 25th percentile |
| 6 | **Q3 / P75** | Upper quartile or 75th percentile |
| 7 | **P90** | 90th percentile |
| 8 | **P95** | 95th percentile |
| 9 | **P99** | 99th percentile |
| 10 | **QRANGE** | Difference between upper and lower quartiles: Q3-Q1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Hypothesis Testing | | | |
| SI.NO | **Keywords** | **Definition** |
| 1 | **PROBT** | Probability of a greater absolute value for the *t* value |
| 2 | **T** | Student's *t* for testing the hypothesis that the population mean is 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Computing Statistics for Categorical Variable*** | | | |
| **PROC FREQ DATA=<DATASETNAME>**  **WD/NOWD \*Decides should the o/p be printed in a dedicated report window;** | | | **Scope** |
| 1 | **TABLE** <Col Names>  / **NOCUM**; | Used to ***mention the column names based on which a frequency table*** needs to be constructed.  ***One column name*** in TABLE will construct a ***simple frequency table with frequency and cumulative frequency and percentage***, totally 4 outputs.  ***NOCUM*** will ***supress the display of cumulative frequency and percentage*** from the output. | Local |
| 2 | **TABLE** <COL1> - <COL5> | This will again create simple frequency table for columns-1 to column-5 | Local |
| 3 | PROC FORMAT;  Value <frmt\_name> range1 ‘label-1’  Range2 ‘label-2’  Range3 ‘label-3’  RUN;  **PORC FREQ** data=<datasetnames>;  **Tables** <cat\_col\_name>;  ***Format weight <frmt\_name>.***; |  |  |
| 4 | **TABLE** <COL1> \* <COL2>; | This will ***create two-way table***. This will ***cross tabulate*** 2 different categorical variables. | Local |
| 5 | **TABLE** <COL1> \* <COL2> \* <COL3>; | This will ***create N-way table***. This will ***cross tabulate*** N different categorical variables. | Local |
| 6 | **TABLE** <COL1> \* <COL2>  / ***CROSSLIST***; | ***CROSSLIST*** will ***display cross tabulation in a ODS format***. This ***ODS output can be customized using*** the ***TEMPLATE*** procedure. | Local |
| 7 | **TABLE** <COL1> \* <COL2>  / ***LIST***; | Produce list output for crosstabulation. Puts frequency table in a simple and short table. | Local |
|  | **TABLE** <COL1> \* <COL2>  / ***nofreq nopercent norow nocol***; | ***Nofreq*** will ***supress the cell frequency***  ***Nopercent*** will supress the ***cell percentage***  ***Norow*** will supress ***row percentages***  ***Nocol*** will supress ***column percentage*** | Local |