\* SAS : READ CHARACTER VARIABLE OF VARYING LENGTH ;

\* Method I : Use COLON Modifier;

/\* Use colon modifier : to tell SAS to read variable until there is a

space or other delimiter \*/

data mylib.example1;

input ID Name :$30. Score;

cards;

1 DeepanshuBhalla 22

2 AttaPat 21

3 XonxiangnamSamnuelnarayan 33

;

run;

proc print data = mylib.example1;

run;

\* Method II : Use Ampersand (&);

\* Use ampersand (&) to tell SAS to read the variable until there are two or more spaces

as a delimeter;

data mylib.example1;

input ID Name & $30. Score;

cards;

1 Deepanshu Bhalla 22

2 Atta Pat 21

3 Xonxiangnam Samnuel Narayan 33

;

Run;

proc print data = mylib.example1;

run;

\* Method III - Read delimited data;

DATA mylib.outdata;

INFILE Datalines dlm =",";

INPUT age gender $ dept obs1 obs2 obs3;

Datalines;

1,F,3,17,6,24

1,M,1,19,25,7

3,M,4,24,10,20

3,F,2,19,23,8

2,F,1,14,23,12

;

run;

proc print data = mylib.outdata;

run;

/\* Import the raw/external files either in Excel, CSV or Text Files \*/

\* CSV File;

PROC IMPORT OUT= mylib.Importcsv

DATAFILE= "/home/u1048896/Test\_2.csv"

DBMS=csv replace;

getnames=Yes;

RUN;

proc print data = mylib.Importcsv;

run;

\* Excel File ;

PROC IMPORT OUT= mylib.Importxls

DATAFILE= "/home/u1048896/Test.xlsx"

DBMS=xlsx REPLACE;

SHEET="Sheet1";

GETNAMES=No;

datarow = 5;

RUN;

proc print data = mylib.Importxls;

run;

\* Txt File with delimiter;

PROC IMPORT OUT= mylib.Importtxt

DATAFILE= "/home/u1048896/Test.txt"

DBMS=dlm REPLACE;

GETNAMES=yes;

delimiter = ',';

datarow = 4;

RUN;

Proc print data = mylib.Importtxt;

run;

\* Txt File with space delimiter;

PROC IMPORT OUT= mylib.Importtxt

DATAFILE= "/home/u1048896/Test\_S.txt"

DBMS=dlm REPLACE;

GETNAMES=yes;

delimiter = ' ';

datarow = 4;

RUN;

Proc print data = mylib.Importtxt;

run;

\* Txt File with multiple delimiter;

PROC IMPORT OUT= mylib.Importtxt

DATAFILE= "/home/u1048896/Test\_MD.txt"

DBMS=dlm REPLACE;

GETNAMES=yes;

delimiter = '#,@';

datarow = 4;

RUN;

Proc print data = mylib.Importtxt;

run;

\* Import data using Infile statement;

FILENAME sample "/home/u1048896/Test.txt" ;

DATA mylib.outdata;

infile sample dlm=',' firstobs =2 obs=3;

INPUT age gender $ dept obs1 obs2 obs3;

run;

Proc print data = mylib.outdata;

run;

\* Import data from external database;

/\*Proc Sql - DBMS Connection\*/

/\*DataserverName - Oracle, TeraData, SQLServer, DB2 etc.\*/

proc sql;

connect to DataserverName(User=username password=pwd path='myserverpath');

create table libraryname.outputdatasetname as select \* from connection to DataserverName(SQL Query);

Disconnect from DataserverName;

quit;

\* SPECIFY A LIST OF VARIABLES;

data mylib.dummy;

input q1 q3 q4 q2 q6$ bu$ q5;

cards;

1 2 3 5 sa an 3

2 4 3 6 sm sa 4

6 5 3 8 cb na 3

;

run;

Proc print data = mylib.dummy;

run;

\* A single dash (-) is used to specify consecutively numbered variables. For example : q1-q4;

/\* A double dash (--) is used to specify variables based on the order of the variables as

they appear in the file, regardless of the name of the variables. \*/

data mylib.dummy1;

set mylib.dummy;

sum = sum(of q1-q4);

sum1 = sum(of q1--q4);

drop q1--q5;

run;

Proc print data = mylib.dummy1;

run;

\* How to specify all NUMERIC variables;

data mylib.dummy1;

set mylib.dummy;

sum = sum(of \_numeric\_);

drop q1--q5;

run;

Proc print data = mylib.dummy1;

run;

\* More Dash Symbol Usage;

\* Print all NUMERIC variables from q1 through q6 ;

proc print data = mylib.dummy;

var q1-numeric-q6;

run;

\* Print all CHARACTER variables from q1 through q6.;

proc print data = mylib.dummy;;

var q1-character-q6;

run;

\* Print all CHARACTER variables.;

proc print data = mylib.dummy;;

var \_character\_;

run;

\* Print all numeric variables.;

proc print data = mylib.dummy;;

var \_numeric\_;

run;

\* WILDCARD CHARACTER;

\* Example 1 : Keep all the variables start with 'X';

DATA mylib.READIN;

INPUT ID X1 X\_T $;

CARDS;

2 3 01

3 4 010

4 5 022

5 6 021

6 7 032

;

RUN;

DATA mylib.READIN2;

SET mylib.READIN;

KEEP X:;

RUN;

Proc print data = mylib.READIN;

run;

Proc print data = mylib.READIN2;

run;

\* Example 2 : Subset data using wildcard character ;

DATA mylib.READIN2;

SET mylib.READIN;

IF X\_T =: '01';

RUN;

Proc print data = mylib.READIN;

run;

Proc print data = mylib.READIN2;

run;

/\* In this case, the COLON (:) tells SAS to select all the cases starting

with the character '01'. \*/

\* Example 3 : Use of WildCard in IN Operator;

DATA mylib.READIN2;

SET mylib.READIN;

IF X\_T IN: ('01', '02');

RUN;

Proc print data = mylib.READIN;

run;

Proc print data = mylib.READIN2;

run;

/\* In this case, the COLON (:) tells SAS to select all the cases starting

with the character '01' and '02'. \*/

\* Example 4 : Use of WildCard in GT LT (> <) Operators;

DATA mylib.READIN2;

SET mylib.READIN;

IF X\_T >: '01';

RUN;

Proc print data = mylib.READIN;

run;

Proc print data = mylib.READIN2;

run;

/\* In this case, the COLON (:) tells SAS to select all the cases from

character '01' up alphabetically. \*/

\* CHARACTER FUNCTIONS;

\* COMPBL Function - It compresses multiple blanks to a single blank.;

Data mylib.char;

Input Name $1-50 ;

Cards;

Sandy David

Annie Watson

Hello ladies and gentlemen

Hi, I am good

;

Run;

Data mylib.char1;

Set mylib.char;

char1 = compbl(Name);

run;

\* STRIP Function - It removes leading and trailing spaces.;

Data mylib.char1;

Set mylib.char;

char1 = strip(Name);

run;

/\* COMPRESS Function -

SYNTAX - COMPRESS(String, characters to be removed, Modifier)

Default - It removes leading, between and trailing spaces. \*/

Data mylib.char1;

Set mylib.char;

char1 = compress(Name);

run;

Proc print data = mylib.char1;

run;

\* Remove specific characters;

Data mylib.char1;

Set mylib.char;

char1 = compress(Name, 'n');

run;

Proc print data = mylib.char1;

run;

/\* Modifier

a – Remove all upper and lower case characters from String.

ak - Keep only alphabets from String.

kd - Keeps only numeric values

d – Remove numerical values from String.

i – Remove specified characters both upper and lower case from String.

k – keeps the specified characters in the string instead of removing them.

l – Remove lowercase characters from String.

p – Remove Punctuation characters from String.

s – Remove spaces from String. This is default.

u – Remove uppercase characters from String.

\*/

Data mylib.char1;

Set mylib.char;

char1 = compress(Name, '', 'u');

run;

Proc print data = mylib.char1;

run;

\* LEFT Function - It moves leading blanks to the end of the value. ;

Data mylib.char1;

Set mylib.char;

char1 = left(Name);

run;

\* TRIM Function - It removes trailing spaces.;

Data mylib.char1;

Set mylib.char;

char1 = trim(Name);

run;

\* CAT Function - It concatenates character strings with delimiter;

\* Can use '||' operator to concatenate;

Data mylib.char;

input fname $ lname $;

datalines;

rajesh kumar

hari singh

;

run;

Data mylib.char1;

Set mylib.char;

char1 = cat(fname, lname);

char2 = cats(fname, lname);

char3 = catx('@', fname, lname);

run;

Proc Print data = mylib.char1;

run;

\* CATS strips both leading and trailing blanks, and does not insert separators.;

\* CATX strips both leading and trailing blanks, and inserts separators. ;

/\* SCAN Function - It extracts words within a value that is marked by delimiters.

SCAN(text, nth word, <delimiters>) \*/

data mylib.char;

input name $40.;

datalines;

Amit@Kumar@Singh@Bangalore

Ajit@Kumar@Verma@Delhi

Prem@Shankar@Gupta@Kashi

Rajesh@Kumar@Sharma@Jhasi

;

run;

data mylib.char1;

set mylib.char;

fname=Scan(name,1,'@');

mname=Scan(name,2,'@');

lname=scan(name,3,'@');

city=scan(name,4,'@');

run;

proc print data=mylib.char1;

run;

data mylib.char1;

set mylib.char;

fname=Scan(name,-4,'@');

mname=Scan(name,-3,'@');

lname=scan(name,-2,'@');

city=scan(name,-1,'@');

run;

proc print data=mylib.char1;

run;

\* SUBSTR: Selects a substring from a character expression;

\* Syntax = substr(String, StartingPosition, TotalChar);

data mylib.char;

input phone $15.;

datalines;

+91-1234567890

+91-0987654321

;

run;

data mylib.char1;

set mylib.char;

str=Substr(phone,5,10);

run;

proc print data=mylib.char1;

run;

/\* LOWCASE, UPCASE and PROPCASE

LOWCASE converts the character string to lowercase.

UPCASE converts the character string to uppercase.

PROPCASE returns the word having uppercase in the first letter and lowercase in the rest of

the letter (sentence format).

\*/

Data mylib.char;

input name $;

datalines;

raJesh

haRi

;

run;

Data mylib.char1;

Set mylib.char;

char1 = lowcase(name);

char2 = upcase(name);

char3 = propcase(name);

run;

proc print data=mylib.char1;

run;

\* INDEX Function - It finds characters or words in a character variable;

Data mylib.char;

input name $;

datalines;

rajesh

hari

;

run;

Data mylib.char1;

Set mylib.char;

char1 = index(name, 'j');

run;

proc print data=mylib.char1;

run;

\* FIND Function - To locate a substring within a string

FIND(character-value, find-string, start);

Data mylib.char;

input name $20.;

datalines;

rajeshkumar

harisingh

;

run;

Data mylib.char1;

Set mylib.char;

char1 = find(name, 'a', 6);

run;

proc print data=mylib.char1;

run;

\* TRANWRD Function - It replaces all occurrences of a word in a character string.

It doesn't replace full phrase (entire value content).

TRANWRD(variable name, find what , replace with);

Data mylib.char;

input name $20.;

datalines;

rajeshkumar

harisingh

;

run;

Data mylib.char1;

Set mylib.char;

char1 = tranwrd(name, 'rajesh', 'suresh');

run;

proc print data=mylib.char1;

run;

\* PRXMATCH

It can be used for the following cases :

a) When you want to identify if there is alphanumeric (has any letter from A to Z)

in a variable.

b) If you need to search a character variable for multiple different substrings.

PRXMATCH (Regular-expression, source)

Regular Expression

^ - start with

$ - end with

\D - any non digits

\d - digits

? - may or may not have?

| - or

\* - repeating

( i:) - turns ON the case insensitive search

(-i:) - turn OFF the case insensitive search

;

DATA mylib.test;

INPUT string $;

CARDS;

ACBED

11

12

zx

11 2c

abc123

;

run;

data mylib.test1;

set mylib.test;

prxmatch=prxmatch("/[a-zA-Z]/",string);

run;

Proc print data = mylib.test1;

run;

\* prxmatch("/[a-zA-Z]/",string) checks first character.;

\*LENGTH: Returns the length of a non-blank character string, excluding

trailing blanks, and returns 1 for a blank character string ;

data mylib.char;

input string $20.;

Cards;

Hello students

I am samuel

I teach sas

;

run;

data mylib.char1;

set mylib.char;

len=length(string);

run;

proc print data=mylib.char1;

run;

\* INPUT and PUT Function

INPUT Function is used to convert character variable to numeric.

new\_num=input(character-variable, length.);

data mylib.temp;

input str $;

cards;

12345

78967

12390

;

run;

data mylib.temp1;

set mylib.temp;

num = input(str, 5.);

run;

\* see output data window;

\* PUT Function is used to convert numeric variable to character.

new\_char=put(numeric, length);

data mylib.temp;

input num;

cards;

12345

78967

12390

;

run;

data mylib.temp1;

set mylib.temp;

str = put(num, 5.);

run;

\* see output data window;

\* COUNT Function - It counts the number of times that a specified substring appears

within a character string.;

Data mylib.char;

input name $20.;

datalines;

rAjeshkumar

hArisingh

;

run;

data mylib.char1;

set mylib.char;

x = count(name,"a");

x1 = count(name,"a","i");

run;

proc print data=mylib.char1;

run;

\* The 'i' modifier ignores case sensitivity;

\* COUNTW Function

It counts the number of words in a character string.;

data mylib.char;

input name$15.;

cards;

Trait Jhonson

RajeshKumar

;

run;

data mylib.char1;

set mylib.char;

x = countw(name);

run;

proc print data=mylib.char1;

run;

\* Numeric Functions;

data mylib.num;

input a b c;

cards;

1 2 3

4 5 6

7 8 9

;

run;

data mylib.num1;

set mylib.num;

sum\_num = sum(a,b,c);

min\_num= min(a,b,c);

max\_num = max(a,b,c);

sq\_nuw= sqrt(a);

run;

proc print data=mylib.num1;

run;