/\* 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 fname $ lname $;

datalines;

rajesh kumar

hari singh

;

run;

Data mylib.char1;

Set mylib.char;

char1 = lowcase(fname);

char2 = upcase(fname, lname);

char3 = propcase(fname);

run;

proc print data=mylib.char1;

run;

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

Data mylib.char;

input fname $ lname $;

datalines;

rajesh kumar

hari singh

;

run;

Data mylib.char1;

Set mylib.char;

char1 = index(fname, '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;

\* SAS FORMATS AND INFORMATS

Informat is an instruction that SAS uses to read datavalues into a variable.

Format is an instruction that SAS uses to display or write datavalues into a variable.

Syntax : <$>informat/format<W>.<d>

$: indicates a character informat/format

informat/format: names the informat/format

w: specifies the current width which for most informat is the no. of column in the input data

d: specifies an optional decimal scaling factor in the numeric informats

;

data mylib.abc;

informat varname comma13.2;

format varname dollar14.2;

input varname;

cards;

10,000,000.23

;

run;

Proc print data=mylib.abc;

run;

data mylib.abc;

informat varname ddmmyy10.;

format varname ddmmyy10.;

input varname;

cards;

12-01-1599

13-02-2001

01-12-1760

;

run;

Proc print data=mylib.abc;

run;

data mylib.abc;

informat varname ddmmyy10.;

format varname ddmmyy8.;

input varname;

cards;

12-01-1599

13-02-2001

01-12-1760

;

run;

Proc print data=mylib.abc;

run;

data mylib.abc;

informat varname ddmmyy10.;

format varname WORDDATE.;

input varname;

cards;

12-01-1599

13-02-2001

01-12-1760

;

run;

Proc print data=mylib.abc;

run;

data mylib.abc;

informat varname ddmmyy10.;

format varname WEEKDATE.;

input varname;

cards;

12-01-1599

13-02-2001

01-12-1760

;

run;

Proc print data=mylib.abc;

run;

data mylib.abc;

informat varname ddmmyy10.;

format varname date9.;

input varname;

cards;

12-01-1599

13-02-2001

01-12-1760

;

run;

Proc print data=mylib.abc;

run;

\* Date Formats

Formats Result

DDMMYYP10. 30.12.2016

DDMMYYS10. 30/12/2016

MMDDYYP10. 12.30.2016

MMDDYYS10. 12/30/2016

WORDDATX19. 30 DECEMBER 2016

;

data mylib.abc;

format varname words40.;

input varname;

cards;

102

2001

1760

;

run;

Proc print data=mylib.abc;

run;

data mylib.abc;

format varname roman10.;

input varname;

cards;

102

2001

1760

;

run;

Proc print data=mylib.abc;

run;

data mylib.abc;

format varname percent10.;

input varname;

cards;

0.1

0.05

0.45

;

run;

Proc print data=mylib.abc;

run;

\* User Defined Format

Example

a) Display "MALE" and "FEMALE" instead of "M" and "F"

b) Re-define categories A, B, C, D and E as Ultra, Super, Average, Low and Poor

c) Display the frequency of Salesamount in three categories "< 8000",

"8000-12000" and ">=12000"

;

Data mylib.sales;

input ID Gender $ SalesAmount Category $;

cards;

1 M 10000 A

2 F 11600 B

3 M 16000 C

4 F 18000 D

5 M 6000 E

;

run;

Proc print data = mylib.sales;

run;

Proc Format;

Value $Genderfmt 'M'='Male'

'F' ='Female';

Run;

Proc Print Data = mylib.sales ;

Format Gender $Genderfmt. ;

Run;

proc format;

value Salegrp 12000-high='High'

8000-< 12000='Medium'

Low-8000 ='Low';

run;

/\* Example

2600-4000 => Between 2600 to 4000 including 2600 and 4000

2600 <-4000 => Between 2600 to 4000 including 4000 and excluding 2600

2600 -< 4000 => Between 2600 to 4000 including 2600 and excluding 4000

2600 <-< 4000 => Between 2600 to 4000 excluding 2600 & 4000

\*/

Proc Print Data = mylib.sales;

Format Salesamount Salegrp.;

Run;

\* multiple formats in one proc format statement;

Proc Format;

value Salegrp 12000-high='High'

8000-< 12000='Medium'

Low-8000 ='Low';

Value $Genderfmt 'M'='Male'

'F' ='Female';

Run;

Proc Print Data=mylib.Sales;

Format Salesamount Salegrp. Gender $Genderfmt.;

Run;

\* Save formats for future use;

Proc Format library=mylib.Gender\_Fmt;

Value $Genderfmt 'M'='Male'

'F' ='Female';

Run;

Options fmtsearch = (mylib.Gender\_Fmt); /\*After this we can use Gender\_Fmt\*/

Proc Print data=mylib.Sales;

Format Gender $Genderfmt.;

Run;

\* SAS PICTURE FORMAT

SAS picture format creates templates in which we define how the numbers are displayed.

With use of PICTURE FORMAT, we can get over multiple display issues of numbers like:-

1. Decimal and Comma Placement

2. Embedding Characters with Numbers

3. Prefixes

;

\* Example - Display the sales amount ending with a % sign and preceded with $ sign;

Proc Format;

Picture New\_fmt low-High='000000%' (Prefix='$');

Run;

Proc print data=mylib.Sales;

Format Salesamount New\_fmt.;

Run;

Data mylib.Telephone;

Input Contact;

datalines;

1111111111

222222222

3333333333

;

Run;

Proc Format;

Picture Tele low-High='000-0000-000';

Run;

Proc print data=mylib.Telephone;

Format Contact Tele.;

Run;

\* SAS Date Functions;

Data mylib.dated;

informat dob ddmmyy10.;

format dob date9.;

input ID dob;

cards;

1 30/12/1988

2 17/08/1985

3 20/06/1999

4 19/01/2007

5 21/02/2009

;

run;

data mylib.dated1;

set mylib.dated;

dy = day(dob);

mnt = month(dob);

yr = year(dob);

qtr = qtr(dob);

weekdy = weekday(dob);

run;

Proc print data=mylib.dated1;

run;

\* The INTCK is one of the most important date function that is used to calculate the

difference between two dates, two times or two datetime values.

The following is a list of common real-world examples where INTCK is used -

Calculation of individual's age

Tenure of an employee with company

Customer's tenure with the organization

Number of working days

Number of hours spent on a particular course

Number of quarterly payments paid

INTCK(date-or-time-interval, start-date-or-time, end-date-or-time, [method])

1. date-or-time-interval : Date or time period needs to be defined in the first parameter.

For eg. MONTH, YEAR, QTR, WEEK, HOUR, MINUTE etc. Specify period in single quotes

2. start-date-or-time : Starting date or time to calculate the number of periods.

3. end-date-or-time : End date or time to calculate the number of periods.

4. method : Optional Parameter. Method to calculate the difference.

Methods are 'CONTINUOUS' or 'DISCRETE'. By default, it is DISCRETE.

;

Data mylib.dated;

informat date1 ddmmyy10. date2 ddmmyy10.;

format date1 date9. date2 date9.;

input date1 date2;

cards;

12/01/1995 30/12/1988

27/06/1987 17/08/1985

30/12/1992 20/06/1999

04/04/2000 19/01/2007

05/06/2003 21/02/2009

;

run;

Data mylib.dated1;

set mylib.dated;

no\_of\_years = intck ('YEAR', date1, date2);

no\_of\_years\_1 = intck ('YEARS', date1, date2);

no\_of\_years\_2 = intck ('yr', date1, date2);

no\_of\_semiyears = intck ('SEMIYEAR', date1, date2);

no\_of\_quarters = intck ('QUARTER', date1, date2);

no\_of\_months = intck ('MONTH', date1, date2);

no\_of\_weeks = intck ('WEEK', date1, date2);

no\_of\_days = intck ('DAY', date1, date2);

run;

Proc print data=mylib.dated1;

run;

\* calculate the number of 4 months interval between two dates;

Data mylib.dated1;

set mylib.dated;

no\_of\_4months = intck ('MONTH4', date1, date2);

run;

Proc print data=mylib.dated1;

run;

\* calculate the number of 2 years interval between two dates;

Data mylib.dated1;

set mylib.dated;

no\_of\_2YEARS = intck ('YEAR2', date1, date2);

run;

Proc print data=mylib.dated1;

run;

\* another example;

Data mylib.dated1;

set mylib.dated;

no\_of\_YEARS = intck ('YEAR', date1, date2);

no\_of\_YEARS\_1 = intck ('YEAR.3', date1, date2);

run;

Proc print data=mylib.dated1;

run;

\* intck ('YEAR', date1, date2) - It checks number of times first of January appears as first

of january is set as a starting point by default. The variable diff returns 1 as it includes

only 01JAN 2016.

\* intck ('YEAR.3', date1, date2) - It checks number of times first of March appears as

YEAR.3 refers to the period starting from 1st of March to end of February next year.

The variable diff2 returns 2 as it includes 01 March 2015 and 01March 2016.

;

Data mylib.dated;

informat date1 date9. date2 date9.;

format date1 date9. date2 date9.;

input date1 date2;

cards;

25OCT2016 03NOV2016

01OCT2016 31OCT2016

;

run;

Data mylib.dated1;

set mylib.dated;

mth = intck('month', date1, date2);

run;

proc print data=mylib.dated1;

run;

\* INTCK checks whether the first day of the month lies with in the range.

In the first case, 01 November falls between October 25 and November 03 so it returns 1.

In the second case, it returns 0 as 01 November does not fall between 01OCT2016 and 31OCT2016.

;

\* Correction - Add one more parameter at end of INTCK function. In the parameter,

specify 'C' which refers to continuous method for calculation.;

Data mylib.dated1;

set mylib.dated;

mth = intck('month', date1, date2, 'C');

run;

proc print data=mylib.dated1;

run;

\* The CONTINUOUS method calculates continuous time from the start-of-period date

specified in the second parameter of INTCK function.;

\* Calculating Weekdays;

data eg;

weekdays = intck('WEEKDAY', '11DEC2016'd ,'18DEC2016'd);

proc print;

run;

Data mylib.dated;

informat date1 date9. date2 date9.;

format date1 date9. date2 date9.;

input date1 date2;

cards;

06NOV2022 12NOV2022

;

run;

Data mylib.dated1;

set mylib.dated;

wkdy = intck('WEEKDAY', date1, date2);

run;

proc print data=mylib.dated1;

run;

\* It returns 5. In this case, saturday and sunday are considered weekends and

excluding from the calculation.;

\* If you need to calculate number of working days between 2 dates considering 6 weekdays;

Data mylib.dated1;

set mylib.dated;

wkdy = intck('WEEKDAY1W', date1, date2);

run;

proc print data=mylib.dated1;

run;

\* WEEKDAY1W implies sunday as weekend (1=Sunday, 2= MONDAY... 7=Saturday);

Data mylib.dated1;

set mylib.dated;

wkdy = intck('WEEKDAY24W', date1, date2);

run;

proc print data=mylib.dated1;

run;

\* WEEKDAY24W means MONDAY and WEDNESDAY are weekends.;

\* Calculate between Datetime values;

data mylib.dt;

hours=intck('hour','01jan2016:10:50:00'dt,'01jan2016:11:55:00'dt);

minutes=intck('minute','01jan2016:10:50:00'dt,'01jan2016:11:55:00'dt);

seconds=intck('second','01jan2016:10:50:00'dt,'01jan2016:11:55:00'dt);

run;

proc print data=mylib.dt;

run;

\* Time Difference;

data mylib.dt;

hours=intck('hour','12:00:00't, '23:05:00't);

minutes=intck('minute','12:00:00't,'23:05:00't);

seconds=intck('second','12:00:00't,'23:05:00't);

run;

proc print data=mylib.dt;

run;

\* INTNX FUNCTION WITH EXAMPLES

INTNX is used to increment SAS date by a specified number of intervals.

It helps to answer the following questions.

Examples

When is next Monday?

When was last Friday?

What would be date after 21 weeks?

Subtract 2 quarters from the current date

INTNX(interval, start-from, increment, [alignment])

Interval is the unit of measurement. The intervals can be days, weeks, months,

quarters, years.

Start-from is a SAS date value which would be incremented.

Increment is number of intervals by which date is incremented. It can be zero,

positive or negative. Negative value refers to previous dates.

Alignment [Optional Parameter] is where datevalue is aligned within interval prior to

being incremented. The values you can specify - 'beginning', 'middle', 'end', 'sameday'.

Default value - 'beginning'.

;

\* Add 7 days to a specific date;

Data mylib.dated;

informat dob ddmmyy10.;

format dob date9.;

input ID dob;

cards;

1 30/12/1988

2 17/08/1985

3 20/06/1999

4 19/01/2007

5 21/02/2009

;

run;

data mylib.dated1;

set mylib.dated;

format day date9.;

day=intnx('day', dob , 7);

run;

proc print data=mylib.dated1;

run;

\* Find Next Sunday - In this case, we need to find answer of the question

'when is next sunday?'. The 02 November,2022 is Wednesday.;

Data mylib.dated;

informat dob ddmmyy10.;

format dob date9.;

input ID dob;

cards;

1 02/11/2022

;

run;

data mylib.dated1;

set mylib.dated;

format nextsunday date9.;

nextsunday=intnx('week', dob , 1);

run;

proc print data=mylib.dated1;

run;

\* It returns 06Nov2022 as it aligns to the 'beginning' period. The 'beginning' alignment

is default in INTNX function. In other words, if you change the mydate to '04Nov2022',

it still returns '06Nov2022' as the next sunday would be same within this week interval.

If you want to add exactly 1 week to the date, you can use the 'sameday' in the fourth

parameter of this function. See the statement below -

weekinc=intnx('week', mydate , 1, 'sameday') returns 09Nov2022

;

data mylib.dated1;

set mylib.dated;

format weekinc date9.;

weekinc=intnx('week', dob , 1, 'sameday');

run;

proc print data=mylib.dated1;

run;

\* Get First Date

Suppose you need to find out the first day of a specific day. For example, today is

02November, 2022 and the first day of this date is 01January,2022 on year basis.;

data mylib.dated1;

set mylib.dated;

format firstdateyr date9. firstdatemonth date9. firstdateweek date9.;

firstdateyr=intnx('year', dob , 0);

firstdatemonth=intnx('month', dob , 0);

firstdateweek=intnx('week', dob , 0);

run;

proc print data=mylib.dated1;

run;

\* By specifying 0 in the third parameter of INTNX function, we can calculate the first

day of the dates.

\* When was Last Tuesday?

It is tricky to figure out the date when it was last tuesday. 02November,2022 is Wednesday.

In real world dataset, we don't have the exact days of a list of dates when we need to

code to get the last tuesday.;

data mylib.dated1;

set mylib.dated;

format lasttuesday date9.;

lasttuesday = intnx('week.4', dob , -1, 'end');

run;

proc print data=mylib.dated1;

run;

\* It returns 01Nov2022 which is previous tuesday. See the changes we have made in this program -

a) -1 instead of 0 as increment value

b) 'end' instead of 'beginning' as date alignment

c) 'week.4' instead of 'week.3' to figure out the last tuesday

;

\* Adjustment within the Interval

This program explains how INTCK function adjusts / align dates within the interval specified.;

data mylib.dated1;

set mylib.dated;

format beginning date9. middle date9. endd date9. sameday date9.;

beginning=intnx('year ', dob , 1, 'b');

middle=intnx('year ', dob , 1, 'm');

endd=intnx('year ', dob , 1, 'e');

sameday=intnx('year ', dob , 1, 's');

run;

proc print data=mylib.dated1;

run;

\* The abbreviation 'b' refers to beginning, 'm' - middle, 'e' - end, 's' - sameday.

The default value is 'b' if you don't specify anything in the fourth parameter.

Result

beginning = 01JAN2023

middle = 02JUL2023

end = 31DEC2023

sameday = 02NOV2023

;

\* SAS INTNX Alignment

Datetime Formats

Like date formats, we can use time and datetime formats in INTNX function to increment

time (seconds / minutes / hours).;

data mylib.datetime;

informat mydt datetime18.;

format mydt datetime20.;

input mydt;

cards;

02NOV2022:08:34:00

;

run;

data mylib.datetime1;

format seconds minutes hours days weeks datetime20.;

set mylib.datetime;

seconds=intnx('second', mydt , 1);

minutes=intnx('minute', mydt , 1);

hours=intnx('hour', mydt , 1);

days=intnx('dtDay', mydt , 1);

weeks=intnx('dtWeek', mydt , 1);

run;

proc print data=mylib.datetime1;

run;