3.sasFunctions

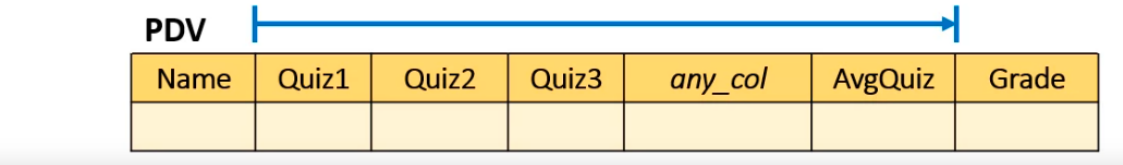
March 3, 2025

0.0.1 Manipulating Data With Functions

In Manipulating Data with Functions you learn to use some new functions that enable you to manipulate numeric, date, and character values. In addition, you learn to use functions that change a column from one data type to another.

```
data quiz_summary;
    set pg2.class_quiz;
    Name=upcase(Name);
    AvgQuiz=mean(of Q:);
    format Quiz1--AvgQuiz 3.1;
    /*OR*/
    format _numeric_ 3.1;
run;
```

col--col



Understanding SAS Functions and CALL Routines

function(*argument1*, *argument2*, ...);
CALL *routine*(*argument-1* <, ...*argument-n*>);

Original Quiz Scores

Name	Quiz1	Quiz2	Quiz3	Quiz4	Quiz5
Alfred	8	7	6	9	8
Alice	7	6	4	9	8
Barbara	9	8	7	.	7
Carol	6	5	5	8	8

```
data quiz_report;
     set pg2.class_quiz;
     call sortn(of Quiz1-Quiz5);
     QuizAvg=mean(of Quiz3-Quiz5);
run;
```

Modified Quiz Scores

Name	Quiz1	Quiz2	Quiz3	Quiz4	Quiz5	QuizAvg
Alfred	6	7	8	8	9	8.33
Alice	4	6	7	8	9	8.00
Barbara	.	7	7	8	9	8.00
Carol	5	5	6	8	8	7.33

0.0.2 Numeric Functions

Using Numeric and Date Functions

- The **RAND function** generates random numbers from a selected distribution. The first argument specifies the distribution, and the remaining arguments differ depending on the distribution. To generate a random, uniformly distributed integer, use 'INTEGER' as the first argument. The second and third arguments are the lower and upper limits.
- The **LARGEST function** returns the kth largest nonmissing value. The first argument is the value to return, and the remaining arguments are the numbers to evaluate. There is also a SMALLEST function that returns the kth smallest nonmissing value.
- The **ROUND function** rounds the first argument to the nearest integer. The optional second argument can be provided to indicate the rounding unit.

RAND(*'distribution', parameter1, ...parameterk*)
LARGEST(*k, value-1 <, value-2 ...>*)
ROUND(*number <, rounding-unit>*)

- These functions can be used to truncate decimal values:

Function	What it Does
CEIL (*number*)	Returns the smallest integer that is greater than or equal to the argument.
FLOOR (*number*)	Returns the largest integer that is less than or equal to the argument.
INT (*number*)	Returns the integer value.

- These functions can be used the extract a date or time component of a datetime value:

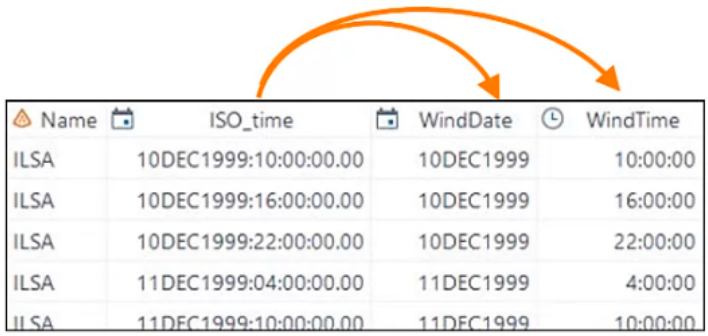
DATEPART(*datetime-value***)**
TIMEPART(*datetime-value***)**

- This function can be used to count the number of intervals that have occured between a start and end date. You can specify 'C' to use the continuous method for counting intervals:

INTCK('*interval***',***start-date,end-date* **<,'***method***'>)**

- This function can be used to adjust or shift date values:

INTNX('*interval***',***start,increment* **<,'***alignment***'>)**

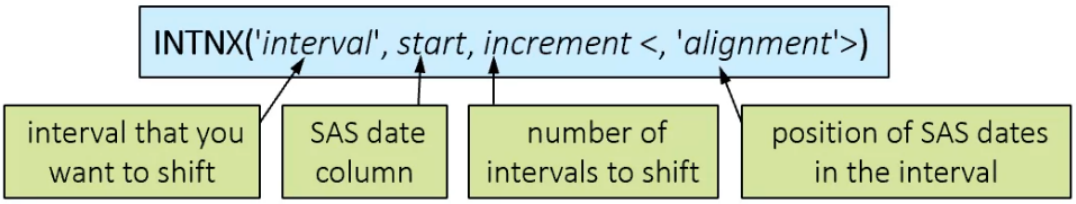


Name	ISO_time	WindDate	WindTime
ILSA	10DEC1999:10:00:00.00	10DEC1999	10:00:00
ILSA	10DEC1999:16:00:00.00	10DEC1999	16:00:00
ILSA	10DEC1999:22:00:00.00	10DEC1999	22:00:00
ILSA	11DEC1999:04:00:00.00	11DEC1999	4:00:00
ILSA	11DEC1999:10:00:00.00	11DEC1999	10:00:00

DATEPART(*datetime-value*)

TIMEPART(*datetime-value*)

```
data storm_detail2;
    set pg2.storm_detail;
    WindDate=datepart(ISO_Time);
    WindTime=timepart(ISO_Time);
    format WindDate date9. WindTime time.;
run;
```



0.0.3 Character Functions

Using Character Functions

- These functions can be used to remove characters from a string:

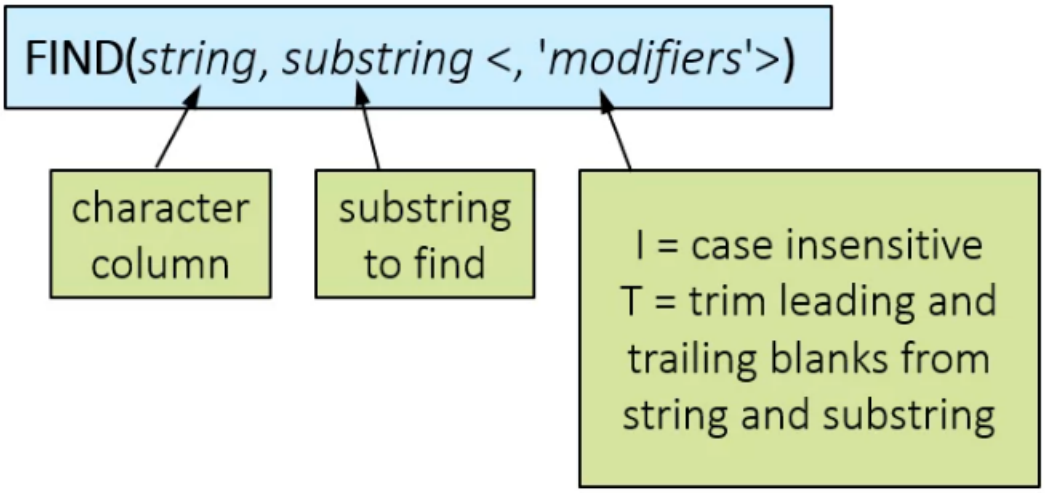
Function	What it does
COMPBL(*string***)**	Returns a character *string* with all multiple blanks in the source string converted to single blanks
COMPRESS (*string* **<,** *characters***>)**	Returns a character string with specified *characters* removed from the source string
STRIP(*string***)**	Returns a character *string* with leading and trailing blanks removed

- The **SCAN function** returns the nth word in a string. If *n* is negative, the SCAN function begins reading from the right side of the string.
  The default delimiters are as follows: blank ! $ % & ( ) * + , - . / ; < ^ |
  The optional third argument enables you to specify a delimiter list. All delimiter characters are enclosed in a single set of quotation marks.

- The **PROPCASE function** converts all uppercase letters to lowercase letters. It then converts to uppercase the first character of each word.
  The default delimiters are as follows: blank / - (. tab
  The optional second argument enables you to specify a delimiter list. All delimiter characters are enclosed in a single set of quotation marks.

SCAN(*string, n* **<,** *'delimiters'***>)**
PROPCASE(*string* **<,** *'delimiters'***>)**

FIND(*string, substring* **<,** *'modifiers'***>)**



- These functions return a numeric value that identifies the location of selected characters:

Function	What it does
LENGTH(*string***)**	Returns the length of a non-blank character string, excluding trailing blanks, returns 1 for a completely blank string
ANYDIGIT(*string***)**	Returns the first position at which a digit is found in the string
ANYALPHA(*string***)**	Returns the first position at which an alpha character is found in the string
ANYPUNCT(*string***)**	Returns the first position at which punctuation character is found in the string

TRANWRD(*source***, *target*, *replacement*)**

- These functions can be used to combine strings into a single character value. The arguments can be either character or standard numeric values.

Function	What it does
CAT(*string1***, ... *stringn*)**	Concatenates strings together, does not remove leading or trailing blanks
CATS(*string1***, ... *stringn*)**	Concatenates strings together, removes leading or trailing blanks from each string
CATX('*delimiter***', *string1*, ... *stringn*)**	Concatenates strings together, removes leading or trailing blanks from each string, and inserts the delimiter between each string

TRANWRD(*source, target, replacement*)

```
Summary2=tranwrd(Summary, 'hurricane', 'storm');
```

Summary	Summary2
Category 3 hurricane initially...	Category 3 storm initially...
The largest (in size) Atlantic hurricane on record...	The largest (in size) Atlantic storm on record...

0.0.4 Converting Column Type

Using Special Functions to Convert Column Type

```
DATA output-table;
    SET input-table (RENAME=(current-column=new-column));
    ...
    column1 = INPUT(source, informat);
    column2 = PUT(source, format);
    ...
RUN;
```

- The **INPUT function** converts a character value to a numeric value using a specified informat. SAS automatically tries to convert character values to numeric values using the w. informat.
- The **PUT function** converts a numeric or character value to a character value using a specified format. SAS automatically tries to convert numeric values to character values using the BEST12. format.
- If SAS automatically converts the data, a note is displayed in the SAS log. If you explicitly tell SAS to convert the data with a function, a note is not displayed in the SAS log.
- Some functions such as the CAT functions automatically convert data from numeric to character and also remove leading blanks on the converted data. No note is displayed in the SAS log.

```
data work.stocks2;
     set pg2.stocks2(rename=(Volume=CharVolume));
     Date2=input(Date,date9.);
     Volume=input(CharVolume,comma12.);
     drop CharVolume;
run;
```

PDV

Stock $ 12	Date $ 9	Open N 8	Close N 8	High $ 6	Low N 8	CharVolume $ 12	Date2 N 8	Volume N 8

Character Value to Convert	Informat for the INPUT Function	Numeric Value Returned
15OCT2018	DATE9.	21472
10/15/2018	MMDDYY10.	21472
15/10/2018	DDMMYY10.	21472
123,456.78	COMMA12.	123456.78
$123,456.78	DOLLAR12.	123456.78
123456	6.	123456

Numeric Value to Convert	Format for the PUT Function	Character Value Returned
21472	DATE9.	15OCT2018
21472	DOWNAME3.	Mon
21472	YEAR4.	2018
123456.78	COMMA10.2	123,456.78
123456.78	DOLLAR11.2	$123,456.78
123.456	6.2	123.46

[ ]: