Summary of Lesson 3: Manipulating Data with Functions

Understanding SAS Functions and CALL Routines

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

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_'>**)**

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

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.