

## Summary of Lesson 4: Preparing Data

---

### Reading and Filtering Data

- Creating a copy of data:

```
DATA output-table;  
  SET input-table;  
RUN;
```

- Filtering rows in the DATA step:

```
DATA output-table;  
  SET input-table;  
  WHERE expression;  
RUN;
```

- Specifying columns to include in the output data set:

```
DROP col-name <col-name>;
```

```
KEEP col-name <col-name>;
```

- Formatting columns in the DATA step:

```
DATA output-table;  
  SET input-table;  
  FORMAT col-name format;  
RUN;
```

### Computing New Columns

- Using expressions to create new columns:

```
DATA output-table;  
  SET input-table;  
  new-column = expression;  
RUN;
```

- The name of the column to be created or updated is listed on the left side of the equals sign.
- Provide an expression on the right side of the equal sign.
- SAS automatically defines the required attributes if the column is new – name, type, and length.
- A new numeric column has a length of 8.
- The length of a new character column is determined based on the length of the assigned string.
- Character strings must be quoted and are case sensitive.
- Creating character columns:

```
LENGTH char-column $ length;
```

- Using functions in expressions:

```
function(argument1, argument 2, ...);
```

```
DATA output-table;  
  SET input-table;  
  new-column=function(arguments);  
RUN;
```

- Functions for calculating summary statistics (ignore missing values):

<b>SUM</b> ( <i>num1, num2, ...</i> )	calculates the sum
<b>MEAN</b> ( <i>num1, num2, ...</i> )	calculates the mean
<b>MEDIAN</b> ( <i>num1, num2, ...</i> )	calculates the median
<b>RANGE</b> ( <i>num1, num2, ...</i> )	calculates the range
<b>MIN</b> ( <i>num1, num2, ...</i> )	calculates the minimum
<b>MAX</b> ( <i>num1, num2, ...</i> )	calculates the maximum
<b>N</b> ( <i>num1, num2, ...</i> )	calculates the nonmissing
<b>NMISS</b> ( <i>num1, num2, ...</i> )	calculates the missing

- Character functions:

<b>UPCASE</b> ( <i>char1</i> ) <b>LOWCASE</b> ( <i>char1</i> )	changes letters in a character string to uppercase or lowercase
<b>PROPCASE</b> ( <i>char1</i> )	changes the first letter of each word to uppercase and other letters to lowercase
<b>CATS</b> ( <i>char1, char2, ...</i> )	concatenates character strings and removes leading and trailing blanks from each argument
<b>SUBSTR</b> ( <i>char, position, &lt;length&gt;</i> )	returns a substring from a character string

- Date functions that extract information from SAS date values:

<b>MONTH</b> ( <i>sas-date-value</i> )	returns a number from 1 through 12 that represents the month
<b>YEAR</b> ( <i>sas-date-value</i> )	returns the four-digit year
<b>DAY</b> ( <i>sas-date-value</i> )	returns a number from 1 through 31 that represents the day of the month
<b>WEEKDAY</b> ( <i>sas-date-value</i> )	returns a number from 1 through 7 that represents the day of the week (Sunday=1)
<b>QTR</b> ( <i>sas-date-value</i> )	returns a number from 1 through 4 that represents the quarter

- Date functions that create SAS date values:

<b>TODAY</b> ()	returns the current date as a numeric SAS date value
<b>MDY</b> ( <i>month, day, year</i> )	returns SAS date value from month, day, and year values
<b>YRDIF</b> ( <i>startdate, enddate, 'AGE'</i> )	calculates a precise age between two dates. There are various values for the third argument. However, "AGE" should be used for accuracy.

## Conditional Processing

- Conditional processing with IF-THEN logic:

```
IF expression THEN statement;
```

- Conditional processing with IF-THEN-ELSE:

```
IF expression THEN statement;  
<ELSE IF expression THEN statement>;  
<ELSE IF expression THEN statement>;  
ELSE statement;
```

- Processing multiple statements with IF-THEN-DO:

```
IF expression THEN DO;  
  <executable statements>  
END;  
<ELSE IF expression THEN DO;  
  <executable statements>  
END;>  
ELSE DO;  
  <executable statements>  
END;
```

- After the IF-THEN-DO statement, list any number of executable statements.
- Close each DO block with an END statement.