Manipulating Data with Functions

1. Modify the **WindAvg1** expression to use the ROUND function to round values to the nearest tenth (.1).

   ```
   WindAvg1=round(mean(of Wind1-Wind4), .1);
   ```

2. Add a FORMAT statement to format **WindAvg2** with the 5.1 format. Run the program. What is the difference between using a function and a format?

   ```
   format WindAvg2 5.1;
   ```

   The values appear the same, but the function changes the stored values, whereas the format affects only the displayed values.

3. Complete the **NewLocation** assignment statement to use the COMPBL function to read **Location** and convert each occurrence of two or more consecutive blanks into a single blank.

   ```
   data weather_japan_clean;
      set pg2.weather_japan;
      NewLocation=compbl(Location);
   run;
   ```

4. Complete the **NewStation** assignment to use the COMPRESS function with **Station** as the only argument. Run the program. Which characters are removed in the **NewStation** column?

   ```
   data weather_japan_clean;
      set pg2.weather_japan;
      NewLocation=compbl(Location);
      NewStation=compress(Station);
   run;
   ```

   Blanks are removed.

5. Add a second argument in the COMPRESS function to specify the characters to remove. All characters should be enclosed in a single set of quotation marks. Run the program.

   ```
   data weather_japan_clean;
      set pg2.weather_japan;
      NewLocation=compbl(Location);
      NewStation=compress(Station,"- ");
   run;
   ```

6. Modify the FIND function to make the search ***case insensitive***. Uncomment the IF-THEN statement to create a new column named **Category**. Run the program and examine the results. How are values for **Category** generated?

```
data storm_damage2;
  set pg2.storm_damage;
  drop Date Cost Deaths;
  CategoryLoc=find(Summary, 'Category', 'i');
  if CategoryLoc > 0 then
    Category=substr(Summary, CategoryLoc, 10);
run;
```

   The SUBSTR function starts at the number stored in **CategoryLoc** and reads 10 characters, and returns the string to **Category**.

7. Add an assignment statement to create **StormID3** that uses the CATX function to concatenate **Name**, **Season**, and **Day** with a hyphen inserted between each value.

```
StormID3=catx("-", Name, Season, Day);
```

8. Modify the **StormID2** assignment statement to insert a hyphen only between **Name** and **Season**.

```
StormID2=cats(Name, '-', Season, Day);
```

9. Add an assignment statement to create a column named **Volume2**. Use the INPUT function to read **Volume** using the COMMA12. informat. Run the program and verify that **Volume2** is created as a numeric column.

```
data work.stocks2;
  set pg2.stocks2;
  Date2=input(Date,date9.);
  Volume2=input(Volume,comma12.);
run;
```

10. Perform following tasks:
    a) Add to the RENAME= option to rename the input column Date as CharDate.
    b) Add an assignment statement to create a numeric column Date from the character column CharDate. The values of CharDate are stored as 01JAN2018.
    c) Modify the DROP statement to eliminate all columns that begin with Char from the output table.

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