Challenge Practice: Processing Statements Conditionally with SELECT-WHEN Groups

SELECT and WHEN statements can be used in a DATA step to process code conditionally as an alternative to IF-THEN statements .

Reminder: If you restarted your SAS session,you must recreate the **PG1** library so you can access your practice files. In SAS Studio, open and submit the **libname.sas** program in the **EPG194** folder. In Enterprise Guide, run the **Autoexec** process flow.

1. Use SAS Help or online documentation to read about using SELECT and WHEN statements in the DATA step.

2. Modify the following program (from the Level 2 practice) to use SELECT and WHEN statements instead of IF-THEN statements.

   When **Type** is *NP*, create a new column named **ParkType** that is equal to **Park**, and write the row to the **parks** table. When **Type** is *NM*, assign **ParkType** as **Monument** and write the row to the **monuments** table.

```
data parks monuments;
    set pg1.np_summary;
    where type in ('NM', 'NP');
    Campers=sum(OtherCamping, TentCampers, RVCampers,
                BackcountryCampers);
    format Campers comma17.;
    length ParkType $ 8;
    if type='NP' then do;
        ParkType='Park';
        output parks;
    end;
    else do;
        ParkType='Monument';
        output monuments;
    end;
    keep Reg ParkName DayVisits OtherLodging Campers ParkType;
run;
```

```
 data parks monuments;
     set pg1.np_summary;
     where type in ('NM', 'NP');
     Campers=sum(OtherCamping, TentCampers, RVCampers,
                 BackcountryCampers);
     format Campers comma17.;
     length ParkType $ 8;
     select (type);
         when ('NP') do;
             ParkType='Park';
             output parks;
                 end;
                 otherwise do;
             ParkType='Monument';
             output monuments;
                 end;
     end;
```

```
    keep Reg ParkName DayVisits OtherLodging Campers ParkType;
run;
```

3. Submit the program and verify that **work.parks** contains 51 rows and **work.monuments** contains 63 rows.