

```

1 *****;
2 * Identifying and Removing Duplicate Values *;
3 *****;
4 * Syntax and Example *;
5 * *;
6 * Remove duplicate rows: *;
7 * PROC SORT DATA=input-table <OUT=output-table> *;
8 *     NODUPRECS <DUPOUT=output-table>; *;
9 *     BY _ALL_; *;
10 * RUN; *;
11 * *;
12 * Remove duplicate key values: *;
13 * PROC SORT DATA=input-table <OUT=output-table> *;
14 *     NODUPKEY <DUPOUT=output-table>; *;
15 *     BY <DESCENDING> col-name (s); *;
16 * RUN; *;
17 *****;
18
19 *****;
20 * Demo *;
21 * 1) Modify the first PROC SORT step to sort by all *;
22 *     columns and remove any duplicate rows. Write the *;
23 *     removed rows to a table named STORM_DUPS. *;
24 *     Highlight the step and run the selected code. *;
25 *     Confirm that there are 107,821 rows in *;
26 *     STORM_CLEAN and 214 rows in STORM_DUPS. *;
27 * 2) Run the second PROC SORT step and confirm that *;
28 *     the first row for each storm represents *;
29 *     the minimum value of Pressure. *;
30 *     Note: Because storm names can be reused in *;
31 *     multiple years and basins, unique storms *;
32 *     are grouped by sorting by Season, Basin, *;
33 *     and Name. *;
34 * 3) Modify the third PROC SORT step to sort the *;
35 *     MIN_PRESSURE table and keep the first row for *;
36 *     each storm. You do not need to keep the removed *;
37 *     duplicates. Highlight the step and run the *;
38 *     selected code. *;
39 *****;
40
41
42 *Step 1;
43 proc sort data=pg1.storm_detail out=storm_clean noduprecs dupout=storm_dups;
44     by _ALL_;
45 run;
46
47 *Step 2;
48 proc sort data=pg1.storm_detail out=min_pressure;
49     where Pressure is not missing and Name is not missing;
50     by descending Season Basin Name Pressure;
51 run;
52

```

```
53 |
54 | *Step 3;
55 | proc sort data=min_pressure nodupkey;
56 |     by descending Season basin Name ;
    |
```