Week 8 Walkthrough
PROC SORT

James Robertson

February 21, 2025

Sorting

One of the main things we do when it comes to manipulating the **rows** of our dataset is **sorting**. When you need to know the top 10 clients by purchase volume, bottom 3 baseball teams by homerun count, or a million other things, **sorting** makes it pretty easy.

When **sorting**, we always have two options: **ascending** order or **descending** order. **Ascending** order goes from smallest to largest for **numeric** values or from A to Z for **character** values. This is what SAS will default to when sorting any variable. **Descending** order is the opposite, largest to smallest for **numeric** values or from Z to A for **character** values.

But now, *how* do we **sort** our data? One of the many nice things about SAS is that whenever you ask such a question, the answer is usually PROC and then the thing you want to do. In this case, it's PROC SORT!

Since I've already brought up baseball and it's a smorgasbord of numeric variables (and a nice way to pass an evening), let's look at the dataset SASHELP.baseball. This dataset is already built-in, so feel free to code along with me in your own SAS session!

Let's start by looking at the dataset directly. For this, our options are to use PROC PRINT or just look at the dataset directly in our libraries. I'll choose the libraries approach! Go over to the left pane and click **Libraries**, then in **My Libraries**, open up **SASHELP** and from there double-click **BASEBALL** to have a peek. From this we get an idea of what is in this dataset: players, teams, and personal stats (including salary!). As a Durham Bulls fan[1], I want to look at nHome, the number of homeruns hit and see the top players/teams, so let's **sort** by that! To do this, I'll run the following code.

```
PROC SORT DATA=SASHELP.baseball;
    BY nHome;
RUN;
```

But now we've gotten an error! More specifically we are being told

```
ERROR: User does not have appropriate authorization level for library SASHELP.
```

This error is trying to tell us that we can't save anything to SASHELP because it's a special SAS directory! This message is appearing because when we only specify DATA, SAS sorts the dataset "**in place**", meaning it saves the sorted dataset as the original dataset, overwriting the original. SAS doesn't want you to alter its special datasets! To get around this, all we just need to specify somewhere else we would like to save the sorted data via the OUT option (boxed for emphasis). Like the below!

```
PROC SORT DATA=SASHELP.baseball OUT=baseball_sorted;
    BY nHome;
RUN;
```

Great, now my code runs with no errors! But looking at my output dataset, I've sorted in **ascending** order (the default!) so right at the top are all the players with 0 home runs, the *opposite* of what I wanted! As discussed earlier, the default is **ascending** but we want it to be **descending**. This is actually pretty easy to accomplish! We just take our previous code and add the word DESCENDING before our variable. Why *before*? Well, if you ask someone to sort something, it would be incredibly rude to tell them how you want it sorted *after* they've already begun. So tell SAS beforehand!

```
PROC SORT DATA=SASHELP.baseball OUT=baseball_sorted;
    BY DESCENDING nHome;
RUN;
```

So now we have what we want, data sorted by number of home runs! As a point of curiosity, though, I'd like to see how much those players made for a salary (one of our variables!). I'm really not that interested in all the other variables at the moment and opening up the dataset in my **WORK library** gives me a bunch of columns I have to scroll past. Instead, let's use PROC PRINT to have a look at our top 10! We can do this with the OBS option as seen before, but now if we also use a

[1] These aren't Minor League stats so there are no Durham Bulls, but I love slugging and homers!

VAR statement, we can choose which columns we would like to see! I'll also include their `team` (also a variable in our data!) so I can think about how the salary compares to cost of living. The below code will print for me only these three columns of interest: `nHome`, `team`, `salary`. And it will print those columns *in that order* because I spelled it out that way!

```
PROC PRINT DATA=baseball_sorted (OBS=10);
    VAR nHome team salary;
RUN;
```

And wow do they get paid a lot of money! And here I am just dreaming of getting a salary. But they love what they do and I love what I do, so I guess the world keeps spinnin' 'round.