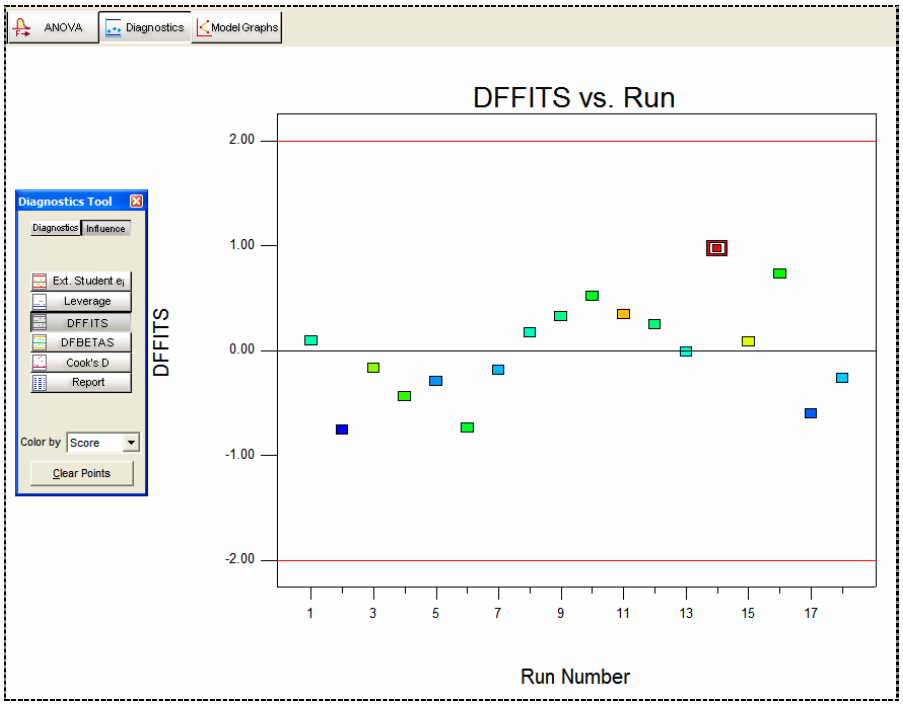
# General One-Factor Tutorial (Part 2 – Advanced Features)

## Digging Deeper Into Diagnostics

If you still have the bowling data active in Design-Ease® software from Part 1 of this tutorial, continue on. If you exited the program, re-start it and use **File**, **Open Design** to open your data file (**Bowling.de7**). Otherwise, go back and set it up as instructed in General One-Factor Tutorial (Part 1 – The Basics). Then under the **Analysis** branch (you may already be here from before) click on the **Score** node and press the ***Diagnostics*** button.

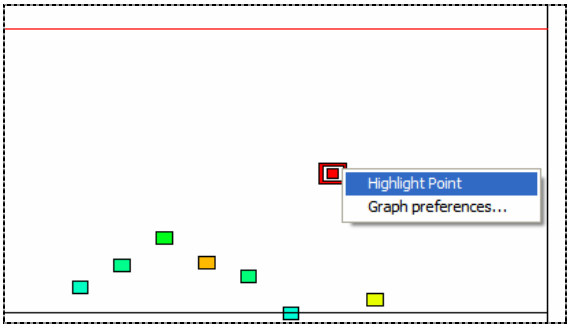
We are going to look at a new graph on the **Diagnostics Tool**. Click the **Influence** option on this floating palette. Then bring up the **DFFITS**. This statistic, which stands for difference in fits, measures the change in each predicted value that occurs when that response is deleted. The larger the absolute value of DFFITS the more it influences the fitted model. (For more details on this statistic and the related deletion diagnostic, DFBETAS, see the program Help or refer to page 284 of Raymond Myers' *Classical and Modern Regression with Applications* (Boston: Duxbury Press, 1986).)




*DFFITS graph – highest point clicked (your graph may differ due to random runs)*

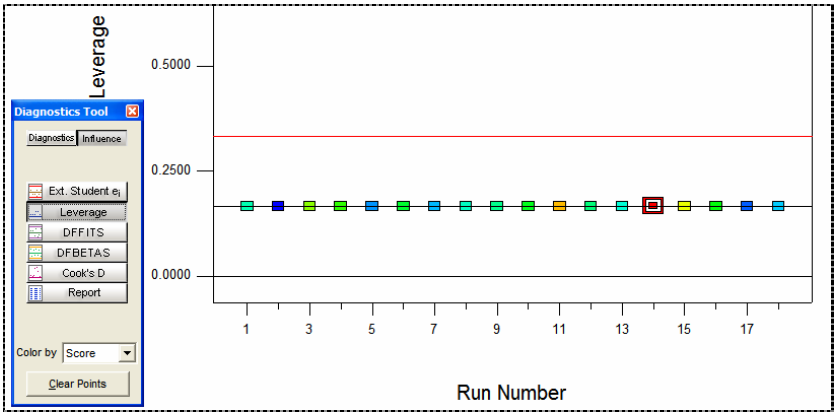
Notice that one point stands out above the rest. (The pattern on your graph may differ from what we show here due to the randomized run order, but that will not be relevant to our discussions.) This is Mark's high game, which earlier created some controversy, particularly among his competitors Pat and Shari. It falls far below a relatively

conservative benchmark of plus-or-minus two for the DFFITS. Thus, taking all the other diagnostics into consideration, we would not advise that this particular run be investigated any further. Nevertheless, for purpose of learning how to use new features in Design-Ease software, please right-click it with your mouse and select **Highlight Point**.




*Highlighting a point*

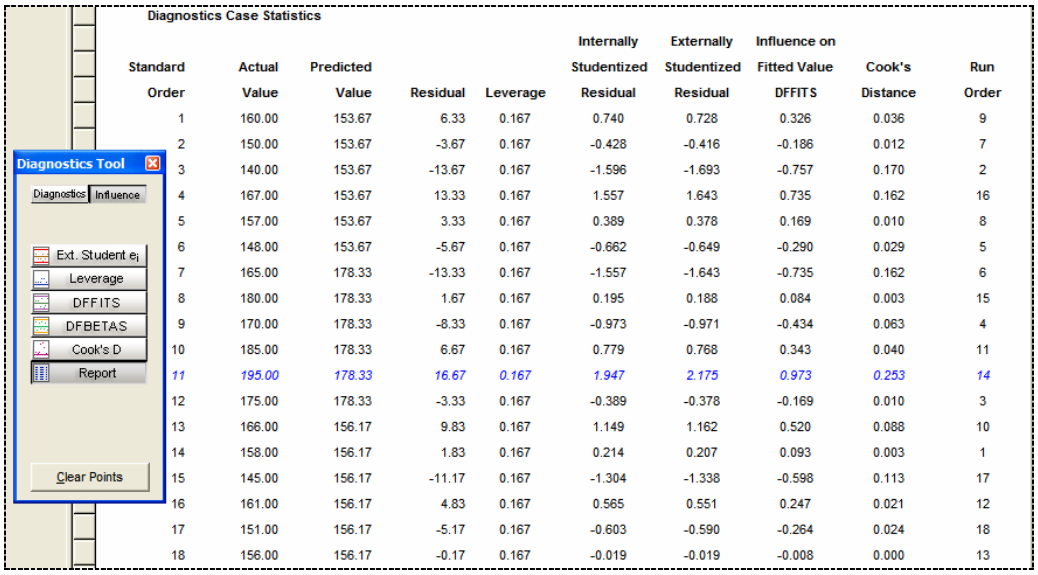
Myers demonstrates mathematically that the DFFITS statistic is really the externally studentized residual multiplied by high leverage points. As you will see by clicking the **Leverage** button, all runs exhibit equal leverage in this case because an equal number of runs were made at each treatment level (in sports talk – all three bowlers rolled six games each).




*Leverages*

Therefore, in this case the DFFITS exhibits a pattern identical to that shown on the externally studentized residual graph, which you looked at in the preceding tutorial. (If you don't recall this, look at the top plot offered, labeled "Ext. Student $e_i$", on this side of the Diagnostics Tool.) The reason we bring this up is to set the stage for what you will do later in this tutorial – unbalance the leverages to make things more interesting for diagnostic purposes.

One final 'heads-up' before we leave the **Diagnostics Tool**: Click the **Report** button to get a table of statistics case-by-case in standard order for the entire experiment. For those of you who like numbers better than pictures (statisticians for sure!), this should satisfy your appetite. Notice that Mark's high game of 195 remains highlighted.




Diagnostics Case Statistics

| Standard Order | Actual Value | Predicted Value | Residual | Leverage | Internally Studentized Residual | Externally Studentized Residual | Influence on Fitted Value DFFITS | Cook's Distance | Run Order |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 160.00 | 153.67 | 6.33 | 0.167 | 0.740 | 0.728 | 0.326 | 0.036 | 9 |
| 2 | 150.00 | 153.67 | -3.67 | 0.167 | -0.428 | -0.416 | -0.186 | 0.012 | 7 |
| 3 | 140.00 | 153.67 | -13.67 | 0.167 | -1.596 | -1.693 | -0.757 | 0.170 | 2 |
| 4 | 167.00 | 153.67 | 13.33 | 0.167 | 1.557 | 1.643 | 0.735 | 0.162 | 16 |
| 5 | 157.00 | 153.67 | 3.33 | 0.167 | 0.389 | 0.378 | 0.169 | 0.010 | 8 |
| 6 | 148.00 | 153.67 | -5.67 | 0.167 | -0.662 | -0.649 | -0.290 | 0.029 | 5 |
| 7 | 165.00 | 178.33 | -13.33 | 0.167 | -1.557 | -1.643 | -0.735 | 0.162 | 6 |
| 8 | 180.00 | 178.33 | 1.67 | 0.167 | 0.195 | 0.188 | 0.084 | 0.003 | 15 |
| 9 | 170.00 | 178.33 | -8.33 | 0.167 | -0.973 | -0.971 | -0.434 | 0.063 | 4 |
| 10 | 185.00 | 178.33 | 6.67 | 0.167 | 0.779 | 0.768 | 0.343 | 0.040 | 11 |
| *11* | *195.00* | *178.33* | *16.67* | *0.167* | *1.947* | *2.175* | *0.973* | *0.253* | *14* |
| 12 | 175.00 | 178.33 | -3.33 | 0.167 | -0.389 | -0.378 | -0.169 | 0.010 | 3 |
| 13 | 166.00 | 156.17 | 9.83 | 0.167 | 1.149 | 1.162 | 0.520 | 0.088 | 10 |
| 14 | 158.00 | 156.17 | 1.83 | 0.167 | 0.214 | 0.207 | 0.093 | 0.003 | 1 |
| 15 | 145.00 | 156.17 | -11.17 | 0.167 | -1.304 | -1.338 | -0.598 | 0.113 | 17 |
| 16 | 161.00 | 156.17 | 4.83 | 0.167 | 0.565 | 0.551 | 0.247 | 0.021 | 12 |
| 17 | 151.00 | 156.17 | -5.17 | 0.167 | -0.603 | -0.590 | -0.264 | 0.024 | 18 |
| 18 | 156.00 | 156.17 | -0.17 | 0.167 | -0.019 | -0.019 | -0.008 | 0.000 | 13 |

*Report with case statistics used for preceding diagnostics graphs*

Remember that you can right-click any number on reports of this nature from Design-Ease software and pull up context-sensitive Help with statistical details.

# Modifying the Design Layout

Design-Ease software offers a great deal of flexibility to modify data in its design layout. Let's see what can be done in the bowling case.

The outcome of the bowling match appears to be definitive, especially from Mark's point of view. However, Pat and Shari demand one more chance to prove themselves worthy of the team. They still think Mark's high game of 195 was a fluke even though this wasn't borne out by the diagnostic analysis. Of course, Mark objects and an argument ensues.

To achieve compromise, the team captain decides to toss out the highest and lowest games for each of the three bowlers, and replace them with two new scores each. A newly-hired programmer and avid bowler, Ben, shows up at the alley and he's allowed to participate in this second block of runs. (Yes, we know this makes little sense, but it will add some interest to this exercise on Design-Ease's flexibility for design and analysis of experiments – no matter how convoluted they become in actuality.)

It quickly becomes apparent that this new kid does things differently. He's a lefty with a huge hook that's hard to control. To aggravate this variability, Ben does something very different from other bowlers – he does not put his thumb in the hole made for that purpose on the ball. When Ben's odd approach works, the pins go flying, but likely as not that ball slides off into the left gutter or careens over the edge on the right.

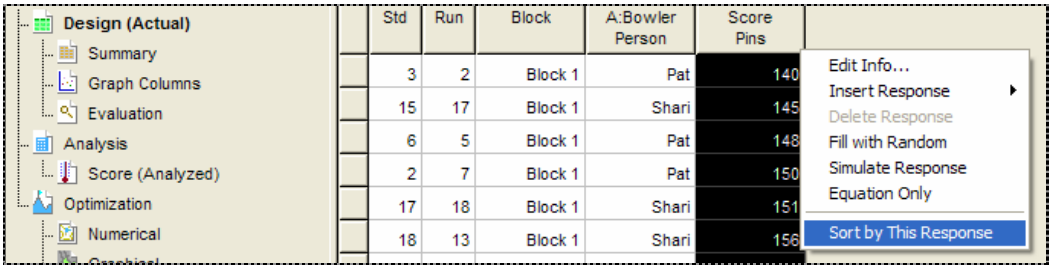
The results for Ben and the three original bowling team candidates can be seen below.

| Block | Game | Pat | Mark | Shari | Ben |
|---|---|---|---|---|---|
| 1 | 1 | 160 | ~~165~~ | ~~166~~ | NA |
| 1 | 2 | 150 | 180 | 158 | NA |
| 1 | 3 | ~~140~~ | 170 | ~~145~~ | NA |
| 1 | 4 | ~~167~~ | 185 | 161 | NA |
| 1 | 5 | 157 | ~~195~~ | 151 | NA |
| 1 | 6 | 148 | 175 | 156 | NA |
| 2 | 1 | 162 | 175 | 163 | 200 |
| 2 | 2 | 153 | 180 | 166 | 130 |

*Bowling scores with high and low games replaced by two new games (plus a new guy)*

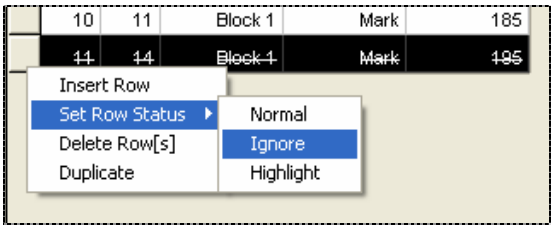
To enter this new data (and cross out some of the old), start by clicking on the **Design** node at the upper left of your screen. You now should see the bowling data from before. Mark's high game will remain highlighted from before if you clicked on it as instructed while doing the diagnostics.

Right click at the top of the **Response** column and choose **Sort by This Response**.




*Sorting Runs by Response*

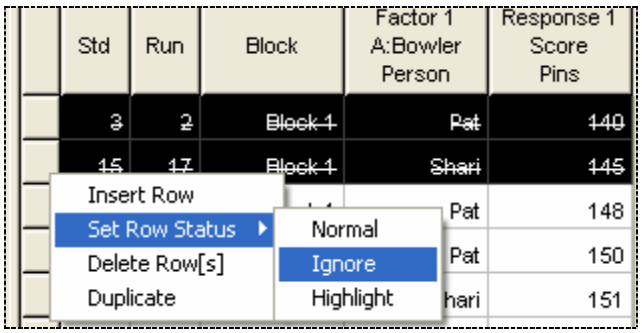
Now Mark's best game drops to the very bottom. Let's pick on him first to placate Pat and Shari. Right-click on the square button to the left of the last row (Mark's score of 195) and select **Set Row Status** and then **Ignore** as shown below.




*Ignoring Mark's high game*

By the way, it's OK to change your mind on such matters: You can 'un-ignore' a row by Set Row Status to Normal.

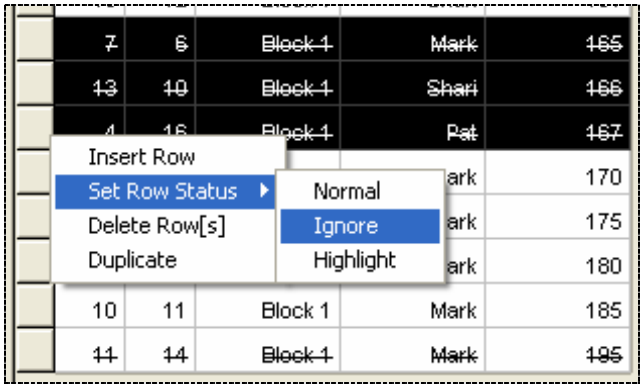
Next let's really get the hopes high for Pat and Shari by excluding their low games from consideration. Click the square button to the left of the first row (Pat's low game) and while holding down the **Shift** key click the button left of the second row as well. Then while keeping our mouse over this column of buttons, do a right-click and **Set Row Status**, **Ignore** these two low games.




*Ignoring the low games for Pat and Shari*

Before continuing on, look at the screen shot below. Go down a number of rows and click the button to the left of Mark's low game of 165.

Notice that the following two rows are the high games for Shari (166) and Pat (167). Finally it's time to them make pay a price for complaining: Hold down the **Shift** key and click the button to the left of the last row in this series of runs; and then right-click to **Set Row Status**, **Ignore**.



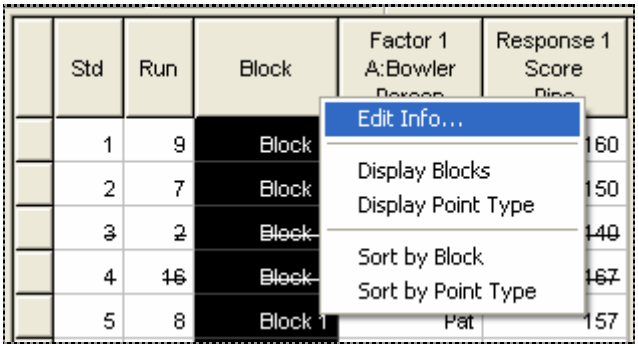

*Ignoring the Mark's low game and the high games for Shari and Pat*

Return the layout to the original order by right-clicking on the **Std** column heading and choosing **Sort by Standard Order**. You'd best check what's on your screen with what we show below. If you got off somewhere, try fixing it now.

| | Std | Run | Block | Factor 1 A:Bowler Person | Response 1 Score Pins |
|---|---|---|---|---|---|
| | 1 | 9 | Block 1 | Pat | 160 |
| | 2 | 7 | Block 1 | Pat | 150 |
| | ~~3~~ | ~~2~~ | ~~Block 1~~ | ~~Pat~~ | ~~140~~ |
| | ~~4~~ | ~~16~~ | ~~Block 1~~ | ~~Pat~~ | ~~167~~ |
| | 5 | 8 | Block 1 | Pat | 157 |
| | 6 | 5 | Block 1 | Pat | 148 |
| | ~~7~~ | ~~6~~ | ~~Block 1~~ | ~~Mark~~ | ~~165~~ |
| | 8 | 15 | Block 1 | Mark | 180 |
| | 9 | 4 | Block 1 | Mark | 170 |
| | 10 | 11 | Block 1 | Mark | 185 |
| | ~~11~~ | ~~14~~ | ~~Block 1~~ | ~~Mark~~ | ~~195~~ |
| | 12 | 3 | Block 1 | Mark | 175 |
| | ~~13~~ | ~~10~~ | ~~Block 1~~ | ~~Shari~~ | ~~166~~ |
| | 14 | 1 | Block 1 | Shari | 158 |
| | ~~15~~ | ~~17~~ | ~~Block 1~~ | ~~Shari~~ | ~~145~~ |
| | 16 | 12 | Block 1 | Shari | 161 |
| | 17 | 18 | Block 1 | Shari | 151 |
| | 18 | 13 | Block 1 | Shari | 156 |

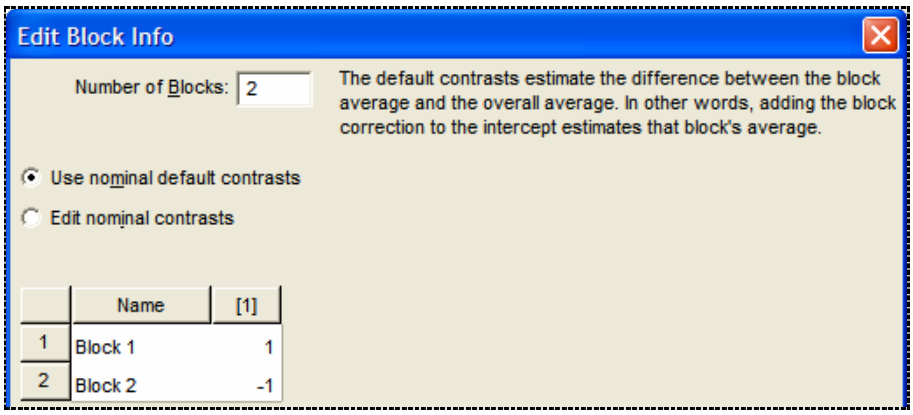
*Back to standard order after low and high game ignored for each bowler*

Now, create a new block (need this for the second round of bowling) by right-clicking at the top of the **Block** column and choosing **Edit Info**.




*Creating a new block*

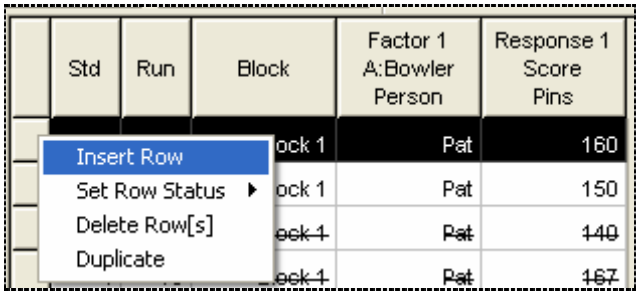
You now see a form that allows users to give whatever name they want to the block(s). Don't bother doing this now. As shown below, change the **Number of Blocks** to 2. Press the **Tab** key to see this change take effect.




*Adding a second block of runs*

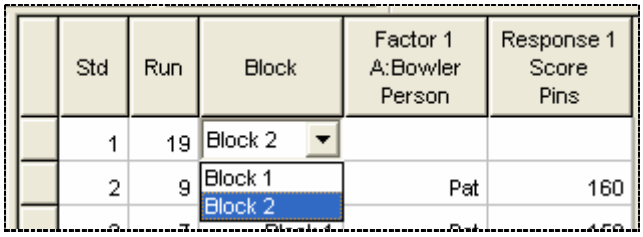
Click **OK**. It seems that nothing changed, but actually the program now knows that you will be doing another block of runs.

Now you are ready to begin adding and/or duplicating rows. This can be accomplished in different ways, depending on your ingenuity. We will follow a procedure that exercises as many of the editing features as possible, so it may not be the most elegant approach. Right click on the square button at the left of the first row to bring up the editing menu. Move the mouse over the first selection **Insert Row** and click.



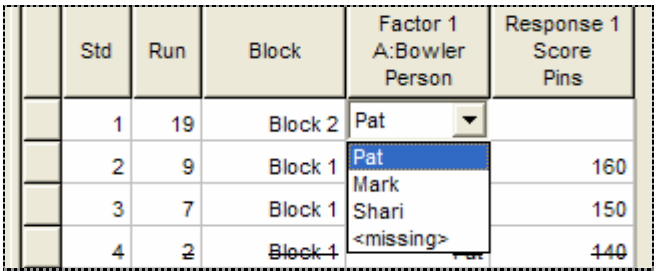

*Inserting a new row*

You now will see a new row with blanks for the bowler and the score. Click on the block field and then on the list arrow: Select **Block 2** as shown below.



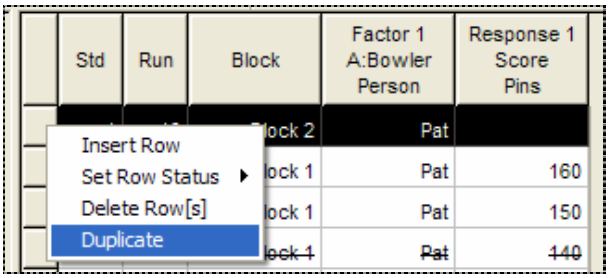

*Changing block number*

Next, click on the blank field for bowler and then on the list arrow. Select **Pat**. (If this were a numerical field, you would simply enter the value.)



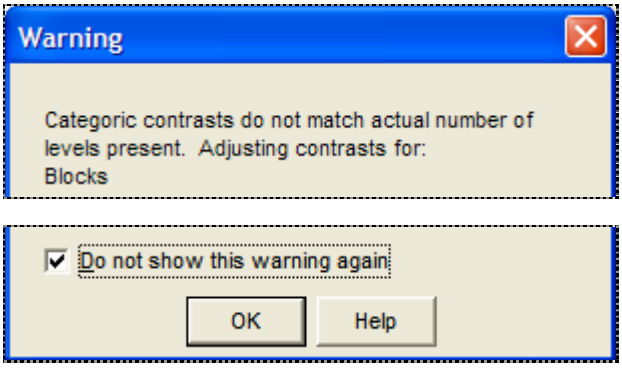

*Entering a categorical value for factor*

Again right-click on the open square at the left of the first row to bring up the editing menu. Move the mouse over the selection to **Duplicate** and click.




*Duplicating a row*

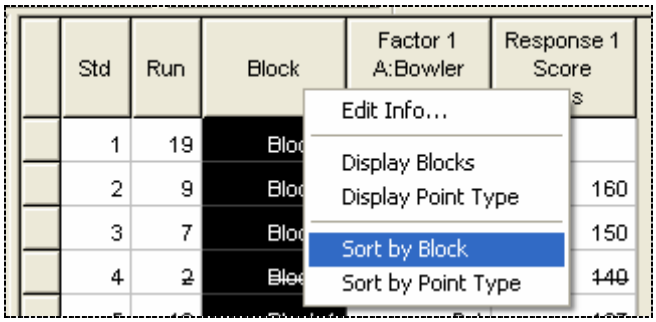
Somewhere along the way, Design-Ease may pop up a warning like the one shown below.




*Warning about categoric contrasts*

It does not like having only one bowler in the second block, but it needn't be worried at this stage, because you will be adding others. Click to check (√) the option **Do not show this warning again**. This will save you aggravation later. Don't worry – the warning will be re-enabled the next time you start up the programs so you will not go unprotected indefinitely.

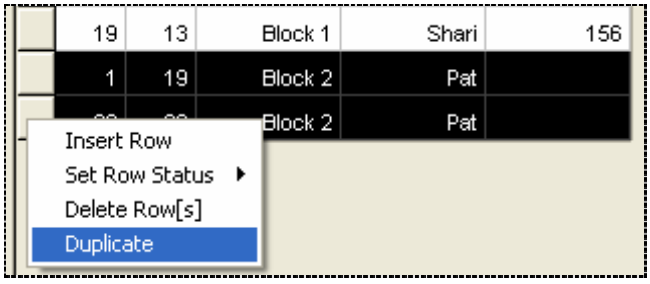
Right-click on the **Block** column heading and **Sort by Block**.




*Sorting by block*

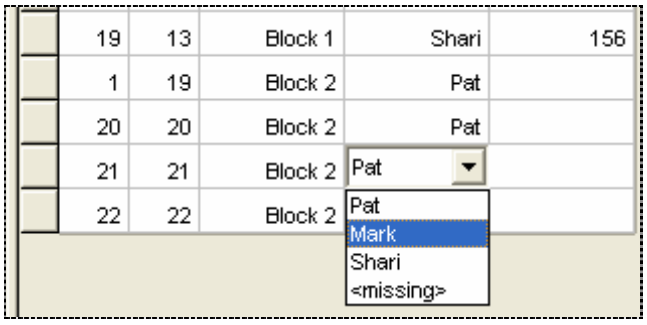
Now, the two new rows can now be seen at the bottom of the design layout. We need two new rows apiece for Shari and Mark. Let's just duplicate the new rows for Pat and change the bowler's name. Do this by first left-clicking the button (open square) to the left of the first new row for Pat, so it is highlighted. Then while holding down the **Shift** key, click the button to the left of the second new row for Pat. Now both rows should be highlighted. (This is a bit tricky, but it will save time.)

Now you can right-click on any button (open square) on the left-most part of the highlighted block and select **Duplicate**. (If the warning screen pops up again, click OK.)




*Duplicating a block of rows*

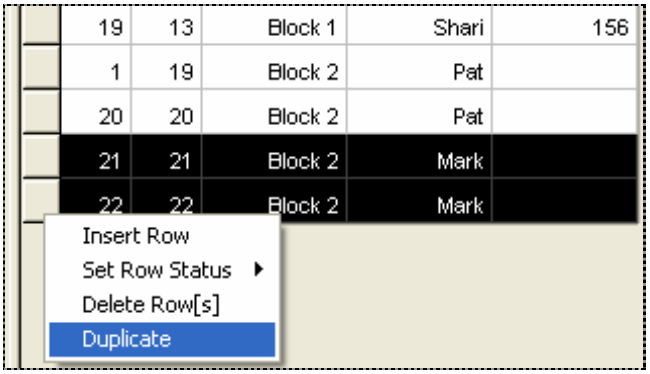
In the first duplicated row, click on the field for **Bowler** and select **Mark**.




*Changing name of bowler*

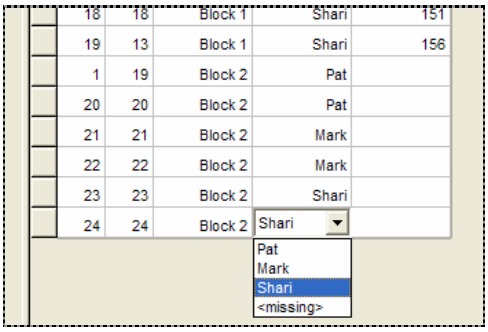
Do the same for the last row. You now should have two new rows for both Pat and Mark. Click the button to the left of the first new row for Mark, so it is highlighted. Then while holding down the **Shift** key, click the button to the left of the second new

row for Mark. Both rows should now be highlighted. Right-click on any button (open square) in the left-most part of the highlighted block and select **Duplicate**.



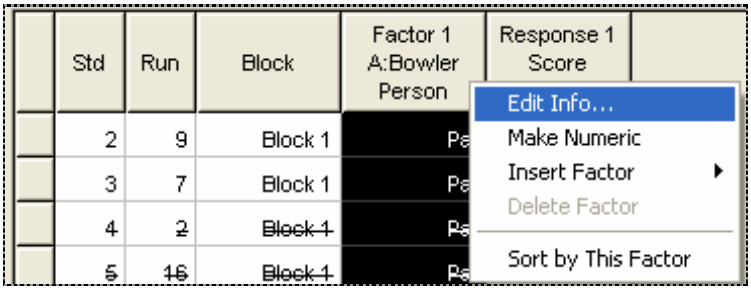

*Duplicating two more rows*

In the first duplicated row, click on the field for **Bowler** and select **Shari**. Do the same for the last row.



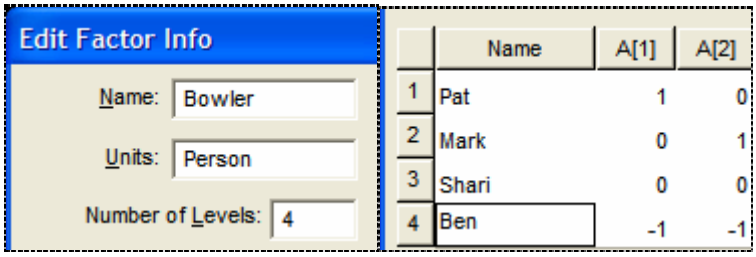

*Completing lineup for block 2 – the second round of bowling*

But what about the new kid – Ben? We need to identify him as a new competitor in this bowling contest. Do this by right-clicking the header for **Bowler**.



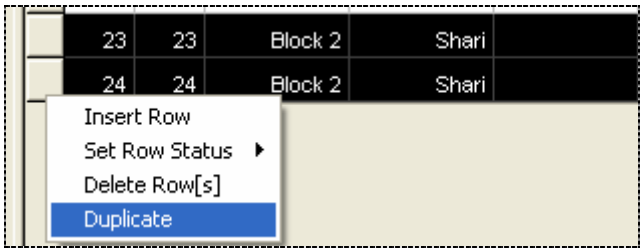

*Getting ready to add a new level for the factor*

Change the **Number of Levels** to **4** and then **Tab** to the new field for **Name** and enter **Ben**.



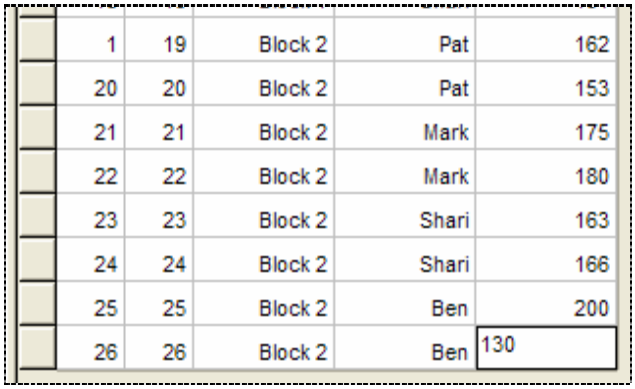

*Entering the new bowler*

Press **OK**. Now duplicate two more rows by clicking the button to the left of the first of Shari's two new games and then while holding down the **Shift** key, clicking the button to the left of the last run, and finally right-clicking a button in the left-most part of the highlighted block and selecting **Duplicate**.



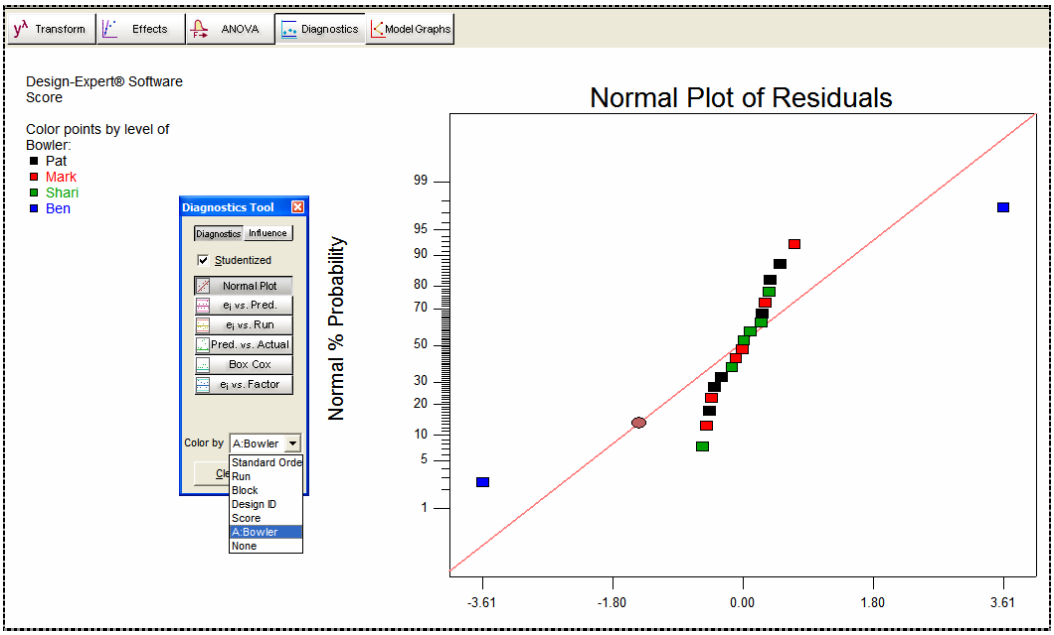

*Duplicating two more rows so the new bowler can be included*

In both these new duplicated rows, click on the field for **Bowler** and select **Ben**. You are now ready to enter the new data as shown below.




*Data Entered for Second Block of Games*

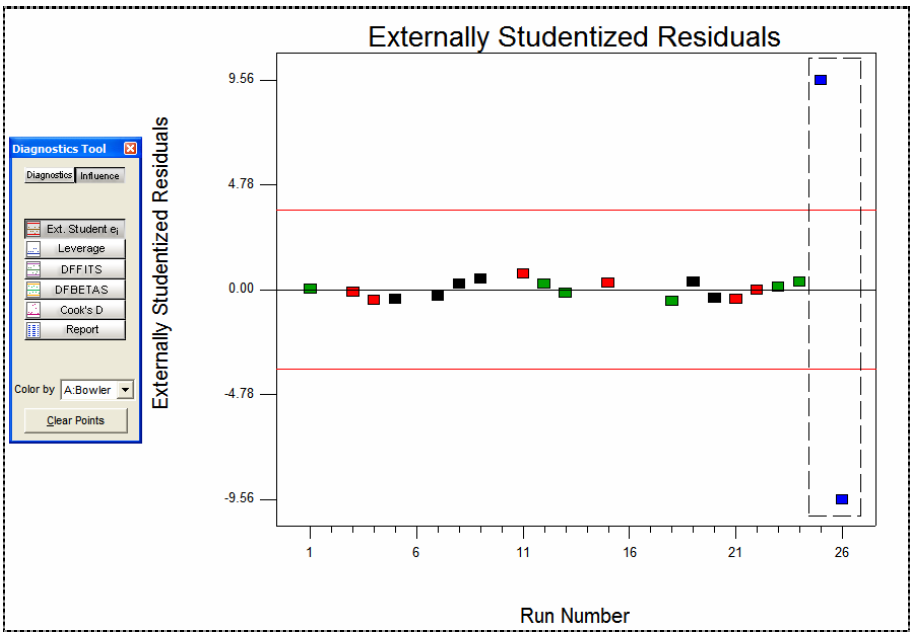
Go ahead now and re-analyze the data by clicking under **Analysis** the node for **Score**. Then press ahead through ***Transform***, ***Effects***, ***ANOVA*** and ***Diagnostics***. Do not worry if the model gets labeled "insignificant." As you will see, something is abnormal about the data: Do you notice that the residuals now line up very poorly, especially at the extreme points? On the floating **Diagnostics Tool** change **Color by** to **A: Bowler**.




*Diagnostics for bowling results – part two: Normal plot colored by bowler*

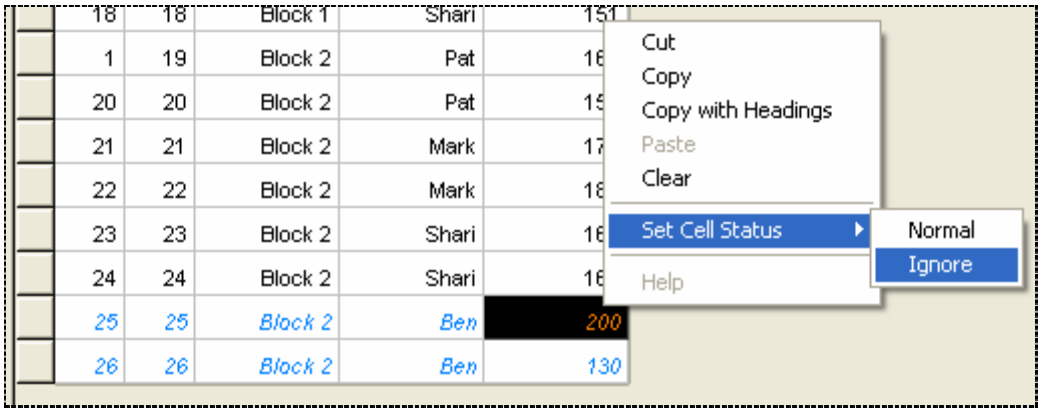
Now you see that the results from Ben do not fit with the others, which should be no surprise considering his odd style of bowling. Click the **Influence** button to bring up the externally studentized residuals – a good tool for detecting outliers.

Drag your mouse over the Ben's residuals at the far right. Both points should now become highlighted. We need to ignore or delete them. (Sorry Ben, odd behavior by programmers is considered normal at Stat-Ease, but not when it comes to their bowling team!)



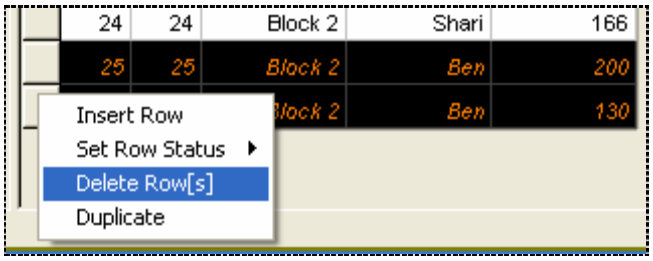

*Ben's games selected on 'outlier' plot (externally studentized residuals)*

Click the **Design** node to get back to the home base of the design layout. Notice that Ben's games are conveniently highlighted so they can be given the axe. There are a number of ways to do this in Design-Ease software. It provides no advantage in this case, which features only one response measure, but you can ignore a specific result by right-clicking that cell and setting Set Cell Status to Ignore as shown below.




*Ignoring a single cell – an option that's not recommended for this case*

In this case it will be much better to ignore the entire runs in the manner described earlier in this tutorial, or perhaps better yet, simply delete them altogether. No offense to Ben, but given that he only got the chance to bowl two games and his unorthodox style creates such abnormal variability, it will be best now to click the button to the left of his first score of 200 (making him feel really bad ☹), shift-click the button for the second game of 130 (not so sorry to see this gone!), then without moving your mouse right-click and select **Delete Row(s)**.




*Deleting Ben's games*

Click **Yes** on the warning that pops up about deleting rows (a safety precaution) and **OK** for the heads-up the program gives you about eliminating an entire categoric contrast (no more Ben). Then go ahead and re-analyze the results.

It turns out that the added games cause no change in the overall conclusions as to who's the better bowler. Mark remains on top. It would now be appropriate to recover the low and high games for each bowler from block 1. Since this data was not deleted, only ignored, getting it back is simply a matter of right-clicking to the left of each of the six suspect rows and changing Set Row Status to Normal. Give this a try! Then re-analyze one last time.

By working through this exercise you now see how easy it is to manipulate data in Design-Ease's design layout. Keep this in mind should the need arise for editing your own data.

PS. Are you still feeling bad about deleting Ben's scores? Do not worry: He gets to bowl with Pat and Sherry in a lesser league. After bowling for an entire year (roughly 100 games), it will become clear whether Ben's crazy way of bowling will pay off by achieving a good average overall.