SPSS Tutorial 5: Normal Distribution & Z-Scores

Goals

- To calculate standard scores (aka *z*-scores) using the conceptual formula, but using SPSS (instead of a calculator) to do the calculations
- To calculate standard scores using the DESCRIPTIVES procedure

Calculating *z*-scores using the conceptual formula

A *z*-score is a transformed score that tells you the distance of a score from the mean of its distribution, with distance measured in standard deviation units. Putting that into symbols:

$$z_X = \frac{X - \mu_X}{\sigma_X} \qquad z\text{-score for population data}$$

$$z_X = \frac{X - \bar{X}}{s_X} \qquad z\text{-score for sample data}$$

As you will recall from Tutorial 4, σ_X = the population standard deviation (with division by N), and s_X = the sample standard deviation (with division by *n*-1).

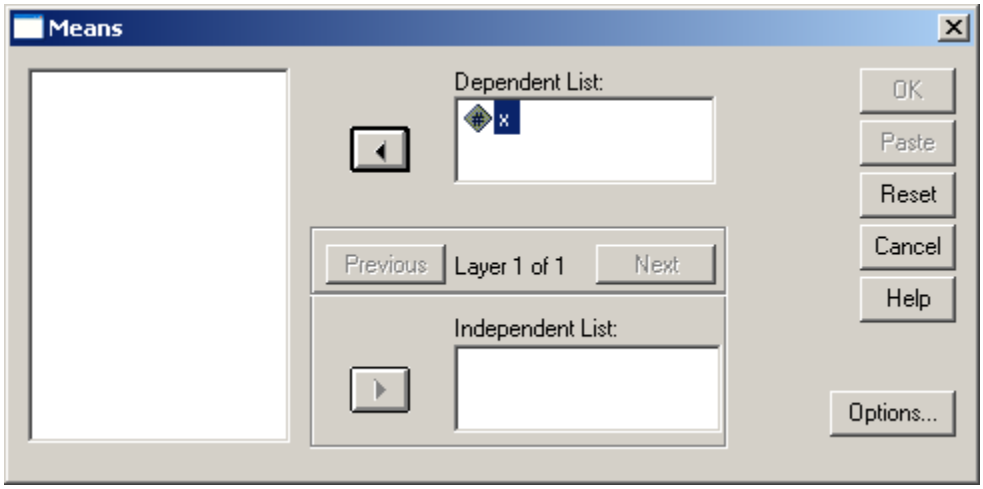
Let's start by reading in the X-scores shown in Table 4-1 of *Bare Essentials* (2nd Edition, p. 29). The following DATA LIST syntax can be used to read in the data.

```
DATA LIST FREE / X (f2.0).
BEGIN DATA.
1 3 4 7 9 9 11 12 16 18
END DATA.
```

To compute the two *z*-scores shown above (i.e., one for population data, and one for sample data), we need the mean of the *X*-scores, the population SD, and the sample SD. You can obtain the mean and sample SD using the MEANS procedure, which you used in earlier tutorials. You may recall that when you use the pull-down menu approach to generate your MEANS syntax (Analyze→Compare Means→Means), the dialog box requires you to select at least one dependent variable and at least one independent variable. But your current working data file has only one variable, X, so this is not possible (see example of dialog box below). Therefore, in order to use the MEANS procedure when there is no independent variable, you must type the command directly into the syntax window yourself. For your current data file, the command is as follows:

```
means x.
```

Dialog box for MEANS procedure with only 1 variable (X) in the data file.



Now use the results of the MEANS procedure in COMPUTE statements to create new variables MEAN.X, N, and S.X (where N = the number of scores, and S.X = the sample SD of the X scores).

Report

X

N	Mean	Std. Deviation
10	9.00	5.497

The output table from which you take the values for N, MEAN.X, and S.X should be as shown above. Notice that the SD is reported to only 3 decimal places. However, many more decimal places are available, and should be used to avoid rounding error. To copy the SD with all available decimal places, proceed as follows:

- Double-click on the Report table that shows the N, Mean and SD for variable X (in the Output window); this will open the pivot table for editing
- Double-click on the cell that shows the standard deviation (i.e., the cell with 5.497 in it); this will open that cell for editing, and the SD will be shown to several decimal places; it will also be highlighted
- When the number in the SD cell is highlighted, right-click it and select Copy
- Then click anywhere in the Output window to close the pivot table
- Switch to your syntax window, and paste the value of the SD into your compute statement, as shown below

```
compute s.x = 5.49747416749.
```

Use another COMPUTE statement to create variable SIGMA.X, the population SD. Recall from Tutorial 4 that:

$$\text{Population SD} = \text{Sample SD} \times \sqrt{\frac{N-1}{N}}$$

Your file should now look like this:

X	MEAN.X	N	S.X	SIGMA.X
1	9.0000	10	5.4975	5.2154
3	9.0000	10	5.4975	5.2154
4	9.0000	10	5.4975	5.2154
7	9.0000	10	5.4975	5.2154
9	9.0000	10	5.4975	5.2154
9	9.0000	10	5.4975	5.2154
11	9.0000	10	5.4975	5.2154
12	9.0000	10	5.4975	5.2154
16	9.0000	10	5.4975	5.2154
18	9.0000	10	5.4975	5.2154

We now have all the pieces we need to compute z-scores assuming we have population data, and z-scores assuming we have only sample data. Type the following COMPUTE statements into your syntax file, and run them. Then look at your data file.

```
compute z.pop  = (x - mean.x) / sigma.x.
compute z.samp = (x - mean.x) / s.x.
exe.
```

You should now add some variable labels, if you have not already done so.

```
var lab
 mean.x   'Mean'
 s.x      'Sample SD'
 sigma.x  'Population SD'.
 z.pop    'Z-score (population data)'
 z.samp   'Z-score (sample data)'.
```

Here's what you should now see in your data file:

X	MEAN.X	N	S.X	SIGMA.X	Z.POP	Z.SAMP
1	9.0000	10	5.4975	5.2154	-1.53	-1.46
3	9.0000	10	5.4975	5.2154	-1.15	-1.09
4	9.0000	10	5.4975	5.2154	-.96	-.91
7	9.0000	10	5.4975	5.2154	-.38	-.36
9	9.0000	10	5.4975	5.2154	.00	.00

9	9.0000	10	5.4975	5.2154	.00	.00
11	9.0000	10	5.4975	5.2154	.38	.36
12	9.0000	10	5.4975	5.2154	.58	.55
16	9.0000	10	5.4975	5.2154	1.34	1.27
18	9.0000	10	5.4975	5.2154	1.73	1.64

Notice that the z-scores in the Z.POP column match the z-scores reported in Table 4-1 of Bare Essentials. Thus, despite that fact that equation 4-1 appears to give the formula for the z-score using sample data, the authors obviously used the population SD (σ), not the sample SD (s).

Computing z-scores with the DESCRIPTIVES Procedure

As you may have guessed, there is a somewhat easier way to compute z-scores in SPSS. Try the following:

- Click on **Analyze→Descriptive Statistics→Descriptives**
- Move X into the Variable(s) box
- Put a check in the **Save standardized values as variables** box
- Click on PASTE, and run the syntax

Look in your data file, and you will see that a new variable, ZX, has been added.

X	SIGMA.X	S.X	Z.POP	Z.SAMP	ZX
1	5.2154	5.4975	-1.5339	-1.4552	-1.4552
3	5.2154	5.4975	-1.1504	-1.0914	-1.0914
4	5.2154	5.4975	-.9587	-.9095	-.9095
7	5.2154	5.4975	-.3835	-.3638	-.3638
9	5.2154	5.4975	.0000	.0000	.0000
9	5.2154	5.4975	.0000	.0000	.0000
11	5.2154	5.4975	.3835	.3638	.3638
12	5.2154	5.4975	.5752	.5457	.5457
16	5.2154	5.4975	1.3422	1.2733	1.2733
18	5.2154	5.4975	1.7257	1.6371	1.6371

Notice that the values of ZX are identical to the values of Z.SAMP. In other words, when you use the DESCRIPTIVES procedure to obtain z-scores, SPSS computes z-scores using the sample SD (division by n-1), not the population SD. If need z-scores computed using a population SD (division by N), you will have to compute the z-scores yourself, as we did earlier.

Using SPSS to Obtain the Area under the Normal Curve

Table A in Bare Essentials is a table of the standard normal distribution. The Area below column in that table shows the proportion of area between 0 and the z-score listed in that row. For example, the proportion of area between 0 and z = .40 = 0.1554 (or 15.54%). Let's see if we can create our own version of Table A in an SPSS data file.

The first thing we have to do is get z-values from 0.00 to 4.00 (with steps of 0.01) into a data file. That's a lot of numbers to type. Fortunately, there is an easy method for creating such a file. Copy the following syntax into your syntax window and run it.

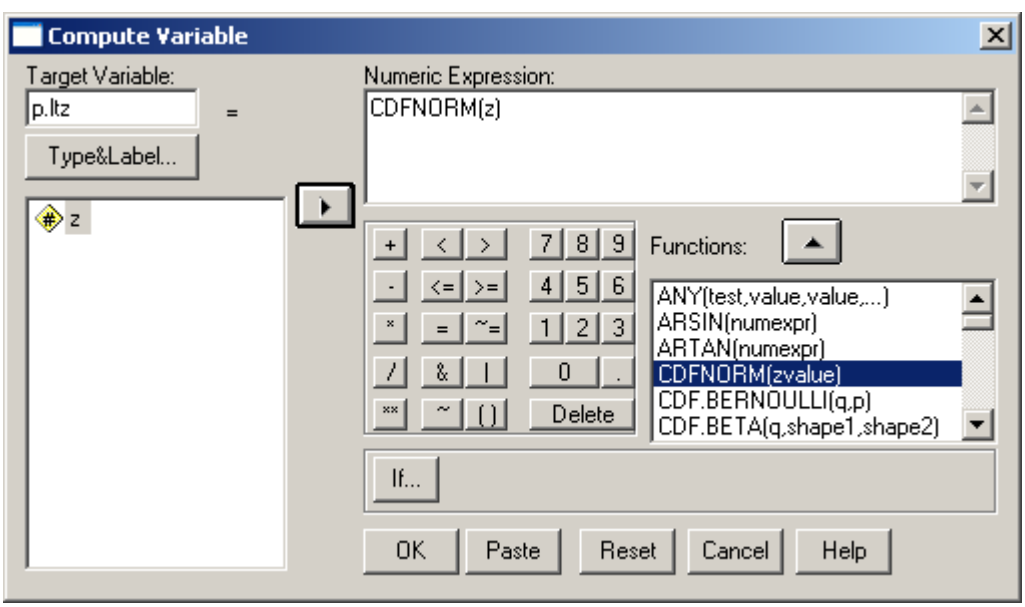
```
* --- Create Table of Standard Normal Distribution ----- .

* Create new data file with z = 0.00 to 4.00,
* as in Table A of Bare Essentials .

new file.
input program.
loop #i = 0 to 400.
      compute z = #i/100.
      end case.
end loop.
end file.
end input program.
exe.
```

Now we need to compute the values shown in the **Area Below** column, i.e., the proportion of area between 0 and each z-score. Just for fun, let's also include an **Area Above** column, which gives the proportion of area under the curve beyond each z-score. (Many tables of the standard normal distribution do in fact include an **Area Above** column.)

In the dialog box for COMPUTE (under **Transform/Compute** when the data editor is active), one of the available functions is **CDFNORM(zvalue)**. Select this function from the function list, and move it into the **Numeric Expression** box, as shown below. Make your variable Z the argument for the CDFNORM function—i.e., it should say CDFNORM(z). And make the Target Variable name p.ltz (for proportion of area less than z). Then paste the syntax and run it.



If you look in your data file, the first few records will now look like this:

Z	P.LTZ
.00	.5000
.01	.5040
.02	.5080
.03	.5120
.04	.5160
.05	.5199

The CDFNORM(z) function computed the proportion of area less than z, and of course, half of the area under the normal curve falls below z = 0. We forgot to subtract 0.5 from the result of the CDFNORM(z) function. So, we need to edit our compute statement to make it look like this:

```
COMPUTE p.ltz = CDFNORM(z) - 0.5.
```

And we also need to add a compute statement to create variable `p.gtz`, which will show the proportion of area greater than z. We need to format our two area columns to show 4 decimal places, as in Table A; and as always, we should add some variable labels.

```
compute p.gtz = 0.5 - p.ltz.
EXECUTE .
format p.ltz p.gtz (f8.4).
var lab
 p.ltz    'Area between 0 and z'
 p.gtz    'Area beyond z'.
```

Finally, we need to display our table. There are various ways to do this including the simple LIST command.

```
LIST.
```

You may also want to try a couple of the other methods for listing cases, because they produce much sharper looking output. Try the following:

- Analyze→Reports→Case Summaries
- Analyze→Reports→Report Summaries in Rows

Here are the areas I calculated for selected values of z.

Area Under the Normal Curve for Selected z-values

	Z	Area between 0 and z	Area beyond z
1	1.64	.4495	.0505
2	1.65	.4505	.0495
3	1.96	.4750	.0250
4	2.32	.4898	.0102
5	2.33	.4901	.0099
6	2.57	.4949	.0051
7	2.58	.4951	.0049

This table was created using Analyze→Reports→Case Summaries. Here's the syntax I used:

```
compute f = any(z,1.64,1.65,1.96,2.32,2.33,2.57,2.58).
filter by f.
exe.

SUMMARIZE
  /TABLES=z p.ltz p.gtz
  /FORMAT=VALIDLIST NOCASENUM TOTAL
  /TITLE='Area Under the Normal Curve for Selected z-values'
  /MISSING=VARIABLE
  /CELLS=NONE.
```

That's all for this Tutorial. Save your syntax file for future reference.